

STATISTICAL PATTERN RECOGNITION APPROACHES FOR RETRIEVAL-BASED MACHINE TRANSLATION SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Dwi Sianto Mansjur

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

STATISTICAL PATTERN RECOGNITION APPROACHES FOR RETRIEVAL-BASED MACHINE TRANSLATION SYSTEMS

Approved by:

Professor Biing-Hwang (Fred) Juang,
Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Mark Clements
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Elliot Moore II
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Ye (Geoffrey) Li
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Ming Yuan
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Date Approved: 15 September 2011

To my family,
Mansjur, Mei Sen,
Yenty, Dewi, Maya, Herna, Dwifuzi,
and Yuan Yuan Li.

ACKNOWLEDGEMENTS

Experiencing the twists and turns of a research journey is as valuable as the dissertation writing itself. Many roadblocks have to be cleared in between the twists and turns of the problem formulations and solution presentations. Nevertheless, looking back at the beginning of this research journey, I am grateful to have the final writing turn out to be like this. This dissertation marks an important milestone in my life. An equally important milestone is the friendships that I have made along the research journey.

I would like to express my deepest gratitude to my advisor, Professor Biing Hwang Juang. I am grateful for his insightful criticisms and patient encouragement throughout my thesis writing. His insights have enabled me to develop an appreciation for pattern recognition systems and natural language processing fields. He also spent countless hours showing me how to approach a research problem as well as teaching me how to present my work in a scholarly manner. I have been fortunate to be trained under the expertise of Professor Juang.

My special thanks also go out to Professor Mark Clements and Professor Elliot Moore for serving on the reading committee of my dissertation. They have provided me with valuable feedback on my research. I also thank Professor Ye Li and Professor Ming Yuan for serving on my thesis committee and for providing me with valuable feedback. I would like to take this opportunity to thank Professor James H. McClellan for his assistance during my early years of graduate school. I was fortunate to be a graduate teaching assistant for his course.

I am grateful to the Center for Signal and Image Processing (CSIP) staff, Catherine Gholson, Tammy Scott, and Stacie Speights. Thanks to their efforts, the students are

being offered the resources to do their research. I have also been very fortunate to have the company of great friends at Georgia Tech. In particular, I owe special thanks to Ted Wada for his assistance in the formulation of the model regularization strategy and Antonio Moreno for the explanation of the minimum verification error principle. I also owe special thanks to Deryck Yeung, Rungsun Munkong, Gaofeng Yue, Enrique Robledo Arnuncio, Woojay Jeon, Soo Hyun Bae, Qiang Fu, Yong Zhao, and Sabato Marco Siniscalchi for their cheerful talks and for their dependable friendships. It has been a great pleasure having Umair Altaf, Jason Wung, Mingyu Chen, SungHwan Shin, Jinyu Li, Jeremy Reed, Chengyuan Ma, Yu Tsao, and Jinwoo Kang as friends, and it will be pleasant to remember our time together.

I thank my parents Mei Sen Hong and Mansjur, my sisters Yenty, Dewi, Maya and Herna, and my brother Dwifuzi Mansjur, for their endless love, support, encouragement, and sacrifice throughout my life and during my Ph.D. studies. I also want to express my gratitude to and thank my wife, YuanYuan Li, who has given me tremendous support and positive encouragement as long as I have known her. She has given me a lot of constructive criticism on my writing as well as my presentation. Without her patience and love, this work would not have been completed.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	x
SUMMARY	xiii
I INTRODUCTION AND BACKGROUND	1
1.1 Relevant Natural Language Processing Techniques	4
1.1.1 Semantic Feature Construction	4
1.1.2 Statistical Text Categorization System	10
1.1.3 Machine Translation Systems	12
1.2 Relevant Pattern Recognition Techniques	16
1.2.1 Minimum Classification Error Principle	16
1.2.2 Support Vector Machine Classification	23
1.3 Organization of the Dissertation	26
II EMPIRICAL PATTERN CLASSIFIER DESIGNS	28
2.1 Empirical Mixture Model	29
2.1.1 Kernel Functional Form	30
2.1.2 Corrective Bandwidth Learning Algorithm	32
2.1.3 Experimental Results	34
2.2 Classification-Based Model Selection	41
2.2.1 Mixture Model Selection	42
2.2.2 Incremental Mixture Learning Procedure	44
2.2.3 Experimental Results	46
2.3 Variability Regularization in Margin-Based Classifiers	49
2.3.1 Empirical Risk Minimization Principle	51
2.3.2 Regularized Empirical Error Margin	54

2.3.3	Experimental Results	60
2.4	Summary	65
III	SUBJECTIVE JUDGMENTS IN THE ERROR-COST LEARNING PROCEDURE	67
3.1	Active Control of Interclass Confusions	68
3.1.1	Minimum Error-Cost Learning Procedure	70
3.1.2	Experimental Results	76
3.2	Multi-level Relevance Performance Measure	87
3.2.1	Multi-Level Irrelevance Performance Measure	89
3.2.2	Differentiable Relevance Measure	90
3.2.3	Experimental Results	96
3.3	Summary	99
IV	RETRIEVAL-BASED MACHINE TRANSLATION SYSTEMS 101	
4.1	Design Compromises in Machine Translation Systems	103
4.2	Retrieval-Based Machine Translation System	109
4.2.1	Construction of a term-to-term association matrix	114
4.2.2	Identification of document topics	115
4.2.3	Generation of sensible variations for each source feature vector	117
4.3	Experimental Results	118
4.4	Summary	125
V	CONCLUSION AND FUTURE WORK	126
5.1	Summaries and Contributions	129
5.2	Avenues of Future Research	131
	REFERENCES	133
	VITA	143

LIST OF TABLES

1	The descriptions of the datasets used to verify the corrective bandwidth learning algorithm.	39
2	The average accuracy rates for the conventional KDE-based classifiers based on ROT, MLCV and NNE, and the standard OAO SVM classifier.	40
3	The average accuracy rates for the KDE-based classifiers with the corrective learning algorithm (the EMM classifiers), and the standard OAO SVM classifier.	40
4	The average accuracy and the standard deviation (in parentheses) of four types of mixture learning techniques using a ten-fold cross-validation procedure.	48
5	The average and the standard deviation (in parentheses) of the number of free parameters for verifying mixture learning algorithms using ten-fold cross-validation.	48
6	Confusion matrices of true (b), baseline (c), and cost-optimized (d) models based on cost matrix (a) for cost-insensitive classification.	79
7	Confusion matrices of true (b), baseline (c), and cost-optimized (d) models based on cost matrix (a) for cost-sensitive classification.	79
8	The mean (standard deviation) of row one and column four of the multi-class confusion matrix across true models, baseline models, and cost-optimized models.	80
9	The signs of difference between the R1C4 entries of the confusion matrices produced by the baseline and the cost-optimized models.	80
10	The label-difference cost matrix assigns error-cost based on the absolute value of the difference in the class labels. The exponential-label-difference cost matrix is obtained from taking the exponential of the difference in class labels.	82
11	The normalized multi-class confusion matrices using the baseline (a) and cost-optimized (b) models for features with 80% of the total PCA variances using the label-difference cost matrix.	83
12	The performance measures of the baseline and cost-optimized models using the label-difference and the exponential-label-difference cost matrices.	84
13	The average error rate across different costs in the confusion matrix using the label-difference and the exponential-label-difference cost matrices.	86

14	The Weighted Total Confusion (WTC) of the baseline and cost-optimized confusion matrices.	86
15	Six examples of multi-level relevance ranking lists demonstrating the difference between the conventional DCG criterion and the proposed PCC criterion. The DCG criterion indicates the value to be maximized and emphasizes the gain value. However, the PCC criterion denotes the cost to be minimized and emphasizes the ranking loss.	91
16	Degree of relevance for the 20 Newsgroups dataset is assigned based on the following tri-relevance measure: a relevance of two for documents in the same sub-categories, a relevance of one for documents in the same major category, and a relevance of zero otherwise.	97
17	Compromises among external and internal qualities due to the lengths of translation units	109
18	Semantic categories for the hotel reservation task in the database . .	119
19	Examples of parallel English-Indonesian sentences used to train a domain-specific translation system for a hotel reservation purpose.	120

LIST OF FIGURES

1	Graphical model for the probabilistic latent semantic analysis (PLSA) technique for a collection with N documents and T -word lexicon as listed in the bottom-right corner. Shaded and non-shaded variables indicate observed and latent variables, respectively.	8
2	Conventional text categorization procedures based on the statistical pattern recognition procedure.	11
3	Grouping of rule-based machine translation systems based on the linguistic abstraction with the interlingua approach at the top, followed by the transfer-based approach, and the direct approach.	13
4	The three main steps in RBMT and EBMT systems. The steps in RBMT and EBMT systems are written in italics and upper case respectively.	15
5	Architecture of a conventional SMT system using a language model and a translation model.	16
6	Examples of the smooth zero-one loss functions.	21
7	Illustration of the gradients of the smooth zero-one loss functions. . .	22
8	An overview of the dissertation.	27
9	Examples of the training and testing samples used for verifying the corrective bandwidth learning algorithm.	35
10	The average error-rates for training and testing samples using conventional KDE classifiers based on ROT, MLCV and NNE.	37
11	The average error-rates for training and testing samples decrease after application of the corrective learning algorithm.	38
12	Incremental learning in discriminative mixture model classifiers. In (a), we start with one mixture for each class, i.e., class 1 and class 2. In (b), two candidates are proposed for the class with solid ellipsoid (class 1). In (c), the most discriminative candidate is added as the second mixture for class 1 using the Discriminative Information Criteria (DIC). In (d), another two candidates are proposed for the class with dash ellipsoid (class 2). In (e), the most discriminative candidate is added as the second mixture for class 2 using the Discriminative Information Criteria (DIC).	47

13	Empirical error rates for classifiers are designed according to the following 4 different scenarios. In scenario (A), classifier training and selection is carried out with the standard SVM objective. In scenario (B), classifier training and selection is carried out with the proposed regularized objective. In scenario (C), classifier training is performed with the standard SVM objective and classifier selection is carried out based on the empirical error-rate. In scenario (D), classifier training is performed with proposed the regularized objective and classifier selection is carried out based on the empirical error-rate. The ten-fold cross-validation experimental results lead us to claim that the proposed regularized objective used by the classifiers in scenarios (B) and (D) has lower empirical error rates than the standard SVM objective used by the classifiers in scenarios (A) and (C).	62
14	Scatter plot for the 3-class dataset along with decision boundaries for a margin-based classifier.	63
15	Model selection based on standard regularized SVM objective values for 13 trade-off parameter $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. The objective values are plotted at the top, and the corresponding empirical error rates are plotted at the bottom. Model selection using the standard SVM criterion chooses $\mathfrak{C} = 2^{-8}$ with the empirical error rate of 0.18. This indicates that the standard SVM optimization objective value is not directly related to the empirical error rate and that a small objective value does not correspond to a low empirical error rate.	64
16	Model selection based on the proposed regularized objective values for 13 trade-off parameter $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. The objective values are plotted at the top, and the corresponding empirical error rates are plotted at the bottom. Model selection using the proposed regularized criterion chooses $\mathfrak{C} = 2^{-8}$ with the empirical error rate of 0.08. This indicates that the proposed criterion minimizes the variability of the empirical error rate evenly across the trade-off parameters, where a small objective value corresponds to either a low empirical error rate or low variability of the empirical error rate.	65
17	Scatter plot of the true model for the experiments on active-control of interclass confusion.	77
18	Sample images from the USPS digit dataset for a digit recognition task.	81

19	Performance measures of three different retrieval systems across various recall points on the 20 Newsgroups dataset. The DCGs of the proposed approach are higher than others across all recall points. The PCCs of the proposed approach are lower at low recall points and become similar to other approaches at higher recall points. The higher precision of the proposed approach is the indirect results of the proposed learning algorithm.	98
20	The space of machine translation models	104
21	Three major steps in a conventional EBMT system. The retrieval-based MT steps are shown in lower-case; those for EBMT are in upper-case [109].	111
22	Five levels of syntactic structures: word, phrase, clause, sentence, discourse.	112
23	Performance of retrieval-based MT systems using different classification schemes is computed across different generation factors. The most effective generation factor for the proposed MT system is ten, which means ten additional feature vectors are generated for each source feature vector in the database.	122
24	Comparison of BLEU scores of the four MT systems for different generation factors of five, ten, and fifteen.	123

SUMMARY

This dissertation addresses the problem of Machine Translation (MT), which is defined as an automated translation of a document written in one language (the source language) to another (the target language) by a computer. The MT task requires various types of knowledge of both the source and target language, e.g., linguistic rules and linguistic exceptions. Traditional MT systems rely on an extensive parsing strategy to decode the linguistic rules and use a knowledge base to encode those linguistic exceptions. However, the construction of the knowledge base becomes an issue as the translation system grows. To overcome this difficulty, real translation examples are used instead of a manually-crafted knowledge base. This design strategy is known as the Example-Based Machine Translation (EBMT) principle. Traditional EBMT systems utilize a database of word or phrase translation pairs. The main challenge of this approach is the difficulty of combining the word or phrase translation units into a meaningful and fluent target text.

A novel Retrieval-Based Machine Translation (RBMT) system, which uses a sentence-level translation unit, is proposed in this study. An advantage of using the sentence-level translation unit is that the boundary of a sentence is explicitly defined and the semantic, or meaning, is precise in both the source and target language. The main challenge of using a sentential translation unit is the limited coverage, i.e., the difficulty of finding an exact match between a user query and sentences in the source database. Using an electronic dictionary and a topic modeling procedure, we develop a procedure to obtain clusters of sensible variations for each example in the source database. The coverage of our MT system improves because an input query text is matched against a cluster of sensible variations of translation examples instead of

being matched against an original source example. In addition, pattern recognition techniques are used to improve the matching procedure, i.e., the design of optimal pattern classifiers and the incorporation of subjective judgments.

A high performance statistical pattern classifier is used to identify the target sentences from an input query sentence in our MT system. The proposed classifier is different from the conventional classifier in terms of the way it addresses the generalization capability. A conventional classifier addresses the generalization issue using the parsimony principle and may encounter the possibility of choosing an oversimplified statistical model. The proposed classifier directly addresses the generalization issue in terms of training (empirical) data. Our classifier is expected to generalize better than the conventional classifiers because our classifier is less likely to use over-simplified statistical models based on the available training data.

We further improve the matching procedure by the incorporation of subjective judgments. We formulate a novel cost function that combines subjective judgments and the degree of matching between translation examples and an input query. In addition, we provide an optimization strategy for the novel cost function so that the statistical model can be optimized according to the subjective judgments.

CHAPTER I

INTRODUCTION AND BACKGROUND

“Let us reflect about the mechanism of human translation of elementary sentences at the beginning of foreign language learning. ... Along the same lines as this learning process, we shall start the consideration of our machine translation system, by giving lots of example sentences with their corresponding translations. The system must be able to recognize the similarity and the difference of the given example sentences. Initially a pair of sentences are given, a simple English sentence and the corresponding Japanese sentence. The next step is to give another pair of sentences (English and Japanese), which is different from the first only by one word.”

(Nagao,1984) [77]

The example-based machine translation (EBMT) principle was proposed by Professor Makoto Nagao in his seminal paper, “A Framework of a Mechanical Translation between Japanese and English by Analogy Principle” [77]. In this paper, Professor Nagao indicated that a human learns a second language by an analogy-based generalization, not by a deep linguistic analysis. When a simple sentence along with its translation is memorized, variations of this sentence can also be translated using a reasoning by analogy. For example, if a Japanese translation of the sentence “This is the state-of-the-art technology” is memorized, then the sentence “This is the cutting-edge technology” can be translated correctly by analogy because the semantic relation between “state-of-the-art” and “cutting-edge” can be established using analogical reasoning. In a typical scenario, an EBMT system is supplied with a set of sentences in the source language and their corresponding translations in the

target language. The EBMT system uses the provided examples to translate other source sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct. Which example(s) an EBMT system determines to be equivalent (or at least similar enough) to the text to be translated varies from one system to another. The main advantage of the EBMT principle is the use of translation examples instead of handcrafted rules in the construction of a translation system [102]. Determining which rules to add is difficult to anticipate in a real-world translation system [103]. Designing an effective EBMT system remains a challenging problem mainly because it requires the capability to recognize the similarities and differences among example sentences [54]. Several pattern recognition techniques have been developed in the literature to address the issues of example generalization and disambiguation [2, 12, 42].

Recent advances in statistical pattern-recognition research include discriminative training procedure as an alternative to the maximum likelihood (ML) training procedure for a classifier design task [59, 106]. The main characteristic of the ML training procedure is the maximization of a likelihood function according to a predefined probabilistic model for the data. The ML training procedure usually requires large numbers of training data and a correct choice of the probabilistic model, both of which are hard to achieve in many real-world scenarios. Insufficient training data and an incorrect modeling assumption often yield an unreliable classifier. The discriminative training procedure is developed to minimize the probability of error rather than to approximate a true (correct) distribution function. The general procedure of the discriminative training principle works as follows: During the training stage, the system evaluates each given pattern (example feature vector) and optimizes the system parameters in response to the precise outcome of the system, i.e., taking both the true class label and the system decision into account. Our work in this dissertation focuses on the discriminative training procedure rather than the ML training procedure

because true (correct) probabilistic models for the translation examples are usually unavailable.

The research objective of this dissertation is to *develop novel statistical pattern recognition techniques for retrieval-based MT systems*. Our retrieval-based MT system is a variation of an EBMT system with a sentential translation unit. We achieve the research objective by utilizing discriminative training for a classifier design so that the classifier can better identify the similarities and differences among different sentence pairs. Our developed MT system works as follows: First, an electronic dictionary is used to generate multiple variations of the existing sentences in the source database. Each generated sentence is associated with a cluster centered on the original sentence. Second, a classifier is trained with these generated sentences to discriminate one cluster from another. We have focused on improving the classifier design so that our translation system has a reliable matching procedure. Finally, a novel learning strategy is developed to enable incorporation of human judgment into the matching procedure. The addition of human judgment is expected to improve the quality of the matching procedure because this information can be used to augment conventional knowledge sources such as a word dictionary. In summary, our work in this dissertation can be divided into three key components:

- To incorporate information from an electronic dictionary into the matching procedure, we develop a feature generation strategy to obtain a set of sensible variations of source text in the database. These sensible variations allow us to obtain clusters for the source texts in the database and to obtain a better feature normalization procedure. Performance of the retrieval-based MT systems can be improved because the input query matching can be executed against a cluster of examples rather than against a single example.
- To improve the quality of the query matching procedure, we design statistical

pattern classifiers for the cluster of examples. For the design of pattern classifiers, we develop a model regularization principle that seeks a balance between the minimization of error and of the error variability. This strategy allows us to obtain classifiers with smaller error variations and better generalization capability.

- To incorporate human judgments in addition to a word dictionary, we construct a relevant feedback learning framework based on a multi-level relevant performance measure. Performance of retrieval-based MT systems can be improved because this additional information allows more relevant sentences to be ranked higher in the retrieval list.

The rest of this chapter is organized as follows: Section 1.1 reviews existing natural language processing techniques and machine translation principles. Section 1.2 provides relevant background information of pattern recognition techniques.

1.1 Relevant Natural Language Processing Techniques

This section provides an overview of natural language processing techniques used in the following chapters. Section 1.1.1 describes techniques used to obtain the feature vectors from the perspective of pattern recognition. Section 1.1.2 describes the text categorization problems, and Section 1.1.3 reviews several main paradigms for the machine translation problem.

1.1.1 Semantic Feature Construction

The semantic feature construction procedure is used to convert text documents to feature vectors. The feature vectors can then be used in a probabilistic modeling procedure. This feature construction procedure can be separated into several steps, i.e., dictionary-construction, term-counting, term-selection, term-weighting, document-length normalization, and term-extraction steps.

The dictionary-construction and the term-counting steps are usually performed at the same time. First, a dictionary is constructed from a set of training documents. Then, the occurrences for each term in the dictionary in each document are recorded. In other words, each document has the terms as its features, and each feature takes on an integer value accounting for the number of times that a particular term occurs in the document. This kind of representation is called the “bag-of-words” document representation. A set of morphological transformations may be applied to the dictionary by discarding words with little semantic content (stop-words removal), merging words with similar roots (word-stemming), and joining word sequences as one term (word-collocations).

The most widely used term-selection technique in the text categorization field is based on mutual information ¹. Let $\mathcal{C} = \{C_1, \dots, C_M\}$ be a random variable representing M class labels, and Ψ be a random variable indicating the absence or the presence of a j^{th} term \mathbf{t}_j in a document. For example, the random variable $\psi_j = 0$ indicates the absence of the term \mathbf{t}_j , whereas $\psi_j = 1$ indicates the presence of the term \mathbf{t}_j , i.e., $\psi_j \in \{0, 1\}$. The mutual information is defined as the difference between the entropy of the class variable $H(\mathcal{C})$ and the entropy of the class variable conditioned on the absence or the presence of the word $H(\mathcal{C}|\Psi)$ [26, 71, 116]:

$$\begin{aligned}
 IG(\mathcal{C}, \mathbf{t}_j) &= H(\mathcal{C}) - H(\mathcal{C}|\Psi) \\
 &= - \sum_{i=1}^M P(C_i) \log P(C_i) + \sum_{\psi_j \in \{0,1\}} P(\psi_j) \sum_{i=1}^M P(C_i|\psi_j) \log P(C_i|\psi_j) \\
 &= \sum_{i=1}^M \sum_{\psi_j \in \{0,1\}} P(C_i, \psi_j) \log \left(\frac{P(C_i, \psi_j)}{P(C_i)P(\psi_j)} \right). \tag{1}
 \end{aligned}$$

In practice, text documents often have different numbers of words. To account for this issue, term-frequency inverse-document-frequency (**tf – idf**) normalization is carried out to obtain term-weighting and document-length normalization. The

¹The term selection technique is also called information gain (IG) in the text categorization literature.

tf – idf normalization indicates the relevance of a term in a document based on both term frequency (**tf**) and inverse document frequency (**idf**) quantities. The **tf** score indicates the relevance of a word to a document. The more a word appears in a document, the higher the **tf** score is. On the other hand, the **idf** score, which measures how infrequent a word is in a collection, is computed using the whole training text collection. If a word rarely occurs in the text collection, then the word is considered to be representative for those documents. Thus, the more a word appears in the text collection, the lower its **idf** score is.

The **tf – idf** normalization can be mathematically defined as follows: Suppose that our database contains a text collection $\mathcal{D} = \{d_1, \dots, d_N\}$ of N documents and a dictionary $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_T\}$ with T words. Neglecting the order of words in each document, one may represent the collection of documents as a $T \times N$ co-occurrence matrix $A = \{a_{ij}\}$, where a_{ij} denotes the number of occurrences of the word \mathbf{t}_i in the document d_j . That is, each document is represented in the word space by the j^{th} column vector of the matrix A . Many variants of **tf – idf** term weighting are available in the natural language processing literatures [63, 97, 114]. This research uses the following common variant.

$$\begin{aligned} \mathbf{tf} - \mathbf{idf}_{ij} &= \mathbf{tf}_{ij} \times \mathbf{idf}_i \\ &= \left(\frac{a_{ij}}{\sum_k a_{kj}} \right) \times \log \left(\frac{N}{|\{d_j : \mathbf{t}_i \in d_j\}|} \right), \end{aligned} \quad (2)$$

where $|\{d_j : \mathbf{t}_i \in d_j\}|$ is the number of documents containing the word \mathbf{t}_i , and $\mathbf{tf} - \mathbf{idf}_{ij}$ is the i^{th} row and j^{th} column entry of a new co-occurrence matrix \hat{A} .

The term-extraction can be obtained using a topic model. The assumption of the topic model is that words in documents may have multiple meanings (polysemy) or that several different words have the same meaning (synonymy). The topic model also implies that hidden topics can be extracted from a bag-of-words document representation. Three widely used topic models include Latent Semantic Analysis (LSA) [27],

Probabilistic Latent Semantic Analysis (PLSA) [51] and Latent Dirichlet Allocation (LDA) [6]. These topic models share one common principle — the choice of words in the generation of a document is independent given the topic (aspect) variables.

To reveal the semantic spaces, the LSA technique projects documents onto low-rank rather than full-rank subspaces [27]. The singular value decomposition (SVD) technique can then be used to decompose the co-occurrence matrix \hat{A} as $\hat{A} = U\Sigma V'$, where Σ is the diagonal matrix containing the singular values of \hat{A} , U and V are the orthonormal matrices consisting of eigenvectors from the word space and the document space, respectively, and V' indicates the transposition of V . By retaining the K largest singular values in Σ_K and the corresponding eigenvectors in \tilde{U} and \tilde{V} , \hat{A} is approximated as $\tilde{A} = \tilde{U}\Sigma_K\tilde{V}'$, which represents the K -most latent ensembles of words and documents.

The LSA technique also uses the projection procedure for all test documents. Given a test document d_j in the word space of the matrix A , one can apply the **tf** – **idf** normalization to the test document to obtain \hat{d}_j of the co-occurrence matrix \hat{A} . The corresponding representation of the document in the latent semantic space \tilde{A} is obtained from the transformation $\tilde{d}_j = \Sigma_K^{-1}\tilde{U}'\hat{d}_j$. Moreover, representing a term in the semantic space \tilde{A} , one can follow the same principle, i.e., the projection $\tilde{\mathbf{t}}_i = \Sigma_K^{-1}\tilde{V}'\hat{\mathbf{t}}_i$, where $\hat{\mathbf{t}}_i$ represents the term \mathbf{t}_i after **tf** – **idf** normalization.

The Probabilistic Latent Semantic Analysis (PLSA) technique was introduced as a probabilistic version of the LSA technique [51]. The PLSA technique adopts the aspect model to represent co-occurrence data associated with topic or latent variables $z_k \in Z = \{z_1, z_2, \dots, z_K\}$ and uses the Maximum Likelihood (ML) principle to estimate the model parameters. The graphical model of a PLSA model is shown in Figure 1. The PLSA technique assumes that a document d_j is selected with the probability $P(d_j)$ and an aspect z_k is selected from the conditional probability $P(z_k|d_j)$.

Each word \mathbf{t}_i is selected according to the conditional probability $P(\mathbf{t}_i|z_k)$. Training corpus \mathcal{D} consists of word-document pairs (\mathbf{t}_i, d_j) collected from N documents $d_j \in \{d_1, d_2, \dots, d_N\}$ with a vocabulary T words $\mathbf{t}_i \in \{\mathbf{t}_1, \dots, \mathbf{t}_T\}$. The joint probability of the word-document pairs (\mathbf{t}_i, d_j) can be defined as

$$p(\mathbf{t}_i, d_j) = P(d_j) \sum_{k=1}^K p(\mathbf{t}_i|z_k) p(z_k|d_j), \quad (3)$$

where $p(\mathbf{t}_i|z_k)$ is the word distribution in the topic z_k , $p(z_k|d_j)$ is the topic distribution in the document d_j , and $P(d_j)$ is the prior probability of the document d_j .

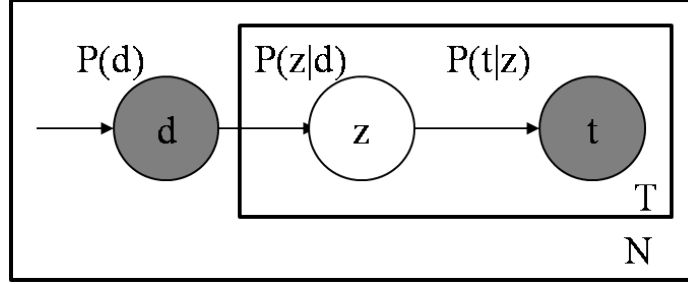


Figure 1: Graphical model for the probabilistic latent semantic analysis (PLSA) technique for a collection with N documents and T -word lexicon as listed in the bottom-right corner. Shaded and non-shaded variables indicate observed and latent variables, respectively.

The parameters of the PLSA model, which are the conditional probability distributions $P(\mathbf{t}_i|z_k)$ and $P(z_k|d_j)$, are estimated by maximizing the likelihood of the observed documents in the collection \mathcal{D} . Mathematically, the estimation procedure can be defined as

$$\Lambda_{ML} = \arg \max_{\Lambda} \log p(\mathcal{D}|\Lambda), \quad (4)$$

where $p(\mathcal{D}|\Lambda)$ indicates the log-likelihood of the training corpus \mathcal{D} , which is defined as

$$\log p(\mathcal{D}|\Lambda) = \sum_{i=1}^T \sum_{j=1}^N \text{count}(\mathbf{t}_i, d_j) \log p(\mathbf{t}_i, d_j), \quad (5)$$

where the number $\text{count}(\mathbf{t}_i, d_j)$ represents the occurrence of word \mathbf{t}_i in the document d_j , and the prior probability of a document $P(d_j)$ is usually assumed to be equal for all documents.

Because the latent variable z_k is embedded in the likelihood function, an Expectation Maximization (EM) algorithm can be applied to solve the incomplete data problem [28]. The auxiliary function of the EM algorithm is defined as

$$Q(\hat{\Lambda}|\Lambda) = E_z[\log P(\mathcal{D}, Z|\hat{\Lambda})|\mathcal{D}, \Lambda] \\ \propto \sum_{i=1}^M \sum_{j=1}^N \text{count}(\mathbf{t}_i, d_j) \sum_{k=1}^K P(z_k|\mathbf{t}_i, d_j) \log[\hat{P}(\mathbf{t}_i|z_k) \hat{P}(z_k|d_j)],$$

In the EM technique, the log-likelihood never decreases, i.e. $Q(\hat{\Lambda}|\Lambda) \geq Q(\Lambda|\Lambda)$ and $p(\mathcal{D}|\hat{\Lambda}) \geq p(\mathcal{D}|\Lambda)$. The initial values for $p(\mathbf{t}_i|z_k)$ and $p(z_k|d_j)$ can be set to have uniform or random distributions [51]. In a few words, the EM algorithm alternates between an expectation and a maximization step. For the PLSA algorithm, the maximization step (M-step) in the EM algorithm are defined as

$$p^{(n+1)}(t_i|z_k) = \frac{\sum_{j=1}^N \text{count}(\mathbf{t}_i, d_j) p^{(n)}(z_k|\mathbf{t}_i, d_j)}{\sum_{j=1}^N \sum_{i=1}^T \text{count}(\mathbf{t}_i, d_j) p^{(n)}(z_k|\mathbf{t}_i, d_j)}, \\ p^{(n+1)}(z_k|d_j) = \frac{\sum_{i=1}^T \text{count}(\mathbf{t}_i, d_j) p^{(n)}(z_k|\mathbf{t}_i, d_j)}{\sum_{l=1}^K \sum_{i=1}^T \text{count}(\mathbf{t}_i, d_j) p^{(n)}(z_l|\mathbf{t}_i, d_j)}.$$

In the expectation step (E-step) of the EM algorithm, the conditional probability distribution of the latent aspect z_k is computed based on the previous estimates of the parameters. Mathematically, the expectation step (E-step) can be defined as

$$p^{(n)}(z_k|\mathbf{t}_i, d_j) = \frac{p^{(n)}(\mathbf{t}_i|z_k) p^{(n)}(z_k|d_j)}{\sum_{l=1}^K p^{(n)}(\mathbf{t}_i|z_l) p^{(n)}(z_l|d_j)}. \quad (6)$$

The outputs of the PLSA technique are the two multinomial distributions $p(\mathbf{t}_i|z_k)$ and $p(z_k|d_j)$, where $\sum_{i=1}^T p(\mathbf{t}_i|z_k) = 1$ and $\sum_{k=1}^K p(z_k|d_j) = 1$.

The technique for estimating the topic of an unseen document is described in Algorithm 1. When an unseen document d_{unseen} is given, the conditional probability $P(z|d_{unseen})$ can be estimated by a folding-in procedure [51]. By fixing the previously learned parameters $P(\mathbf{t}_i|z_k)$ in the EM algorithm, one can estimate the $P(z_k|d_{unseen})$ which by maximizing the likelihood of the new document with respect to the parameters $P(\mathbf{t}_i|z_k)$.

Algorithm 1 Folding in procedure for PLSA.

Require: $P(\mathbf{t}_i|z_k)$ from EM on \mathcal{D}

\mathcal{D}_{unseen} a set of unseen documents

- 1: **for** each unseen document d_j in \mathcal{D}_{unseen} **do**
 - 2: Initialize $P(z_k|d_j)$ randomly
 - 3: **while** EM has not converge **do**
 - 4: Perform EM technique to estimate $P(z_k|d_j)$ without re-estimate $P(\mathbf{t}_i|z_k)$.
 - 5: **end while**
 - 6: Return the final $P(z_k|d_j)$
 - 7: **end for**
-

1.1.2 Statistical Text Categorization System

Two major paradigms that can be used to design a text categorization system are the knowledge engineering paradigm and the statistical pattern recognition paradigm [91]. In general, the text categorization systems based on the knowledge engineering paradigm use more linguistic rules. Designing and collecting useful linguistics rules for natural languages is often challenging, labor intensive, and tedious [48, 49]. Statistical pattern recognition-based text categorization systems are data-driven and generally yield high performance [114, 115]. In this research, we focus on the statistical pattern recognition paradigm, since we believe that semantic ambiguity in text categorization problems can be systematically and consistently represented using probability rules.

In the statistical pattern recognition paradigm, the text categorization problem can be formulated as the task of designing a classification function $\mathbb{C} : \mathcal{D} \rightarrow \mathcal{C}$, from a set of labeled text documents $(d_j, C_i) \in \mathcal{D} \times \mathcal{C}$, where $\mathcal{D} = \{d_1, \dots, d_N\}$ is a set of documents, and $\mathcal{C} = \{C_1, \dots, C_M\}$ is a set of M predefined categories. The classification function \mathbb{C} can later be used to categorize a new document d_j to a class C_i according to a pre-defined decision rule. Figure 2 shows a conventional text categorization procedure that consists of two main steps: a feature construction step and a classifier design step. The first task extracts a set of feature vectors from training documents, and the second task extracts a classification function \mathbb{C} .

In the feature construction step, a string (also referred to as a “term” or “word”)

separated by spaces in a document is the most commonly used feature in a text categorization problem. Other possibilities include either the context of a word [23] or sparse phrases [23]. Some researchers have also tried to use either word N -grams (N ordered sequences of words) [83] or morphemes [44] as the feature. However, most of these approaches only capture dependencies among words within a window of three to five words and become inefficient when the window becomes longer. Semantic features as described in Section 1.1.1 can also be used. The topic features capture the dependencies of all the words in the training data. These topic features are usually extracted from a word document co-occurrence matrix [27, 51].

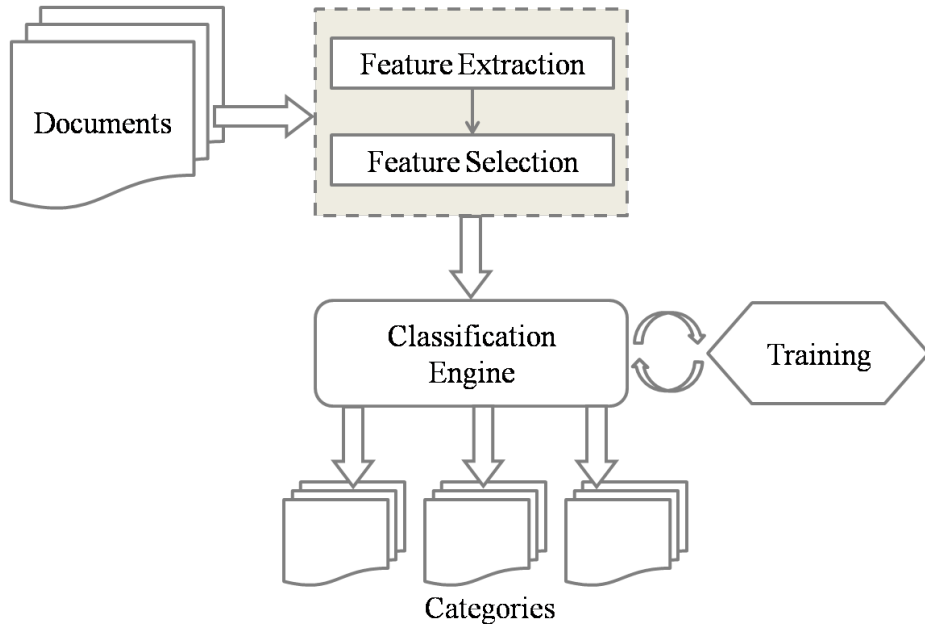


Figure 2: Conventional text categorization procedures based on the statistical pattern recognition procedure.

In the classifier design step, a learning principle is a major factor because of the relatively high number of features compared to the number of available training documents. In this study, both Empirical Risk Minimization (ERM) and Structural Risk Minimization (SRM) principles are used as classifier design principles [59, 106]. Many modern categorization systems are designed based on the SRM principle implemented in the form of the Support Vector Machine (SVM) classifier [106, 114]. The

SRM principle may overemphasize the importance of a model’s structure and under-emphasize the importance of error minimization. This dissertation uses the ERM principle because the goal is to minimize the error of future observations regardless of the classifiers’ functional forms.

1.1.3 Machine Translation Systems

Machine Translation (MT) systems are defined as the use of computational machines to analyze input text from the source language and to produce equivalent text in the target language [55]. MT systems currently provide a faster but lower quality translation than human translators. Three major paradigms are available to design a MT system: rule-based, corpus-based, and hybrid-based paradigms [24, 65, 79]. Rule-Based Machine Translation (RBMT) systems can be further grouped into three classes: direct approach, transfer approach, and interlingua approach [24]. Figure 3 illustrates the three classes in the RBMT paradigm [55]. The grouping of the RBMT systems is based on the amount of linguistic abstraction required by the translation methodology.

- In the direct approach, translation rules operate directly on the source and target sentences. Relying heavily on the dictionary, this approach mostly performs word-for-word or phrase-to-phrase translation. This approach is limited in its ability to address the problem of structural discrepancies between source and target sentences. Structural discrepancies occur because some languages are mainly subject-verb-object languages while others are subject-object-verb languages.
- In the transfer approach, translation rules operate at one level of abstraction above the word level. A source sentence is encoded in an intermediate representation, which is usually represented by the syntactic information of the

sentence. A translation procedure is then carried out by translating an intermediate representation of the source sentence to an intermediate representation of the target sentence, from which a target output sentence is then decoded.

- In the interlingua approach, translation rules operate at the semantic level between the two languages. The semantic information is a universal representation of the source sentence meaning from which the output sentence is generated in the target language.

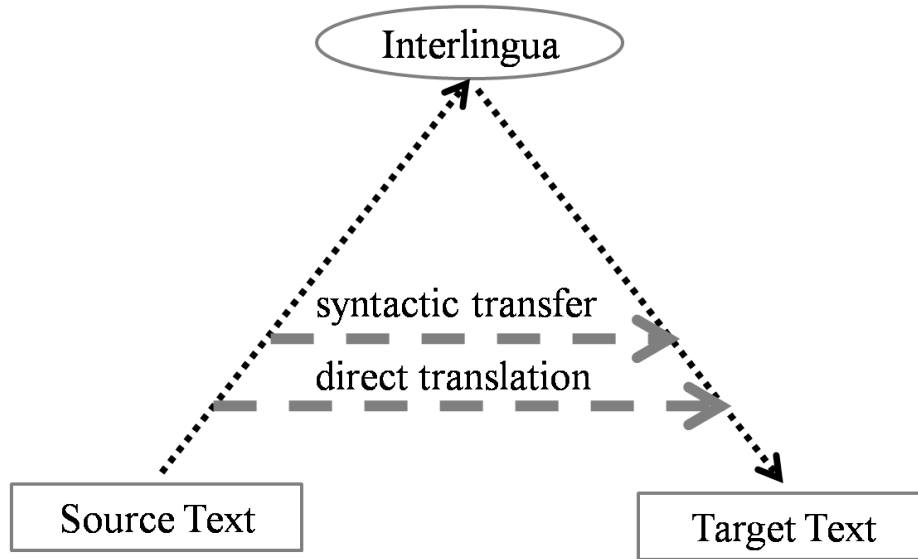


Figure 3: Grouping of rule-based machine translation systems based on the linguistic abstraction with the interlingua approach at the top, followed by the transfer-based approach, and the direct approach.

RBMT systems represent the classic approach in MT design and may encounter a knowledge-acquisition bottleneck. Improving the performance of existing RBMT systems is difficult because incorporating new rules into an RBMT system requires specialized expertise and is expensive [103]. Moreover, rule interactions are hard to understand in a big system [102].

While RBMT systems are still widely used in commercial applications, most MT research focuses on corpus-based approaches [64]. In corpus-based approaches, the

knowledge-acquisition bottleneck can be avoided because the translation knowledge is extracted automatically from translation corpora rather than being manually crafted by humans [110]. Two widely used corpus-based approaches are Example-Based Machine Translation (EBMT) and Statistical Machine Translation (SMT) systems.

The major principle of the EBMT system is to translate a source sentence by imitating the translation example of a similar sentence in the database [89]. Typical components in this paradigm can be inferred from the following quotation.

“Man does the translation, first, by properly decomposing an input sentence into certain fragmental phrases (very often, into case frame units), then, by translating these fragmental phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference.”

(Nagao,1984) [77]

This statement indicates that three major components are necessary in an EBMT system [95]. Figure 4 shows the tasks of EBMT superimposed on the Vauquois diagram [96]. The source-text analysis in conventional RBMT system is replaced by the matching the input against a set of translation examples. Once a relevant example has been identified, the corresponding fragments in the target text must be selected. This step has been termed an alignment step and is equivalent to the transfer step in the conventional RBMT system. Once the appropriate fragments are selected, the fragments can be combined to form an output target text, just as the generation stage of the conventional MT system. The parallel with a conventional MT system is reinforced by the fact that both the matching and recombination stages can, in some implementations, use techniques very similar to the analysis and generation in a conventional MT system.

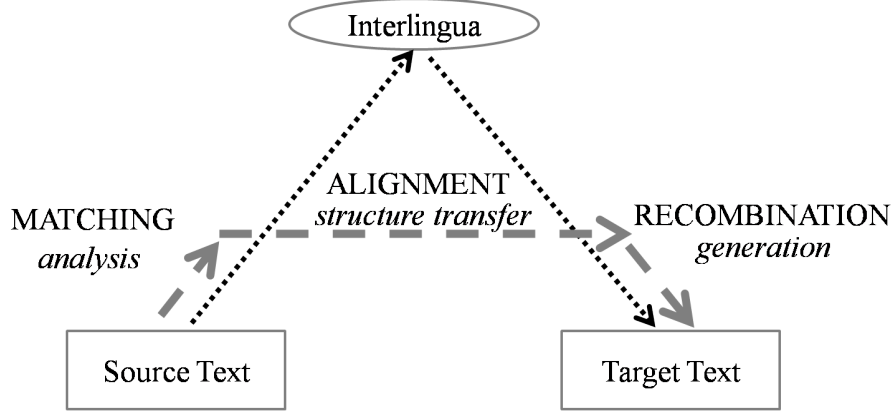


Figure 4: The three main steps in RBMT and EBMT systems. The steps in RBMT and EBMT systems are written in italics and upper case respectively.

The use of the SMT paradigm in the MT field is motivated by the success of the statistical approach in the speech recognition field. For an automatic translation of a source input sentence $e_1^I = e_1 \cdots e_i \cdots e_I$ into a target output sentence $f_1^J = f_1 \cdots f_j \cdots f_J$. The statistical approach to MT problems is formulated as a decision problem: given a source sequence e_1^I , the optimal translation is the target sequence f_1^J that maximizes the posterior probability $P(f_1^J | e_1^I)$.

$$\hat{f}_1^J = \arg \max P(f_1^J | e_1^I) . \quad (7)$$

According to the Bayes rule, this posterior probability can be divided into two probabilities: the target language model probability $P(f_1^J)$ and the translation model probability $P(e_1^I | f_1^J)$. The translation model probability describes the correspondences between the words in the source and the target sequence. The language model probability describes the word order of the target language. The best translation is found by maximizing the translation and language model probabilities. Figure 5 shows a system architecture of a conventional SMT system using the language model and the translation model.

The main advantage of the SMT paradigm is that it can be automatically trained from a parallel corpus and, thus, can be easily adapted to a new language. Recent advances in the SMT paradigm are concerned with the choice of translation units

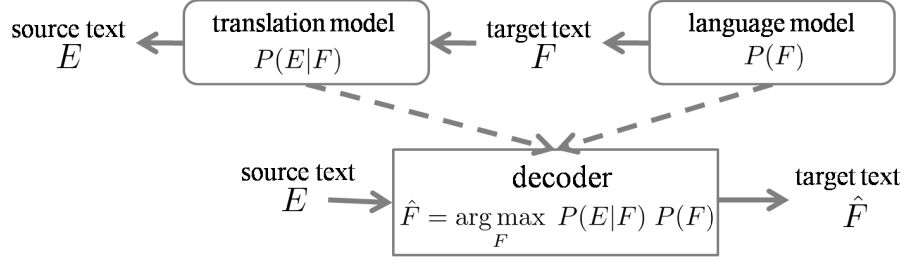


Figure 5: Architecture of a conventional SMT system using a language model and a translation model.

and the choice of linguistic model combinations. Phrase rather than word pairs are chosen as the translation units in state-of-the-art SMT systems [65, 79]. In the phrase-based SMT systems, sequences of source words (source phrases) are translated into sequences of target words (target phrases) [64]. These phrases are simply sequences of words extracted using statistical methods from the parallel corpus and are not necessarily phrases in the linguistic sense [13].

1.2 *Relevant Pattern Recognition Techniques*

This dissertation deals with supervised learning problems. Supervised learning problems assume the availability of a set of training observations x_n along with their associated class labels C_n . The main objective of Supervised learning problems is to obtain a set of discriminant functions, which are then used to predict the class labels of test observations. This section introduces advanced pattern recognition principles to deal with supervised learning problems. Section 1.2.1 and Section 1.2.2 provide reviews of the minimum classification error principle and the support vector machine classification, respectively.

1.2.1 Minimum Classification Error Principle

The Minimum Classification Error (MCE) principle is an optimal classifier design principle formulated as an alternative to the distribution estimation paradigm. The conventional classifier design approach estimates the distribution of the data without

specifically knowing if a pattern is classified correctly or not. Classifier design based on the MCE principle directly evaluates the recognition system using the specific response of the system to each given pattern.

In practice, the MCE principle offers a framework to combine the classification decision rule and the classifier performance into a novel objective function so that the system performance can be evaluated and optimized. The general steps in the implementation of the MCE principle can be explained in a step-by-step manner [38]: (i) specification of the ultimate objective function, (ii) description of the decision rule, (iii) formulation of the optimization objective function, (iv) incorporation of the decision rules into the optimization performance measure, (v) formulation of the differentiable performance measure, and (vi) optimization procedure. The general steps in the MCE principle are explained in detail in the remaining portion of this section.

(i) Specification of the ultimate objective function. The MCE principle is an optimal classifier design principle according to the Bayes decision theory. According to the Bayes decision theory, an optimal classifier is a classifier that minimizes the expected loss, which is defined as follows:

$$L = \int_{\mathcal{X}} R(\mathbb{C}(X)|X) p(X) dX , \quad (8)$$

where $\mathbb{C}(X)$ indicates the classification decision and assumes a value of one of the M classes, i.e., $\{C_1, C_2, \dots, C_M\}$, based on a random observation X drawn from a probability density function $P(X)$. The conditional risk R for the decision $\mathbb{C}(X) = C_i$ can be defined as follows:

$$R(C_i|X) = \sum_{j=1}^M \epsilon_{ij} P(C_j|X) , \quad (9)$$

where $P(C_j|X)$ is the posterior probability for class $P(C_j|X)$, and ϵ_{ij} indicates the loss incurred after misclassifying observation X from class C_j as class C_i . The expected

loss in (8) can be rewritten as follows:

$$L = \sum_{i=1}^M \int_{\mathcal{X}} \epsilon_{ij} P(C_j, X) dX, \quad (10)$$

where $P(C_i, X)$ denotes the joint distribution function of the class identity and the observations. Thus, the ultimate objective of the MCE principle is to minimize the expected loss as defined in (10).

(ii) **Description of the decision rule.** Classification, in a loose sense, can be defined as a process of mapping from one space to another, i.e. from a feature space to the space of class labels [104]. The decision rule is defined as the rule to assign a class label to every feature vector presented to the classifier. It can be shown that the expected loss (10) is minimized when the classifier $\mathbb{C}(X)$ applies the following decision rule [31].

$$\begin{aligned} \mathbb{C}(X) = C_i \text{ if } i &= \arg \min_k R(C_k|X) \\ &= \arg \min_k \sum_{j=1}^M \epsilon_{kj} P(C_j|X), \end{aligned} \quad (11)$$

where $P(C_j|X)$ is the posterior probability for class C_j , and the loss for an observation X belonging to a class C_j can be summarized as

$$\epsilon_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases} \quad (12)$$

Using this zero-one loss, the decision rule in (11) for X belonging to a class C_j can be rewritten as

$$\begin{aligned} \mathbb{C}(X) = C_i \text{ if } i &= \arg \min_k \sum_{j=1, j \neq k}^M P(C_j|X) \\ &= \arg \max_k P(C_k|X), \end{aligned} \quad (13)$$

where $P(C_k|X)$ is the posterior probability of class C_k . This decision rule is usually referred to as the *maximum a posteriori* (MAP) decision rule. In a practical scenario,

the *a posteriori* probability $P(C_k|X)$ is unknown and has to be estimated from the training data. For consistency, we use a discriminant function $g_k(X; \Lambda)$ parameterized by Λ to represent the *a posteriori* probability $P(C_k|X)$. The classification rule defined in (13) can be written as follows:

$$\mathbb{C}(X) = C_i \text{ if } i = \arg \max_k g_k(X; \Lambda), \quad (14)$$

where $g_k(X; \Lambda)$ is the discriminant function for class C_k .

(iii) **Formulation of optimization objective function.** The optimization objective function is used to approximate the ultimate objective function in (10) because the true posterior probability $P(C_k|X)$ is unknown in practice [72] and has to be estimated from a limited number of training data $\{x_{11}, \dots, x_{21}, \dots, x_{mn}, \dots, x_{MN_m}\}$ where x_{mn} represents the n^{th} training data in class C_m , M indicates the number of classes, and N_m indicates the number of training data for class C_m . The optimization objective function is basically an estimation of the expected loss in (10) using the provided training data. The optimization objective function is defined as follows:

$$\tilde{L} = \frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{n=1}^{N_m} \epsilon_{im} \mathbf{1}[x_{mn} \in C_m]. \quad (15)$$

Estimation of the posterior probability $P(C_m|X)$ is challenging for two reasons. The first reason is the lack of the true knowledge of the probability distribution of the data as required in the Bayes formulation for an optimal classifier. The second reason is that the amount of data for deriving statistical knowledge may be insufficient. Several approaches to address these issues are described in Chapter 2.

(iv) **Incorporation of decision rules into the optimization performance measure.** A new optimization objective function can be obtained by incorporating the decision rules (14) into the optimization objective function (15), and can be written as

$$\hat{L}(\Lambda) = \frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{n=1}^{N_m} \mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)] \mathbf{1}[x_{mn} \in C_m]. \quad (16)$$

The approximation in (16) uses two indicator functions. The indicator function $\mathbf{1}(\cdot)$ is used to indicate the membership of an element in a set, i.e., it assumes the value of 1 if the argument is true and 0 otherwise. The first indicator function $\mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)]$ represents the classifier decision rule in (14). The second indicator function $\mathbf{1}[x_{mn} \in C_m]$ means that the observation x_{mn} belongs to class C_j .

(v) **Formulation of the differentiable performance measure.** The next challenge is to convert the non-continuous objective function into a smooth continuous function so that it can be optimized. A misclassification measure is introduced for the training data as follows:

$$d(x_{mn}; \Lambda) = -g_m(x_{mn}; \Lambda) + \log \left[\frac{1}{M-1} \sum_{j=1, j \neq m}^M \exp[g_j(x_{mn}; \Lambda)\eta] \right]^{1/\eta}, \quad (17)$$

where η is a design parameter. As the parameter $\eta \rightarrow \infty$, the misclassification measure becomes

$$d(x_{mn}; \Lambda) = -g_m(x_{mn}; \Lambda) + \max_{j, j \neq m} g_j(x_{mn}; \Lambda). \quad (18)$$

This quantity is the difference between the true class discriminant function and the combination of the discriminant functions from other classes. For x_{mn} from class C_m , $d(x_{mn}; \Lambda) \leq 0$ indicates a correct classification, and $d(x_{mn}; \Lambda) > 0$ indicates an incorrect classification.

In theory, the misclassification measure $d(x_{mn}; \Lambda)$ cannot represent a zero-one value because its numerical value ranges from $-\infty$ to $+\infty$. A smooth zero-one function is defined in terms of the misclassification measure $d(x_{mn}; \Lambda)$, i.e.,

$$\ell(x_{mn}; \Lambda) = \ell(d(x_{mn}; \Lambda)) = \frac{1}{1 + \exp[-\xi(d(x_{mn}; \Lambda) + \alpha)]}, \quad (19)$$

where $\ell(x_{mn}; \Lambda)$ is a smooth zero-one function and both ξ and α are the design parameters. The contribution of ξ and α to the learning process can be observed in Figure 6.

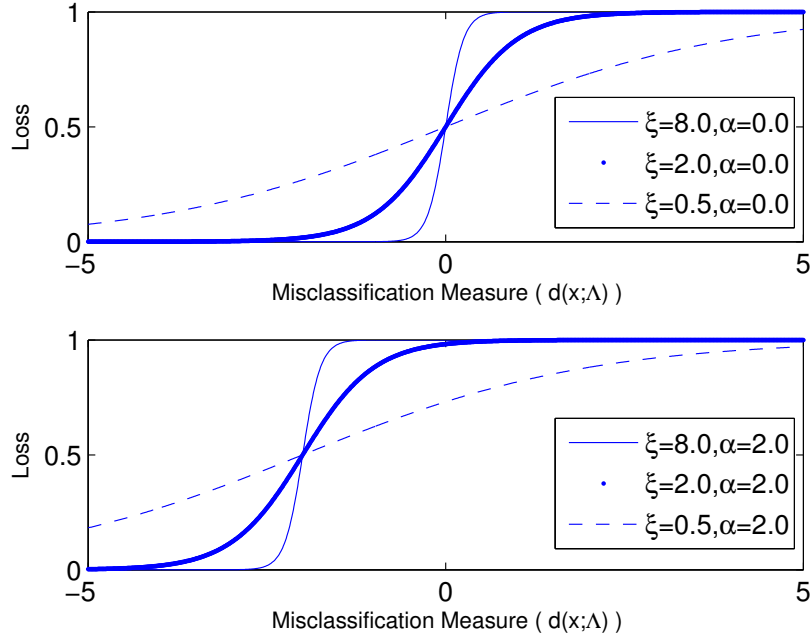


Figure 6: Examples of the smooth zero-one loss functions.

Figure 6 shows how the parameter ξ contributes to the speed of the transition from 0 to 1. The parameter α contributes to the shift of the misclassification measure from the origin. Note that the larger ξ is, the faster the transition is from 0 to 1. Thus, the data with $\ell(x_{mn}; \Lambda) = 0$ is classified correctly, and the data with $\ell(x_{mn}; \Lambda) = 1$ is classified incorrectly. The observations with $\ell(x_{mn}; \Lambda) = 1$ or $\ell(x_{mn}; \Lambda) = 0$ do not contribute to the learning. As shown in Figure 7, the gradient of $\ell(x_{mn}; \Lambda)$ is non-zero only when $0 < \ell(x_{mn}; \Lambda) < 1$. Thus, only the observations with $0 < \ell(x_{mn}; \Lambda) < 1$ contribute to the learning procedure because the training data is located in the decision boundary.

Finally, the continuous error rate that is suitable for gradient descent optimization is written as

$$\hat{L}(\Lambda) \approx \frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{n=1}^{N_m} \ell(x_{mn}; \Lambda) 1[x_{mn} \in C_m]. \quad (20)$$

The discriminant functions in (18) can be chosen according to the task at hand. In this work, we have used both parametric and non-parametric mixture models as the

discriminant functions for the probabilistic models.

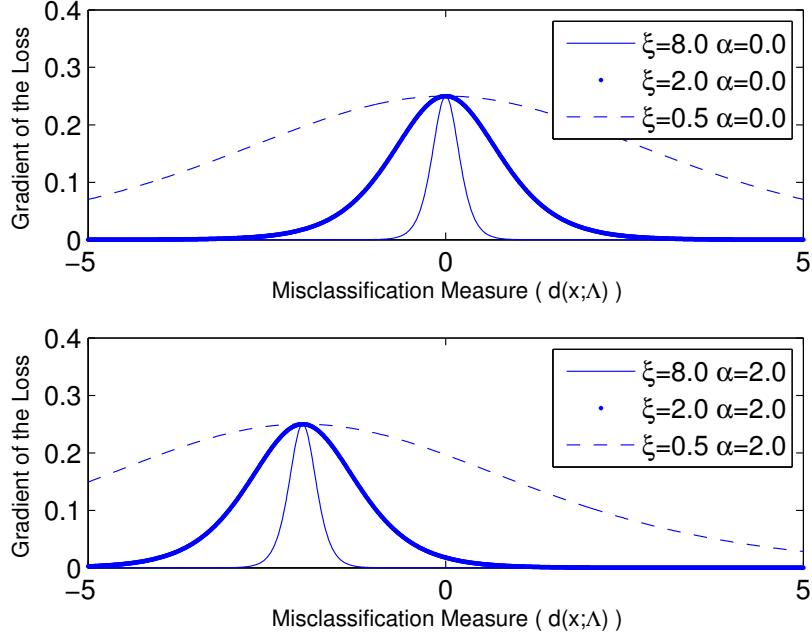


Figure 7: Illustration of the gradients of the smooth zero-one loss functions.

(vi) **Optimization procedure.** The Generalized Probabilistic Descent (GPD) algorithm [61] can be employed to minimize the empirical loss in (20). Let t denote the epoch index for parameter adjustment upon presentation of a training pattern, and the GPD algorithm iteratively modifies the parameter values according to the following update equation

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon^{(t)} \nabla \ell(X; \Lambda)|_{\Lambda=\Lambda^{(t)}}, \quad (21)$$

where $\ell(X; \Lambda)$ is defined as in (19). The algorithm will converge to a solution Λ^* almost surely (with probability one) when a number of conditions are met. The convergence property of the method can be found in [60, 22]. The update equation of ((21)) can also be generalized into

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon^{(t)} U^{(t)} \nabla \ell(X; \Lambda)|_{\Lambda=\Lambda^{(t)}}, \quad (22)$$

where U_t is a positive definite matrix [22] for speeding up the rate of convergence.

Other theoretical properties of the probabilistic descent algorithm under the name of stochastic approximation can be found in [7, 30, 86]. The descent algorithm in (22) is an unconstrained optimization scheme and has to satisfy the given constraints in many applications, some due to the underlying statistical model and some due to the physical process in the application.

1.2.2 Support Vector Machine Classification

The problem which drove the initial development of Support Vector Machine (SVM) classifiers is an overfitting phenomenon, which is well known in the pattern recognition literatures [43, 76]. The overfitting phenomenon indicates a circumstance when the performance of a pattern classifier on the training data does not reflect the classifier performance on the test data. Thus, the classifier performance on the test data cannot be accurately predicted and high-performance classifiers cannot be reliably obtained. The SVM classifier is a hyperplane classifier trained in such a way that its performance on the test data is within a bound with a certain probability. The theoretical foundation for the bound is explained in statistical learning theory. This section reviews the underlying principle of the SVM training algorithm according to statistical learning theory. Exhaustive discussions of statistical learning theory are widely available in the pattern recognition literatures [14, 106].

The training algorithm for SVM classifiers was first developed for a restricted case of separating training data into two classes without errors [10, 107]. The training algorithm is designed to find a hyperplane classifier, which is defined by a weight vector w and a bias b . The input to the training algorithm is a set of N examples $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$ with the class label $y_n = +1$ if $x_n \in C_1$ and $y_n = -1$ if $x_n \in C_2$. The training data is linearly separable if a weight vector w and bias b can be found so that the following inequalities are valid for all training

data.

$$\begin{cases} w \cdot x_n + b \geq +1 & \text{if } x_n \in C_1 \\ w \cdot x_n + b \leq -1 & \text{if } x_n \in C_2. \end{cases} \quad (23)$$

The inequality constraints in (23) are used to identify training data with a correct classification decision and to identify the training data in the class boundaries. Training data are located on the class boundaries if they satisfy either one of the equality constraints in (23). In other words, the training data x_n is located on the class boundary if $w \cdot x_n + b = +1$ for $x_n \in C_1$ or $w \cdot x_n + b = -1$ for $x_n \in C_2$. Thus, the separation between the two classes is proportional to $2/\|w\|$. The SVM training algorithm is concerned with maximizing the separation between hyperplanes by minimizing $\|w\|^2$ and increasing the generalization of the classifiers according to statistical learning theory [14].

The training algorithm for SVM classifiers was designed to automatically tune the capacity of the classification function by maximizing the margin of the training examples closest to the class boundary [10]. The basic assumption of the algorithm is that the generalization capability of a pattern classifier can be estimated when the capacity of the classification function is comparable to the size of the training data. Classifiers with a large capacity in their classification function are likely to perform without errors on the training data but are very likely to exhibit a poor generalization capability. Conversely, classifiers with insufficient capacity in their classification function might not be able to perform well on the training data at all. In between, there is a classifier with an optimal capacity and a good generalization capability.

Two important factors in the formulation of an SVM classifier are the capacity of the classification function and the number of errors in the training data. The number of errors in the training data can be approximated from the inequality constraints in (23). For example, zero training error is accomplished if all the training data satisfy

the inequality constraints. Moreover, the capacity of the hyperplane classifier is proportional to the norm of the weight vector $\|w\|$. Thus, one possible interpretation of the SVM training algorithm is to obtain a hyperplane classifier with the smallest norm of the weight vector and with the smallest number of training errors. Mathematically, the training algorithm for an SVM classifier can be stated as follows:

$$\min_{w,b} \frac{1}{2} \|w\|^2, \quad (24)$$

under the following constraints

$$\begin{cases} w \cdot x + b \geq 1 & \text{if } x_n \in C_1 \\ w \cdot x + b \leq -1 & \text{if } x_n \in C_2. \end{cases} \quad (25)$$

A standard implementation of the training algorithm is based on a quadratic optimization problem with inequality constraints, which is a well-studied optimization problem.

A useful extension to the original SVM training algorithm is the introduction of a tradeoff coefficient to reduce the capacity of the classification function at the expense of small classification errors in the training data [25]. The new formulation can still be solved as a quadratic optimization problem and has the same main objective as the original formulation of classification. That is, to identify a hyperplane classifier with a good generalization capability. Other worthwhile improvements to the SVM training algorithm include the improvements to incorporate non-linear classification functions and the improvements to solve the problem of multi-class classification problems. Non-linear classification functions can be incorporated into the SVM training algorithm by using a kernel trick [10]. The kernel trick is usually used to map the training data into a high dimensional space where a hyperplane can be used to do the separation between classes with low classification error rates. The resulting algorithm is similar to the conventional SVM training algorithm except that every dot-product in the classification function is replaced by a non-linear kernel function [90]. The

multi-class classification problems using the SVM classifiers can be solved by using two approaches. The first approach is to combine several binary classifiers and the second approach is to simultaneously consider all categories at once [53]. To combine several binary classifiers, one may use either “one-against-one” or “one-against-many” strategies. In the “one-against-one” strategy, an SVM classifier is constructed for every pair of categories. In the “one-against-many” strategy, training data from the target category is considered to be the positive class and training data from other categories as the negative class. Thus, an SVM classifier can be constructed for each positive class.

1.3 Organization of the Dissertation

The organization of this dissertation is illustrated in Figure 8. Several advanced strategies to design pattern classifiers for the clusters of textual sentences are examined in Chapter 2. Several strategies are developed to improve the generalization capability of the pattern classifiers designed from limited training data. Specifically, model selection and model regularization strategies are developed for mixture-based and margin-based classifiers. The effectiveness of the developed strategies is then demonstrated in several pattern recognition and text categorization tasks.

The incorporation of subjective preference or judgment into statistical models through the use of an error-cost learning procedure are investigated in Chapter 3. The subjective judgment can be used in addition to the traditional lexical knowledge source such as an electronic dictionary. Specifically, we develop a feedback training procedure based on a novel performance measure for document retrieval tasks. Experiments on document retrieval tasks offer new insights into how to improve the performances of the conventional natural language processing systems.

An overview of a retrieval-based MT system are presented in Chapter 4. The main advantage of the retrieval-based MT system is that the output sentence in the

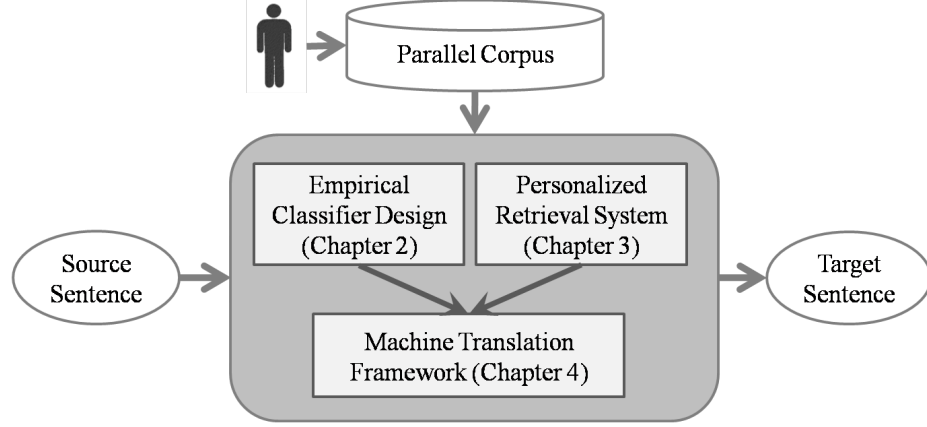


Figure 8: An overview of the dissertation.

target language is guaranteed to be meaningful or correct because those sentences are directly retrieved from the database. The main disadvantage of the retrieval-based MT system is the limited coverage of the examples in the database, i.e. an input query rarely matches exactly with the examples in the database. In our MT design, sensible variations of the source texts are automatically generated to improve the coverage of the retrieval-based MT systems. These sensible variations are centered on the original translation examples and are kept in the database. When an input query text is presented, the query text is matched against the cluster of text instead of being matched against a single text. Our experiments on the retrieval-based MT system also use the pattern recognition techniques developed in Chapter 2 and Chapter 3. Finally, in Chapter 5, we summarize the contributions and conclusions of this research and discuss possible avenues of future work.

CHAPTER II

EMPIRICAL PATTERN CLASSIFIER DESIGNS

A retrieval-based machine translation system uses pattern recognition techniques to extract a word occurrence feature vector and to identify a matching example during the translation process. Chapter 4 describes in detail the design of a retrieval-based MT system. The main drawback of retrieval-based MT systems is their limited coverage, which makes it difficult to locate exact matching between the input query text and examples in the database. In this dissertation, the coverage of the proposed MT system is improved by generating sensible variations of original translation examples. These additional examples are clustered along with the original translation examples in the database. This chapter focuses on the design of statistical pattern classifiers to represent the clusters of translation examples. The proposed translation system uses the pattern classifiers to assign a class label to an input query text. The target text corresponding to the assigned class label is then presented as the output target translation.

This chapter investigates several design issues to obtain a high-performance statistical pattern classifier. The theoretical foundation for designing a statistical pattern classifier is the Bayes decision theory, which focuses on the probability of error or the expected error rate. Despite years of research effort, obtaining the minimum probability of error (the Bayes risk) is still not possible in many real-world applications for a number of reasons. One of the key reasons is the lack of true knowledge of the probability distribution of the data as required in a Bayes formulation for an optimal classifier. Another reason is that, in practice, it may not be possible to obtain sufficient data to derive the necessary statistical knowledge.

Three different classifiers are studied to address the issue of unknown probability distribution functions and the issue of insufficient training data problems. Specifically, Kernel Density Estimation (KDE)-based classifiers are examined to address the issue of the unknown probability distribution function. Model selection and regularization principles are investigated to deal with the insufficient training data problem for mixture and hyperplane classifiers, respectively. Both mixture and hyperplane classifiers are commonly used for speech and natural language processing [67]. In real world scenarios, some classifiers are more suitable for a specific application than others depending on the computational resources and the intended task.

The rest of this chapter is organized as follows: Section 2.1 examines KDE-based classifiers to address the issue of distribution function form. Section 2.2 investigates a mixture model selection strategy for a mixture model classifier. Section 2.3 describes a study of a model regularization strategy for a hyperplane classifier. Finally, Section 2.4 summarizes our study on empirical pattern classifier design.

2.1 Empirical Mixture Model

Kernel Density Estimation (KDE)-based classifiers, which belong to a class of estimators called non-parametric density estimators, are commonly used in the pattern recognition field. The KDE-based classifiers smooth out the contribution of each training datum over a local neighborhood using a kernel function [67]. It has been shown in [39] that the estimated density function converges to the true density function as the data size approaches infinity under certain conditions. However, existing KDE classifiers attempt only to optimize the density estimation instead of attempting to minimize the expected error-rate, which is the final objective of an optimal classifier design.

In this section, we propose a KDE-based classifier with the objective of minimizing the expected error rate. Specifically, we convert a non-continuous error rate objective

function into a continuous function using the MCE principle explained in Section 1.2.1. We name the proposed classifier the Empirical Mixture Model (EMM) classifier because the functional form of the classifier assumes that of a mixture model. Our experimental results show that the conventional KDE classifiers do not perform well when the data size is small. This is our motivation to study the EMM classifier. To the best of our knowledge, there is no existing work in the literature that estimates parameters of KDE-based classifiers with the objective of minimizing the empirical error rate.

The rest of this section is organized as follows: Section 2.1.1 introduces a functional form of the EMM classifier. Section 2.1.2 provides a gradient descent algorithm to optimize the expected error-rate as the objective function. Our experimental results demonstrate the effectiveness of the proposed EMM classifier in Section 2.1.3.

2.1.1 Kernel Functional Form

The functional form of the EMM classifier is defined as a multivariate Parzen window estimator [93], which can be written as

$$\hat{f}_m(x) = \frac{1}{N_m} \sum_{n=1}^{N_m} \left\{ \prod_{k=1}^K \frac{1}{h_{mnk}} \mathcal{N} \left(\frac{x_k - x_{mnk}}{h_{mnk}} \right) \right\}, \quad (26)$$

where N_m is the training data size for class C_m , K denotes the number of dimensions, \mathcal{N} represents a Gaussian kernel function, and h_{mnk} represents the n^{th} kernel bandwidth for class C_m in dimension k . The training data provides information on K , N_m , and x_{mnk} . However, the bandwidth parameters h_{mnk} cannot be obtained directly from the training data and have to be estimated. This estimation is crucial to the classifier performance and has to be carried out carefully. To estimate bandwidth parameters of the EMM classifier, we use a corrective bandwidth learning algorithm, which is explained in detail in Section 2.1.2. Existing bandwidth selection algorithms do not work well when the sample size is small [46]. We postulate that the developed corrective bandwidth learning algorithm (the EMM classifiers) works well for both

large and small sample sizes. In this study, three bandwidth selection algorithms are used: Silverman's rule-of-thumb (ROT) [93], maximum likelihood cross-validation (MLCV) [47], and the nearest neighbor estimator (NNE) [68] techniques.

Silverman [93] proposed using a Gaussian kernel as a reference distribution to approximate the underlying density function f_m . Using this reference distribution, the optimal h_m for class C_m is found using the following rule of thumb.

$$h_{mk} = \left\{ \frac{4}{(D+2)N_m} \right\}^{1/(D+4)} \min \left(\sigma_{mk}, \frac{IR}{1.34} \right), \quad (27)$$

where IR indicates the inter-quantile range, D denotes the number of dimensions, σ_{mk} is the empirical standard deviation for dimension k , and N_m denotes the number of observations for class C_m .

The maximum likelihood cross-validation (MLCV) technique is based on the idea of minimizing the Kullback-Leibler information between two density functions [47]. The estimation of the likelihood function is obtained from leaving one observation out at a time. Given a set of independent observation x_{mn} for class C_m , the bandwidth h_m is chosen based on the likelihood function. The bandwidth parameter can be computed as

$$h_{mk} = \arg \max_{h_{mk}^{-n}} \frac{1}{N_m} \sum_{n=1}^{N_m} \log \tilde{f}_{mk}^{-n}(x_{mn}), \quad (28)$$

where $\tilde{f}_{mk}^{-n}(x_{mn})$ indicates the density function estimated by leaving out the observation x_{mnk} .

Both ROT and MLCV fall into the category of fixed bandwidth KDE. There are situations in which variable bandwidths are preferred, specifically, when the underlying distributions exhibit multi-modality with differences in scale for each mode [68]. In nearest neighbor estimation (NNE), the bandwidth h_{mn} of the n^{th} mixture is defined as follows:

$$h_{mn} = 1/Z_m \sum_{o=1}^{\sqrt{N_m}} \text{dist}(x_{mn}, x_o), \quad (29)$$

where $dist$ indicates the Euclidean distance between two training data, and Z_m is a normalizing factor which is set to be the sum of all the distances. In other words, the bandwidth h_{mn} of mixture n is selected to be proportional to the average Euclidean distance of $\sqrt{N_m}$ nearest neighbor observations from class C_m .

2.1.2 Corrective Bandwidth Learning Algorithm

The corrective learning algorithm is concerned with accurate estimation of the bandwidth parameters of the kernel functional form in (26). The objective of the algorithm is to minimize the expected error rate of the classifier and is derived using the MCE principle explained in Section 1.2.1. For the purpose of completeness, we rewrite some of the main equations of the MCE principle here. The optimization objective function is specified by (15) and is rewritten as follows:

$$\tilde{L} = \frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{n=1}^{N_m} \epsilon_{im} \mathbf{1}[x_{mn} \in C_m], \quad (30)$$

where i is the index of the most possible category. For the corrective bandwidth learning algorithm, the differentiable objective function (20) is used and is rewritten as follows:

$$\hat{L}(\Lambda) \approx \frac{1}{\sum_{m=1}^M N_m} \sum_{m=1}^M \sum_{n=1}^{N_m} \ell(x_{mn}; \Lambda) \mathbf{1}[x_{mn} \in C_m]. \quad (31)$$

where $\ell(x_{mn}; \Lambda)$ is a smooth zero-one function in (19) and is rewritten as

$$\ell(x_{mn}; \Lambda) = \ell(d(x_{mn}; \Lambda)) = \frac{1}{1 + \exp[-\xi(d(x_{mn}; \Lambda) + \alpha)]}, \quad (32)$$

where ξ and α are the design parameters, which have to be chosen based on the validation datasets.

For clarity, we denote $\ell(x_{mn}; \Lambda)$ as ℓ_n , and ℓ_n means the loss of n^{th} training pattern. The misclassification measure in (17) is used and is defined as

$$d(x_{mn}; \Lambda) = -g_m(x_{mn}; \Lambda) + \log \left[\frac{1}{M-1} \sum_{j=1, j \neq m}^M \exp[g_j(x_{mn}; \Lambda)\eta] \right]^{1/\eta}, \quad (33)$$

where $g_m(x_{mn}; \Lambda)$ is the discriminant function for class C_m , and x_{mn} is the n^{th} training data for class C_m . For clarity, we represent $d(x_{mn}; \Lambda)$ using d_n , and d_n means the misclassification measure for the n^{th} training pattern. For the corrective bandwidth learning algorithm, we set the discriminant function $g_m(x_{mn}; \Lambda)$ as

$$g_m(x_{mn}; \Lambda) = \log(\hat{f}_m(x_{mn}; h_m)), \quad (34)$$

where $\hat{f}_m(x_{mn}; h_m)$ is the kernel function defined in (26). For clarity, we denote $g_m(x_{mn}; \Lambda)$ as g_m , and g_m represents the discriminant function for class C_m . For corrective bandwidth learning, only the bandwidth parameters $\{h_m\}_{m=1}^M$ of the kernel functions are updated.

All bandwidth parameters $\{h_m\}_{m=1}^M$ have to stay positive throughout the corrective learning process. Thus, the following parameter transformation is conducted on all bandwidth parameters. For the n^{th} bandwidth from class C_m dimension k , the transformation uses the log function as follows: $h_{mnk} \rightarrow \tilde{h}_{mnk}$, where $\tilde{h}_{mnk} = \log(h_{mnk})$. To optimize the objective function, we use the GPD algorithm, which can be written as

$$h_m^{(t+1)} = h_m^{(t)} - \epsilon^{(t)} \nabla \ell_n^{(t)}, \quad (35)$$

where t indicates the learning iteration, $\epsilon^{(t)}$ is the learning step, and $\nabla \ell_n^{(t)}$ is the gradient of the loss defined in (32). The gradient is computed using the following formula for the training pattern $x_{mn} \in C_m$.

$$\frac{\partial \ell_n}{\partial d_n} = \ell_n(1 - \ell_n), \quad (36)$$

$$\frac{\partial d_n}{\partial g_j} = \begin{cases} -1 & \text{if } C_j = C_m \\ \frac{\exp(g_j \eta)}{\sum_{k=1, k \neq n}^M \exp(g_k \eta)} & \text{if } C_n \neq C_m \end{cases}, \quad (37)$$

$$\begin{aligned} \frac{\partial g_j}{\partial \tilde{h}_{jnk}} &= (\hat{f}_j)^{-1} \frac{1}{N_j} (2\pi)^{-K/2} \left(\prod_{k=1}^K h_{jnk} \right)^{-1} \exp \left\{ -\frac{1}{2} \sum_{k=1}^K \left(\frac{x_k - x_{jnk}}{h_{jnk}} \right)^2 \right\} \\ &\quad \left[\left(\frac{x_k - x_{jnk}}{h_{jnk}} \right)^2 - 1 \right]. \end{aligned} \quad (38)$$

After the learning process takes place, the following inverse transformation is applied as follows: $\tilde{h}_{mnk} \rightarrow h_{mnk}$, where $h_{mnk} = \exp(\tilde{h}_{mnk})$, to obtain new bandwidth parameters for the EMM classifiers.

2.1.3 Experimental Results

Two sets of experiments are conducted to demonstrate the effectiveness of the corrective bandwidth learning algorithm in the context of classification problems. The first set of experiments is carried out on multivariate multi-class datasets. The second set of experiments is carried out on a set of standard machine learning datasets. In addition to the proposed EMM classifiers, the conventional classifications using the KDE and SVM classifiers are also performed on the machine learning datasets.

The first set of experiments is designed with two objectives. The first objective is to observe that the error rates of the classifiers approach those of the true model as the sample sizes increase. The second objective is to observe the effectiveness of the corrective bandwidth learning while minimizing the expected error rates over different sample sizes. The two-dimensional datasets are generated randomly for three classes. Each class is represented using four Gaussians, and each of the Gaussians has a different mean, covariance, and weight. The setup for the three classes is shown in Figure 9. The sample sizes for all classes differ from the range of 16 (2^4) to 1024 (2^{10}). For each sample size, the data is then split into a training set and a testing set with a ratio of 80 to 20 percent.

For each sample size, the error rate is computed for the classifiers learned using three conventional KDE algorithms, namely, ROT, MLCV, and NNE. Figure 10 denotes the average error rates of these KDE-based classifiers. It can be observed from Figure 10 that when sample sizes are smaller than 128 (2^7), the average error rates of the conventional KDE-based classifiers are substantially different from those of the true models. However, these differences are less noticeable for the sample sizes greater

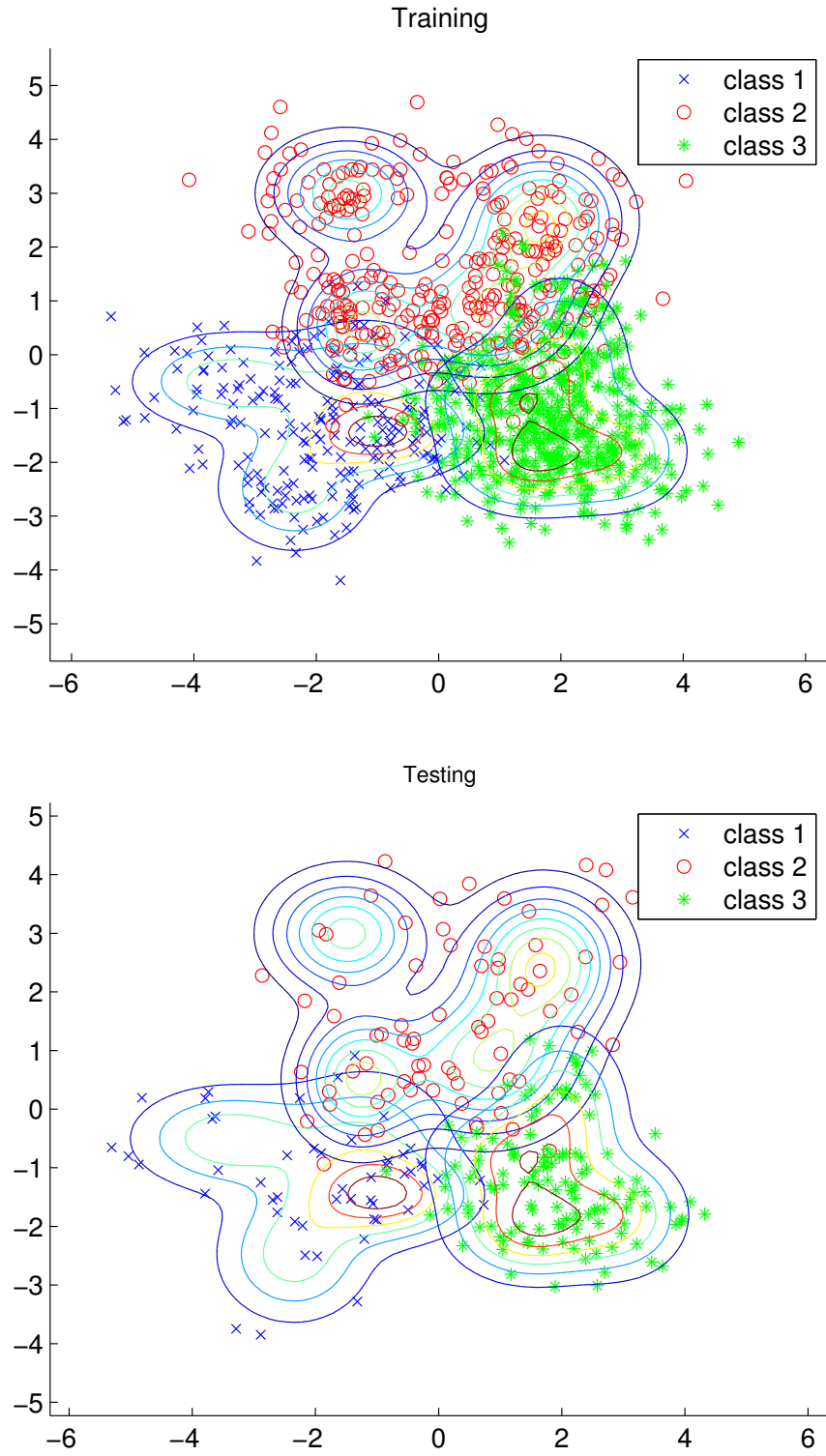


Figure 9: Examples of the training and testing samples used for verifying the corrective bandwidth learning algorithm.

than $128 (2^7)$. The average error rates of the test data shown in Figure 10 indicate more variations compared to those of the training data regardless of the sample size. Note that as the sample size increases, the average error rate decreases for both the training and testing samples. This verifies that the error rates of the EMM classifiers approach those of the true model as the sample size increases. However, the error rates of the true models and the conventional KDE models differ significantly when the sample size is small. This fact provides the motivation for the use of the corrective bandwidth learning algorithm.

The effectiveness of the corrective bandwidth learning algorithm is shown in Figure 11. In these figures, the error rates for “before” and “after” the corrective bandwidth learning are plotted against those of the true model. The classifiers used in Figure 11 are initialized using the NNE method. The improvements due to the corrective bandwidth learning algorithm are noticeable across different sample sizes, and the improvements are especially large for sample sizes smaller than (2^7) for both training and test datasets. Similar results are obtained for the classifiers obtained from the ROT and MLCV methods. These results achieve the second objective that the corrective learning algorithm improve the classification performance.

Additional experiments are conducted using the standard datasets from the UCI machine learning repository [3]. These datasets along with their specifications are listed in Table 1. Three EMM (KDE-based classifiers) and a Support Vector Machine (SVM) (a margin-based classifier) are constructed and compared for those standard datasets. Moreover, in this study, we use the One-Against-One (OAO) strategy to convert a two-class SVM classifier into a multi-class SVM classifier.

The average error-rates of those classifiers are evaluated using a standard 10-fold cross-validation. To ensure fairness, the training and the test datasets are the same for all classifiers. For the SVM classifier, as is commonly done in practice, the dataset is normalized so that the feature values are between -1 and +1. Choosing the SVM

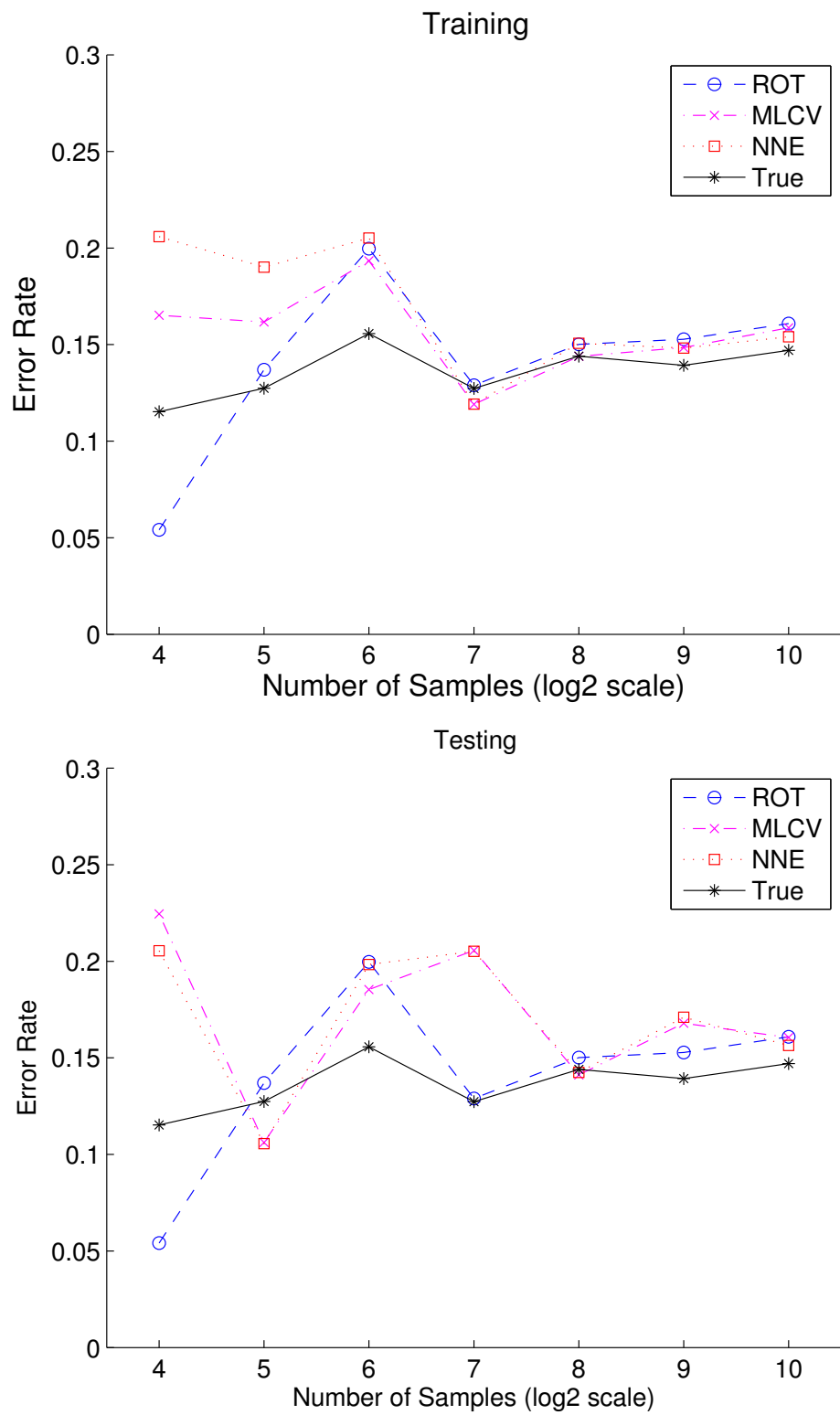


Figure 10: The average error-rates for training and testing samples using conventional KDE classifiers based on ROT, MLCV and NNE.

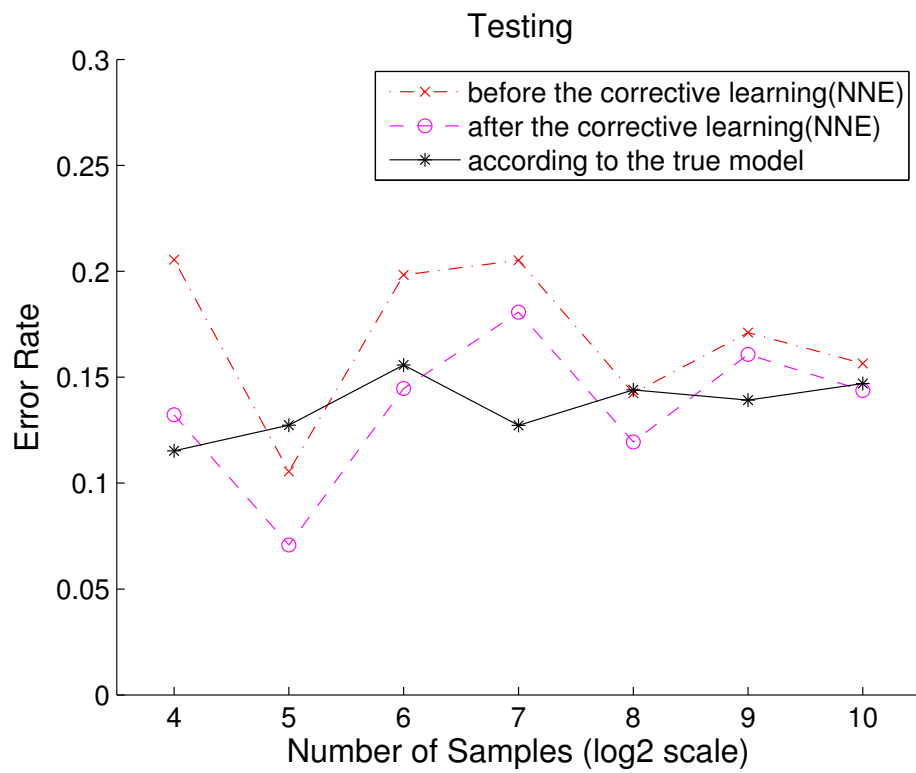
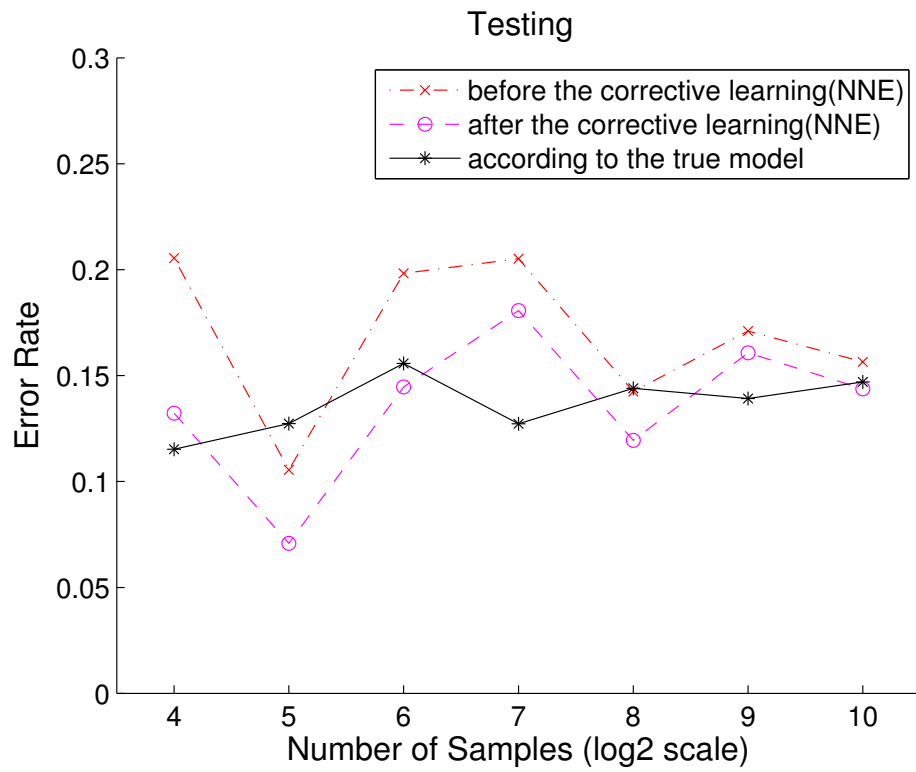


Figure 11: The average error-rates for training and testing samples decrease after application of the corrective learning algorithm.

Table 1: The descriptions of the datasets used to verify the corrective bandwidth learning algorithm.

	Num Classes (M)	Num Samples ($\sum_{i=1}^M N_i$)	Num Features (K)	$\frac{\sum_{i=1}^M N_i}{MK}$
BLD	2	345	6	28.75
Cancer	2	683	9	37.94
Ecoli	8	336	7	6.00
Glass	6	214	9	3.96
Heart	2	270	13	10.38
Ionosphere	2	351	34	5.16
Iris	3	150	4	12.50
New Thyroid	3	215	5	14.33
SPECTF	2	267	44	3.03
Vowel	11	528	10	4.80
Wine	3	178	13	4.56
Zoo	7	101	16	0.90

parameters is done by conducting a cross validation procedure with 225 possible combinations for kernel parameter γ and cost parameter \mathfrak{C} : $\gamma = [2^4, 2^3, \dots, 2^{-10}]$ and $\mathfrak{C} = [2^{12}, 2^{11}, \dots, 2^{-2}]$. For all classifiers, the best parameters (hyper-parameters) are chosen from the training data before the parameters are used to evaluate the test data.

Table 2 lists the accuracy of the conventional KDE classifiers and the SVM classifier. For a specific dataset, the highest accuracy rate is bold-faced. Table 2 indicates that for most datasets, the SVM classifier performs better than the KDE classifiers. Table 3 lists the accuracy rates of the EMM classifiers and the SVM classifier. Table 2 and Table 3 reveal that after application of the corrective bandwidth learning algorithm, the new classifier performs better.

It can be further observed from Table 3 that the classifier designed from NNE bandwidth initialization and the corrective bandwidth learning (NNE+CL) performs better on most datasets than do other classifiers. We believe the reason for the difference in performance is that the NNE approach in (29) uses more parameters

Table 2: The average accuracy rates for the conventional KDE-based classifiers based on ROT, MLCV and NNE, and the standard OAO SVM classifier.

	ROT	MLCV	NNE	SVM
BLD	61.20	55.65	61.98	70.76
Cancer	96.78	95.76	93.41	96.92
Ecoli	76.52	84.50	85.13	86.30
Glass	55.69	55.69	69.26	69.72
Heart	72.96	58.52	73.70	81.85
Ionosphere	90.60	89.17	90.02	93.16
Iris	96.00	94.00	94.67	96.00
New thyroid	94.87	96.28	94.89	96.26
SPECTF	72.74	51.37	79.45	76.08
Vowel	96.77	97.35	96.40	99.06
Wine	98.30	84.34	93.33	96.11
Zoo	95.00	79.27	82.27	94.00

than either the ROT approach in (27) or the MLCV approach in (28). Specifically, for each class C_i , the NNE approach estimates N_i bandwidths with N_i indicates the size of dataset for class C_i , the ROT approach estimates K bandwidths with K indicates the number of categories, and the MLCV approach estimates 1 bandwidth. In other words, the NNE approach requires more data to estimate the bandwidth.

Table 3: The average accuracy rates for the KDE-based classifiers with the corrective learning algorithm (the EMM classifiers), and the standard OAO SVM classifier.

	ROT+CL	MLCV+CL	NNE+CL	SVM
BLD	69.57	70.43	70.69	70.76
Cancer	97.65	96.05	94.00	96.92
Ecoli	79.77	88.39	88.78	86.30
Glass	64.61	65.97	74.85	69.72
Heart	78.52	76.29	87.77	81.85
Ionosphere	95.58	94.86	96.86	93.16
Iris	96.67	94.67	98.00	96.00
New Thyroid	98.61	98.59	99.55	96.26
SPECTF	82.45	75.00	84.71	76.08
Vowel	98.30	97.92	98.30	99.06
Wine	98.30	96.67	98.33	96.11
Zoo	95.00	83.18	86.18	94.00

Thus, we believe that estimation quality is the reason that the “NNE+CL” approach does not perform as well on “cancer” and “zoo” datasets as do the “ROT+CL” and “MLCV+CL” approaches. Table 1 lists the ratio of the size of the datasets over the number of classes times the number of features, i.e., $\frac{\sum_{i=1}^M N_i}{MK}$. This quantity indicates whether the data is large enough to estimate the classifier parameters. The sample size is adequate if this quantity has a large value. The size of the “zoo” dataset is not large enough for “NNE+CL” to obtain an accurate estimation. The size of the “cancer” dataset, however, seems to be large enough relative to the number of parameters. We believe that the poor estimation quality can be improved upon by using either a model selection or a model regularization technique.

2.2 *Classification-Based Model Selection*

Model selection procedure is a widely used strategy to address the problem of an insufficient number of training data to obtain a high-performance classifier. This section investigates the use of a model selection procedure for a Gaussian Mixture Model (GMM) classifier. Moreover, the GMM has been shown to “quickly” approximate any target density with arbitrary precision [67], mitigating problems incurred by the lack of information with respect to the true functional form. This result still holds if the mixture component is learned incrementally, i.e., starting with one mixture and adding one mixture after another. Identifying a new component to add to the model is equivalent to finding the global maximum of a log-likelihood surface. It was proposed in [67] that a grid search be performed in the parameter space. This searching strategy is not feasible in many practical applications, especially in cases with only a small number of observations. We address this problem by using a classification-based mixture selection technique.

In this section, we propose designing a Bayes classifier with the objective of simultaneously achieving good approximations of class-conditional densities and inter-class

decision boundaries. To achieve this objective, we make use of a mixture selection criterion and add mixture components incrementally. The main advantage of the criterion is its capability to use the contribution of the whole training sample to quantify the quality of a particular mixture as the inter-class decision boundaries. The use of incremental learning is later shown to allow the classifier designer to circumvent the need to specify the initial mixtures while still being able to approximate the true density function. The experimental results show that the proposed classifier has good generalization capability in addition to low complexity.

The remainder of this section is organized as follows: Section 2.2.1 introduces the conventional and discriminative mixture selection criteria. Section 2.2.2 describes the incremental learning algorithm used to obtain a low-complexity and highly discriminative classifier. Section 2.2.3 outlines experiments on a set of machine learning datasets.

2.2.1 Mixture Model Selection

For a given set of M categories or classes $\{C_m : m = 1, \dots, M\}$, the classifier task is to assign an incoming pattern X into one of the classes. Decoding based on Bayes classification theory is guaranteed to minimize the probability of error [31]. Thus, a Bayes classifier chooses the class C_m that yields the maximum value of $P(C_m|X)$ (the *a posteriori* probability of the class) given the data. In this way, the Bayes decoding is equivalent to maximizing the product of the class-conditional probability $p(X|C_m)$ and the *a priori* probability of the class $P(C_m)$. That is,

$$X \in C_m \text{ if } m = \arg \max_k p(X|C_k)P(C_k) , \quad (39)$$

where X is the input data to the system. The mixture model selection problem consists of selecting a single topology \mathbb{T}_{mq} as the sole representative of the class C_m . This process is carried out by using selection criteria such that

$$C_m \text{ uses } \mathbb{T}_{mq} \text{ if } q = \arg \max_k C(\mathbb{T}_{mk}) , \quad (40)$$

where we assume that there is a set of Q_m candidate models for each class C_m , $\{\mathcal{M}_{mq} : q = 1, \dots, Q_m\}$. Each model \mathcal{M}_{mq} includes both the model structure (topology) \mathbb{T}_{mq} and the parameter of the model Λ_{mq} . Thus, each model \mathcal{M}_{mq} implements the class-conditional probability $p(X|C_m)$ as $p(X|\mathbb{T}_{mq}, \Lambda_{mq})$.

We consider two mixture model selection approaches for a classifier design: Bayesian Information Criteria (BIC) and Discriminative Information Criteria (DIC). The major difference between the two approaches is that the BIC does not use consider the class label information, while the DIC does.

The standard criterion for choosing the number of components in each class is to use BIC [5], which assigns a score to each topology \mathbb{T}_{mq} based on the following definition.

$$BIC(\mathbb{T}_{mq}) = \log p(X; \mathbb{T}_{mq}, \Lambda_{mq}) - \beta \frac{\mathbb{K}_{mq}}{2} \log N_m, \quad (41)$$

where \mathbb{K}_{mq} is the number of free parameters and N_m is the size of the observations for class C_m . The BIC is the sum of the data log-likelihood and the term $\frac{\mathbb{K}_{mq}}{2} \log N_m$, which is a penalty of the number of free parameters in the model. The β term is used as a regularizing term and is set to be 1 in this study. To quantify the quality of a mixture to specify the inter-class decision boundary, we advocate the use of model selection based on DIC proposed by Alain Biem [5]. This criterion takes into account not only the number of free parameters in the model but also the inter-class boundaries, which are important for the classification process. This criterion is defined as follows:

$$\begin{aligned} DIC(\mathbb{T}_{mq}) = \log p(X_m; \mathbb{T}_{mq}, \Lambda_{mq}) & - \beta \frac{\sum_{j=1, j \neq m}^{M-1} \log p(X_j; \mathbb{T}_{mq}, \Lambda_{mq})}{M-1} \\ & + \frac{\mathbb{K}_{mq}}{2(M-1)} \sum_{j=1, j \neq m}^{M-1} \log \frac{N_j}{N_m}. \end{aligned} \quad (42)$$

The first term $\log p(X_m|\mathbb{T}_{mq}, \Lambda_{mq})$ is the data log-likelihood of the data, and the

second term $\log p(X_j|\mathbb{T}_{mq}, \Lambda_{mq})$ is the average of the anti-likelihood terms. The anti-likelihood of the data X_j from class C_j against the model from class C_m is a likelihood-like quantity in which the data and the model belong to competing categories. The third term accounts for the number of free-parameters in the models and equals zero if the datasets are of the same size. In this study, the regularizing constant β is set to be 1, indicating equal importance between the target class and the competing classes.

2.2.2 Incremental Mixture Learning Procedure

Conventional mixture-based classifiers use the EM algorithm to approximate the class-conditional density. The main attraction of the EM algorithm is that it guarantees that the data log-likelihood will not decrease. Nevertheless, the EM algorithm assumes that the information on the number of mixtures is available for classifier design. To avoid this difficulty, we use incremental mixture learning, in which we increase the number of mixtures one by one. Our novel classifier design technique is described in Algorithm 2 and is illustrated in Figure 12.

For each class C_m , we start the learning process from one mixture model with k is used to denote the number of mixtures in the model. The model \mathcal{M}_{mq} implementing the class-conditional probability $p(x|C_m)$ is defined as $p(x; \mathbb{T}_{mq}^{(k)}, \Lambda_{mq}^{(k)})$. In this case, a new mixture is added with density $\phi_{k+1}(x)$; a new class conditional density $p(x; \mathbb{T}_{mq}^{(k+1)}, \Lambda_{mq}^{(k+1)})$ is defined as follows:

$$p(x; \mathbb{T}_{mq}^{(k+1)}, \Lambda_{mq}^{(k+1)}) = (1 - \alpha) p(x; \mathbb{T}_{mq}^{(k)}, \Lambda_{mq}^{(k)}) + \alpha \phi_{k+1}(x) , \quad (43)$$

where $p(x; \mathbb{T}_{mq}^{(k)}, \Lambda_{mq}^{(k)})$ is the current mixture model, $\phi_{k+1}(x)$ is the new component, and α are the mixing weights for the new components with a range of values between 0 and 1 for $\alpha \in (0, 1)$. This is analogous to the incremental training procedure called Greedy-EM for unsupervised probability density estimation [108]. The most important distinction between our proposed approach and the original Greedy-EM algorithm is in the selection of the mixture components.

Algorithm 2 Incremental classifier learning algorithm

Require: K as the maximum number of mixture

P as the maximum number of candidates

for class indicated by i such that $1 \leq i \leq M$ **do**

 Start with one-mixture model computed as follows:

$$\begin{aligned}\mu_{i1} &= \frac{1}{N_i} \sum_{x \in C_i} x \\ \Sigma_{i1} &= \frac{1}{N_i} \sum_{x \in C_i} (x - \mu_{i1})(x - \mu_{i1})^T \\ w_{i1} &= 1\end{aligned}$$

while the number of mixture k is less than K **do**

 Prepare a set of randomly chosen candidates of size P .

 Apply EM update equations (45), (46), (47) to each of the candidates until convergence.

 Increase the mixture to $k + 1$ by combining the existing k mixtures with the candidates.

 Apply conventional EM update equations for all $k + 1$ mixtures.

 Compute the model selection criteria for each candidate using either (41) or (42).

end while

 Set the candidates with the highest scores as the model for class i

end for

To select a new component, we randomly obtain a set of candidate components from a data partitioning technique such as a KD-tree [45]. Each candidate component will adjust its parameter using the partial EM update, where only the parameters of a new component are updated. In the expectation step of the partial EM [108], we compute the *a posteriori* probability as follows:

$$P(k+1|x_j) = \frac{\alpha \phi_{k+1}(x_n)}{(1 - \alpha) p(x_j; \mathbb{T}_{mq}^{(k)}, \Lambda_{mq}^{(k)}) + \alpha \phi_{k+1}(x_n)}, \quad (44)$$

where $(k+1)^{th}$ is the *a posteriori* probability for the mixture. During the maximization

step, the component parameters are updated as

$$\alpha_{m(k+1)} = \frac{1}{N_m} \sum_{n=1}^{N_i} P(k+1|x_n) , \quad (45)$$

$$\mu_{m(k+1)} = \frac{\sum_{n=1}^{N_m} P(k+1|x_n) x_j}{\sum_{j=1}^{N_m} P(k+1|x_n)} , \quad (46)$$

$$\Sigma_{m(k+1)} = \frac{\sum_{n=1}^{N_m} P(k+1|x_n) (x_n - \mu_{m(k+1)}) (x_n - \mu_{m(k+1)})^T}{\sum_{n=1}^{N_m} P(k+1|x_n)} . \quad (47)$$

After applying the partial EM steps for each candidate component, we compute the mixture selection score for each candidate using either (41) or (42). These steps are then repeated for all classes.

2.2.3 Experimental Results

Experiments are conducted on UCI machine learning datasets [3] to demonstrate the performance of the proposed incremental classifier design. The number of patterns, features, and classes for those datasets have been listed in Table 1. For each dataset, the ten fold cross validation technique is used to obtain an estimate of the generalization error. Ten experiments are conducted, with one of the folds used for testing and the remaining nine folds for training. In each experiment, nine classifiers are constructed using one of the nine folds as a validation set and the remaining eight folds for the incremental training process (training sets). A combination of mixtures that provide the minimum error rate on the nine runs is selected and evaluated using the test data.

Four types of experiments are conducted in this study. The first two are based on non-incremental mixture learning (k-means algorithm) with BIC and DIC model selection criteria. The other two types of experiments use the same mixture selection criteria, but they are based on incremental mixture learning. Most conventional mixture-based classifier designs are based on a non-incremental version with BIC. The incremental version with BIC is somewhat common. The technique proposed in this study is the incremental version with DIC. The non-incremental version with

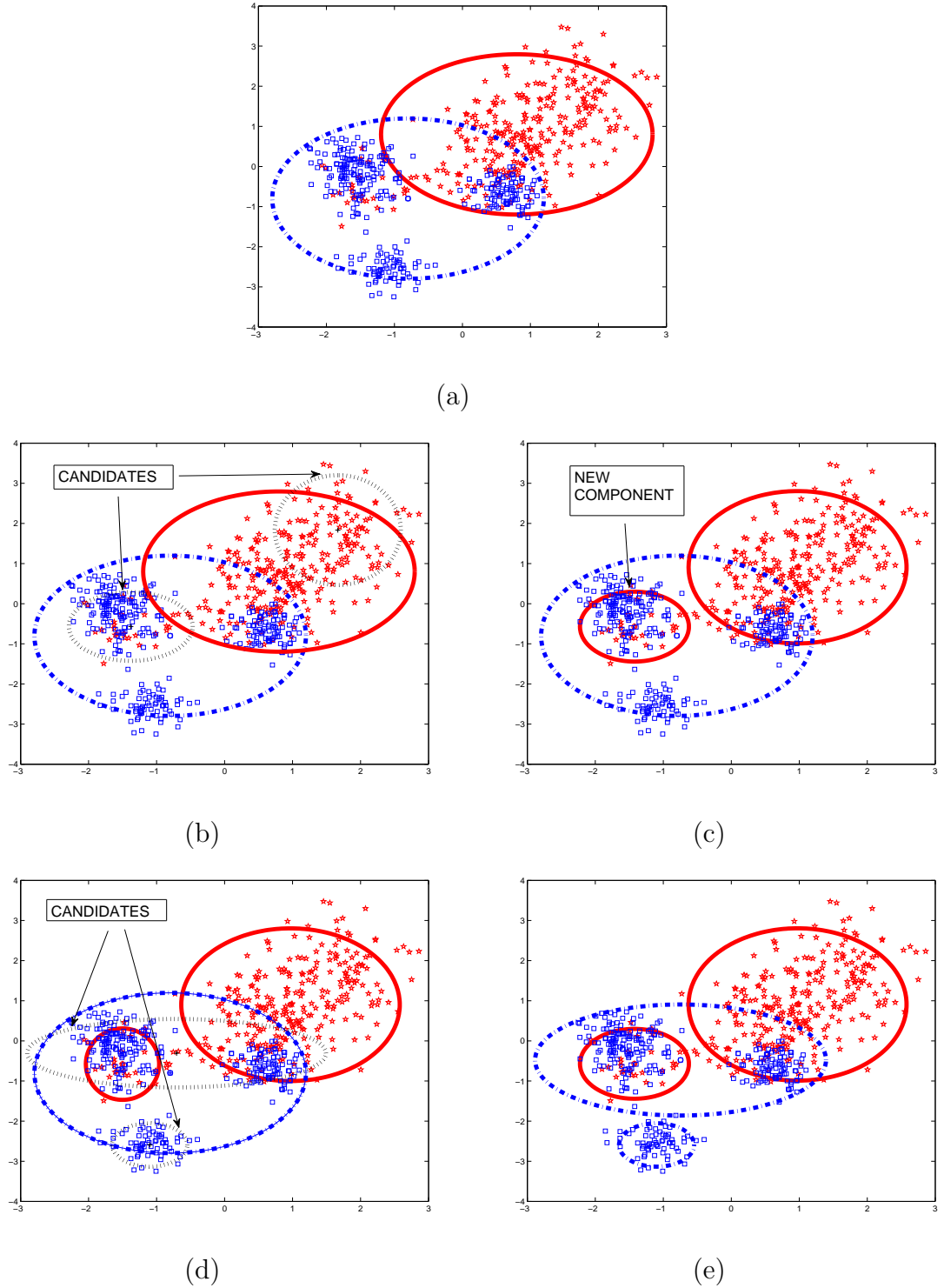


Figure 12: Incremental learning in discriminative mixture model classifiers. In (a), we start with one mixture for each class, i.e., class 1 and class 2. In (b), two candidates are proposed for the class with solid ellipsoid (class 1). In (c), the most discriminative candidate is added as the second mixture for class 1 using the Discriminative Information Criteria (DIC). In (d), another two candidates are proposed for the class with dash ellipsoid (class 2). In (e), the most discriminative candidate is added as the second mixture for class 2 using the Discriminative Information Criteria (DIC).

DIC is included for completeness. The results of this study are summarized in Table 4 and Table 5.

Table 4: The average accuracy and the standard deviation (in parentheses) of four types of mixture learning techniques using a ten-fold cross-validation procedure.

	Non-Incremental		Incremental	
	BIC	DIC	BIC	DIC
Cancer	95.3(2.0)	96.5(2.2)	95.6(7.9)	97.2(1.9)
Heart	81.0(3.7)	82.3(4.0)	80.6(2.9)	83.7(6.4)
Iris	96.7(5.0)	97.3(4.5)	96.7(4.7)	98.0(4.6)
New Thyroid	96.8(5.3)	97.8(3.3)	97.0(5.9)	97.4(3.3)
Wine	95.5(3.5)	96.5(4.1)	95.4(4.0)	96.6(3.5)
Zoo	91.7(10.2)	91.6(8.9)	91.5(9.5)	91.6(10.5)

Our experimental results in Table 4 do not support our hypothesis that models constructed using the DIC have higher accuracy than those constructed using the BIC. Our hypothesis is motivated by the following fact. The BIC for a class is computed from the model and data for that specific class, while the DIC for a class is obtained from all training data including training data that do not belong to the specific class. Moreover, the incremental and non-incremental models with BIC do not show a significant difference in terms of accuracy. Similarly, the incremental and non-incremental models with DIC do not indicate a significant difference in accuracy.

Table 5: The average and the standard deviation (in parentheses) of the number of free parameters for verifying mixture learning algorithms using ten-fold cross-validation.

	Non-Incremental		Incremental	
	BIC	DIC	BIC	DIC
Cancer	115.5(36.7)	97.7(32.4)	110.0(27.9)	66.0(30.9)
Heart	184.3(41.1)	190.0(22.9)	173(37.5)	82.0(12.7)
Iris	57.5(10.6)	58.3(10.4)	52.0(8.9)	42.0(22.7)
New Thyroid	51.0(14.7)	58.1(16.7)	58.6(10.9)	49.8(21.25)
Wine	105.0(31.2)	106.9(12.73)	102.6(7.9)	88.5(20.9)
Zoo	425.3(76.9)	435.6(47.9)	325(50.9)	224.0(60.5)

As observed from Table 5, the incremental or non-incremental models with BIC require approximately the same number of parameters to achieve comparable accuracy.

In the case of DIC, although the accuracies of the incremental models are comparable to those of the non-incremental models, the incremental models consistently require fewer parameters than the non-incremental models. We believe the reason is that the incremental models do not face the same initialization problems as those faced by the non-incremental models. Thus, coupling incremental learning with a model selection criterion is apparently an effective approach for classifier design.

Notice that the mixture structure \mathbb{T}_{mq} is selected discriminatively in this study. However, the mixture parameters Λ_{mq} are not optimized discriminatively. The mixture parameters Λ_{mq} can be further optimized by using the MCE principle, which is explained in Section 1.2.1.

2.3 Variability Regularization in Margin-Based Classifiers

The model regularization procedure is also a widely used strategy to address the problem of an insufficient number of training data to obtain a high-performance classifier. Most state-of-the-art classifiers are optimized directly based on a pre-specified performance measures, such as the training error rate or the error margins. Minimizing the error margin alone has been known to cause overfitting problems and may lower the classification performance for new unlabeled data. This is known as the generalization issue in the machine learning literature. Statistical learning theory has proposed controlling the capacity, or complexity, of a classifier’s parameters and, thus, the testing errors are bounded in a specific range. The Support Vector Machine (SVM) classifier, as a realization of the learning theory, addresses the generalization issue by constructing a hyperplane. In a class-separable case, the SVM classifier is optimized according to a constraint commensurate with its predicted generalizability based on the theory of VC dimension for a given classifier structure. In this way, a regularization characteristic can be achieved through the minimization of both the classifier’s size and the error margins.

In this study, we follow a different approach to address the generalization issue. Our approach is based on the standard bias-variance principle. There exists a close relationship between the bias and the error margin and between the variance and the variability of error margin. We introduce a straightforward idea that the regularization quantity should reflect the variability of the error margin rather than the size of a classifier’s parameters. In other words, as we adopt the idea of regularization in the classical model selection problem, we focus on minimizing the variability of the prescribed performance objective, given the training data and the model structure, instead of controlling the complexity of the model. Recent studies on margin-based classification [92, 58] have placed more emphasis on the error margin and less on the margin regularization. A discriminative information criterion (DIC) was also proposed to address the generalization issues in classification problems [5, 70]. Our work, which focuses on regularizing the error margin by using the variability of the error margin, can be used to enhance the existing margin-based classifiers. Determination of the error margin and the regularization quantity is formulated as a quadratic optimization problem, which can be solved efficiently via the existing mathematical programming algorithms [99].

In short, our design strategy includes a crucial piece of information, namely the variability of the error margin, to enhance the state-of-the-art margin-based classification. Our approach also utilizes a mathematical programming technique that keeps the strength of the existing margin-based designs. The notion of variability regularization can be easily adapted to different classification techniques. After reviewing the fundamental classification design principles in Section 2.3.1, we discuss extensions for margin-based classification with variability regularization in Section 2.3.2. Experimental results on a set of machine learning datasets are reported in Section 2.3.3.

2.3.1 Empirical Risk Minimization Principle

The design of a pattern recognition system includes many design choices such as the choice of classification decision rules, the choice of discriminant functions, the choice of parameter estimation techniques, and the choice of online/offline processing. All of these design choices and their interactions need careful assessment to obtain a pattern recognition system that generalizes well into testing datasets. In this study, we view the generalizability of a pattern recognition system as the characteristic of the system as a whole rather than as the characteristic of the individual component/procedure in the system. While most pattern recognition systems indirectly address the generalizability characteristic using, for example, the size of classification functions, we propose to directly define the generalizability in terms of the variation of the expected loss of the whole pattern recognition system. In other words, a system generalizes better than the others if the system has less variation in its performance across different datasets.

This expected loss cannot be directly computed from the training data in a practical scenario. According to the Bayes decision theory, which is the theoretical foundation for the design of a pattern recognition system, the expected loss corresponds to the generalization of a pattern recognition system. For classification into M different classes, the expected loss of a pattern recognition system is defined as

$$\mathcal{L} = \sum_{m=1}^M \int \epsilon(\mathbb{C}(X)|C_m)P(C_m|X)p(X)dX, \quad (48)$$

where $\{g_m(\cdot)\}_{m=1}^M$ denotes a set of discriminant functions, $\mathbb{C}(X)$ represents the classifier decision for assigning observation X to class C_i , $P(C_m|X)$ is the *a posteriori* probability of class C_m and $\epsilon(C_i|C_m)$ denotes the cost of misclassifying from category C_m into category C_i . The error cost is typically set to be one whenever i and m are different and zero otherwise.

An approximation of the expected loss (48) is usually used in the design of a

pattern recognition system because only a finite number of training data are available in a practical scenario. The approximation is usually called the empirical error rate, which is defined as

$$R_{emp} = \frac{1}{N} \sum_{n=1}^N 1(\mathbb{C}(x_n) \neq y_n), \quad (49)$$

where $\mathcal{N} = \{1, \dots, n, \dots, N\}$ and $\mathcal{M} = \{1, \dots, m, \dots, M\}$ denote the sets of indices for training patterns and class labels, respectively; $\{(x_n, y_n) : n \in \mathcal{N}\}$ denotes a set of labeled training patterns with a pattern $x_n \in R^D$ from a D -dimensional observation space \mathcal{X} and a label $y_n \in \mathcal{M}$. This approximation is oftentimes inaccurate and unreliable in many pattern recognition applications due to the design choices of the classifier and their interactions. The empirical error rate (49) depends on the classifier decision $\mathbb{C}(\cdot)$ regarding the available training data $\{(x_n, y_n) : n \in \mathcal{N}\}$. According to the Bayes decision theory, the expected loss in (48) is minimized if the following decision rule $\mathbb{C}(\cdot)$ is implemented.

$$\mathbb{C}(x_n) = C_m \text{ if } m = \arg \max_k P(C_k | x_n). \quad (50)$$

However, the *a posteriori* probability is not available and has to be approximated from available training data. Without loss of generality, the *a posteriori* probability is represented using a discriminant function $g_m(\cdot)$. To specify the discriminant function, one has to define its functional form and ensure that enough training data are available to estimate the parameters of the chosen functional form. Overfitting phenomenon causes inaccurate and unreliable estimation of the empirical risk in (49).

The discriminant functions in this study are obtained using the Empirical Risk Minimization (ERM) principle, where the quality of a pattern classifier is directly determined by its empirical performance on a finite number of available observations. For a given set of observations, the ERM suggests the choice of the discriminant functions $\{g_m(\cdot)\}_{m=1}^M$ for which the estimation of the empirical error rates is low with small variability across different datasets. In other words, the ERM principle actually

implies a tradeoff between an accurate and a reliable estimation of the empirical error rate. Indeed, our formulation also includes the conventional way of regularizing the empirical error rate of a classifier, i.e., using the model class of the discriminant function. Given a set of observations, one could expect a more accurate estimation using the discriminant function from a richer model class rather than one from a simpler model class. However, the discriminant function from a richer model class is usually less reliable (more variability) than those function from a simpler model class.

Alternatively, the Structural Risk Minimization (SRM) principle suggests a tradeoff between the quality of the estimation of the empirical error rate and the complexity of the discriminant functions. The SRM principle is used to control the generalization of a learning machine in statistical learning theory. For pattern recognition problems, the following bound on expected loss holds with probability $1 - \eta$ according to statistical learning theory.

$$R(\Lambda) \leq R_{emp}(\Lambda) + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}}, \quad (51)$$

where R_{emp} is the empirical risk defined in (49) and h is the VC dimension of a model class associated with the discriminant functions parameterized by Λ . The VC dimension is defined as the maximum number of points that can be separated in all possible manners by the functional form of the model class. For a set of observations $\{(x_n, y_n) : n \in \mathcal{N}\}$, the SRM principle suggests the use of a particular functional form such that the error bound in (51) is minimal.

A distinctive characteristic of the bound in (51) is the VC confidence, which is defined as the second term on the right hand side of the bound. The VC confidence is large when the VC dimension h is large and the VC confidence is small when the VC dimension h is small. The first term of the bound in (51) corresponds to the approximation of the expected loss in (49). However, the overall bounds in (51) can still be large for the small VC dimension h if the $R_{emp}(\Lambda)$ is large. To the best of our knowledge, there is no concrete result regarding the tightness of the bound in (51).

High-performance classifiers with a large VC dimension, such as nearest neighbor classifiers, are widely used in practice. Classifiers with a small VC dimension may perform poorly in practice [104].

The ERM principle is, strictly speaking, different from the SRM principle. The ERM principle is concerned with the assessment of the overall classifier performance, while the SRM principle is concerned with the risk associated with the functional form of the model class. The overall classifier performance assessed by the ERM principle has indeed incorporated the errors due to the wrong choice of the function form. Thus, the bound offered by the SRM principle is not necessary if a classifier is designed based on the ERM principle. In this work, we carefully examine the strength of the SRM in the pattern recognition design and offer one way to improve the quality of a pattern recognition system from the ERM perspective.

2.3.2 Regularized Empirical Error Margin

In this study, we propose a novel regularization procedure for a margin-based classifier to address a multi-class categorization problem. The novel regularization procedure can be formulated as follows:

- We define the discriminant function to be a linear function of the feature values. The linear discriminant function represents one of the simplest discriminant functions and has a straightforward geometrical interpretation. The decision boundary of the linear discriminant function partitions the feature space using a hyperplane decision surface.
- We specify the decision rule for multi-class categorization problems using margin-based classifiers. Most margin-based classifiers use a positive or negative sign as the decision rules because they are usually formulated for two class categorization problems.

- We define the notion of error margin as the separation between the correct class and the rest of the classes. The error margin is more involved in the multi-class than in the two-class categorization problems because of the difference in the decision rules.
- We define a hinge loss as a function of the error margin. The hinge loss is used to approximate the zero-one classification loss, i.e., the loss for correct classification is zero, while for incorrect classification, the loss is linearly proportional to the error margin.
- We define the variability of the separation margin and propose using the quantity to regularize a hyperplane classifiers. Conventional margin-based classifiers usually use the size of the classification function as the regularization criterion.
- We formulate a hyperplane classifier regularized by the variability of the error margin.

Our formulation of margin-based classifiers is different from that of the conventional margin-based classifiers because our classifier is designed from the ERM principle instead of from the SRM principle. Our formulation is novel because it directly relates the classifiers' parameter with the classification performance measure so that the classifiers' parameter can be evaluated and optimized accordingly.

In pattern classification problems, the decision rule is usually defined in terms of the *a posteriori* probability. This study represents the decision rule in terms of linear discriminant functions, which are defined as

$$g_k(x_n) = w_k x_n + b_k \quad (52)$$

where w_k is the weight vector and b_k is the offset from the origin. The linear discriminant function predicts the class membership based on a linear combination of the feature values. The linear function can be interpreted as the hyperplane $w_k x_n + b_k = 0$

in the feature space by separating the training examples of a specific class from the rest of the classes. The magnitude of the discriminant function is proportional to the distance from the pattern to the hyperplane, which is defined as $\frac{|w_k x_n + b_k|}{||w_k||}$. The decision rule for the multi-class categorization problems is defined as

$$\mathbb{C}(x_n) \in C_m \text{ if } m = \arg \max_k g_k(x_n). \quad (53)$$

The decision or classification rule assigns the class label based on the magnitude of the linear discriminant function. Some researchers have used the sign of the discriminant function in the decision rule for two-class categorization problems. However, multi-class categorization problems require more accurate estimation of the discriminant function. That is, the decision rule in the multi-class categorization problems is more sensitive to the variations of the magnitude of the discriminant function than the decision rule is in two-class categorization problems.

The notion of margin is used to indicate the degree of separation between the correct class and the rest of the classes. The margin is computed by evaluating the discriminant functions of each training pattern and is defined as

$$d_n = g_{y_n}(x_n) - g_k(x_n) \quad \forall k \in \mathcal{M} \setminus \{y_n\}. \quad (54)$$

Each training pattern with one true class will have $M-1$ separation margins. Kessler's construction technique is used to simplify the notation of the margin [34]. This technique maps the D -dimensional input space to a higher $M(D+1)$ -dimensional space to transform the non-separable problem into a linearly separable one. A training pattern x_n is expanded to a set of $M-1$ patterns, each of which is represented by the vector $\mathbf{x}_n^m \in R^{M(D+1)}$, $m \in \mathcal{M} \setminus \{y_n\}$, as follows:

$$\mathbf{x}_n^m(j) = \begin{cases} [x'_n, 1]' & j = y_n \\ -[x'_n, 1]' & j = m \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

where $j = 1, \dots, M$, $[x'_n, 1]' \in R^{D+1}$ is the augmented input vector, and $\mathbf{0} \in R^{D+1}$ is a vector with $D + 1$ zeros. For example, when $M = 4$ and $y_n = 2$, $m \in \mathcal{M} \setminus \{y_n\} = \{1, 3, 4\}$, and the corresponding vectors are

$$\begin{aligned} \mathbf{x}_n^1 &= [\quad -[x'_n, 1] \quad [x'_n, 1] \quad \mathbf{0}' \quad \mathbf{0}' \quad]' \\ \mathbf{x}_n^3 &= [\quad \mathbf{0}' \quad [x'_n, 1] \quad -[x'_n, 1] \quad \mathbf{0}' \quad]' \\ \mathbf{x}_n^4 &= [\quad \mathbf{0}' \quad [x'_n, 1] \quad \mathbf{0}' \quad -[x'_n, 1] \quad]' . \end{aligned}$$

Kessler's technique can be used to rewrite the separation margin in (54) as

$$d_n = \mathbf{w} \mathbf{x}_n^k \quad \forall k \in \mathcal{M} \setminus \{y_n\} \quad (55)$$

where the new weight vector $\mathbf{w} \in R^{M(D+1)}$ is a concatenation of the original weight vectors, i.e., $\mathbf{w} = [w'_1, b_1], [w'_2, b_2], \dots, [w'_M, b_M]'$. The margin exists whether or not the system makes the correct decision based on the decision rule (53). If the classifier makes a correct classification decision, then the discriminant function of the true class is the largest, and all of the separation margins are positive. If the classifier makes an incorrect classification decision, then at least one other discriminant function is greater than the discriminant function of the true class.

To distinguish between the margins of the correct and incorrect classification decisions, we define a classification error as a function of the margin. Ideally, the zero-one classification error should be used for pattern categorization problems. However, the zero-one classification error cannot be used during function optimization because the classification error is not a continuous function. This study uses a hinge-loss function to approximate the zero-one classification loss. The hinge loss is defined as follows:

$$\ell(x_n) = \max(1 - \mathbf{w} \mathbf{x}, 0) \quad (56)$$

In general, the hinge loss is linearly proportional to the separation margin for an incorrect classification decision and is zero for a correct classification decision. The constant one is added to the error margin so that a small loss is imposed on the training data near the decision boundary.

This formulation uses hinge loss to approximate the loss of each training pattern. The overall system performance, i.e. the empirical error rates can be written as

$$\bar{R}_{emp} = \frac{1}{N} \sum_{n=1}^N \ell(x_n) \quad \rightarrow \quad R_{emp} = \frac{1}{N} \sum_{n=1}^N 1(\mathbb{C}(x_n) \neq y_n) \quad (57)$$

To optimize the overall system performance, we write the optimization problem as follows:

$$\min_{\mathbf{w}, \xi} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M} \setminus \{y_n\}} \xi_n^m \quad (58a)$$

$$\text{s.t. } \mathbf{w}' \mathbf{x}_n^m \geq 1 - \xi_n^m \quad (58b)$$

$$\|\mathbf{w}\| = \varrho \quad (58c)$$

$$\xi_n^m \geq 0, n \in \mathcal{N}, m \in \mathcal{M} \setminus \{y_n\}, \quad (58d)$$

where ϱ is a pre-specified constant. The slack variables ξ_n^m relax the inequality constraint. The regularization constant \mathfrak{C} identifies the balance between the norm of the parameters and the error margin. The main objective in (58) is to directly minimize the error margin. The norm of the classifier functional form is not changed during the *learning* process, i.e., the model parameter ϱ is not a part of the optimization process.

This study proposes using the variability of margin as the regularization factor. The variability of the separation margin is defined as

$$\text{var}[\mathbf{w}' \mathbf{x}] = \text{E}[(\mathbf{w}' \mathbf{x})^2] - \text{E}[\mathbf{w}' \mathbf{x}]^2 \approx \mathbf{w}' \mathbf{Q} \mathbf{w},$$

where $\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$ and $\mathbf{Q} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n' - \bar{\mathbf{x}} \bar{\mathbf{x}}'$. Finally, the **proposed regularized objective** function for margin-based classifiers can then be defined as follows:

$$\min_{\mathbf{w}, \zeta, \xi} \quad \zeta + \mathfrak{C} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M} \setminus \{y_n\}} \xi_n^m \quad (59a)$$

$$\text{s.t.} \quad \mathbf{w}' \mathbf{x}_n^m \geq 1 - \xi_n^m \quad (59b)$$

$$\mathbf{w}' \mathbf{Q} \mathbf{w} \leq \zeta \quad (59c)$$

$$\|\mathbf{w}\| \leq \varrho \quad (59d)$$

$$\zeta > 0, \xi_n^m \geq 0, n \in \mathcal{N}, m \in \mathcal{M} \setminus \{y_n\}, \quad (59e)$$

where \mathfrak{C} is a regularization coefficient, ϱ is a pre-specified constant for the norm of the parameters, and the first and second-order constraints on the error margin are represented by (59b) and (59c), respectively.

Originally, SVM learning was formulated for two-class classification problems, and many extensions have been proposed for the multi-class classification case. Our formulation is comparable with conventional margin-based classifiers [34, 111]

$$\min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \sum_{m \in \mathcal{M}} (\|\mathbf{w}_m\|^2 + b_m^2) + \mathfrak{C} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M} \setminus \{y_n\}} \xi_n^m \quad (60a)$$

$$\text{s.t.} \quad g_{y_n}(x_n) - g_m(x_n) \geq 1 - \xi_n^m \quad (60b)$$

$$\xi_n^m \geq 0, n \in \mathcal{N}, m \in \mathcal{M} \setminus \{y_n\}. \quad (60c)$$

That is, the minimization of the parameter norm $\|\mathbf{w}_m\|^2 + b_m^2$ leads to the regularization of the error margin. Using the simplified notation, the **standard SVM objective** for multi-class classification problems in (60) can be rewritten as follows [34]:

$$\min_{\mathbf{w}, \xi} \quad \frac{1}{2} \|\mathbf{w}\| + \mathfrak{C} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M} \setminus \{y_n\}} \xi_n^m \quad (61a)$$

$$\text{s.t.} \quad \mathbf{w}' \mathbf{x}_n^m \geq 1 - \xi_n^m \quad (61b)$$

$$\xi_n^m \geq 0, n \in \mathcal{N}, m \in \mathcal{M} \setminus \{y_n\}. \quad (61c)$$

We note that the **proposed regularized objective** for a margin-based classifier in (59) is designed based on the ERM principle, while the **standard SVM objective** in (61) is designed based on the SRM principle. One of the drawbacks of the SVM formulation is that varying the regularization coefficient \mathfrak{C} may over-emphasize the contribution of the norm of the parameters, hence leading to over-generalization, and under-emphasize the wrong classification decision in (60b). Nevertheless, a main strength of the SVM formulation is that the error margin may be expressed as a linear function of the parameters, and thus, can be optimized using the existing mathematical programming algorithms. The proposed formulation combines such an advantage with the strength of the ERM-based classifiers.

2.3.3 Experimental Results

In the following experiments, the design of a margin-based classifier is carried out through a three-step procedure comprised of model training, model selection, and performance analysis. In the model training step, either the proposed regularized objective or the standard SVM objective criterion is used as the optimization objective value. During the model selection, the best trade-off parameter \mathfrak{C} is identified by using one of the three criteria: the proposed regularized objective, the standard SVM objective, or the empirical error rate. The performance analysis is used to identify the generalizability of a margin-based classifier using testing datasets with the empirical error rate used as the performance measure. Ideally, the model selection criterion should be the same as the optimization objective value, that is, the proposed regularized objective or, in this case, the standard SVM objective. However, our experimental results will show that the selection of \mathfrak{C} based on the empirical error rate has better generalizability and that the margin-based criteria are only approximations of the empirical error rate performance measure. To verify the new formulation, we use the well-known mathematical programming software, SeDuMi, which is a software

package for conic optimization [99].

An experiment is conducted on the UCI machine learning datasets [35], summarized in Table 1, to demonstrate the performance of the proposed incremental classifier design. For each dataset, the ten-fold cross-validation technique is used in order to obtain an estimate of the generalization error. Ten experiments are carried out with one of the folds used for testing and the remaining nine folds for training. Cross validation is conducted with 17 possible combinations of the cost parameter $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^8]$ to choose the trade-off coefficient for the SVM classifier. Four different combinations of *model training* and *model selection* scenarios are investigated for these datasets: **(A)** classifier training and selection are carried out with a standard SVM objective, **(B)** classifier training and selection are carried out with the proposed regularized objective, **(C)** classifier training is performed with a standard SVM objective and classifier selection is carried out based on the empirical error rate, and **(D)** classifier training is performed with the proposed regularized objective and classifier selection is carried out based on the empirical error rate. Figure 13 shows the empirical error rates for the four scenarios.

We observe that model selection using the empirical error rate in scenarios **(C)** and **(D)** are usually better than model selection using either the standard SVM objective in scenario **(A)** or the proposed regularized objective in scenario **(B)**. Moreover, the proposed regularized objective in scenarios **(B)** and **(D)** provides better performance than the standard SVM objective in scenarios **(A)** and **(C)**. We believe the reason is that the standard SVM objective overemphasizes the contribution of the norm of the parameters, hence over-generalizes, during the model selection step. The proposed regularized objective, on the other hand, focuses on the error margins along with their volatility and has less error variation across the trade-off coefficients \mathfrak{C} as shown in the next experiment.

Another experiment is performed using a simple example to verify our hypothesis.

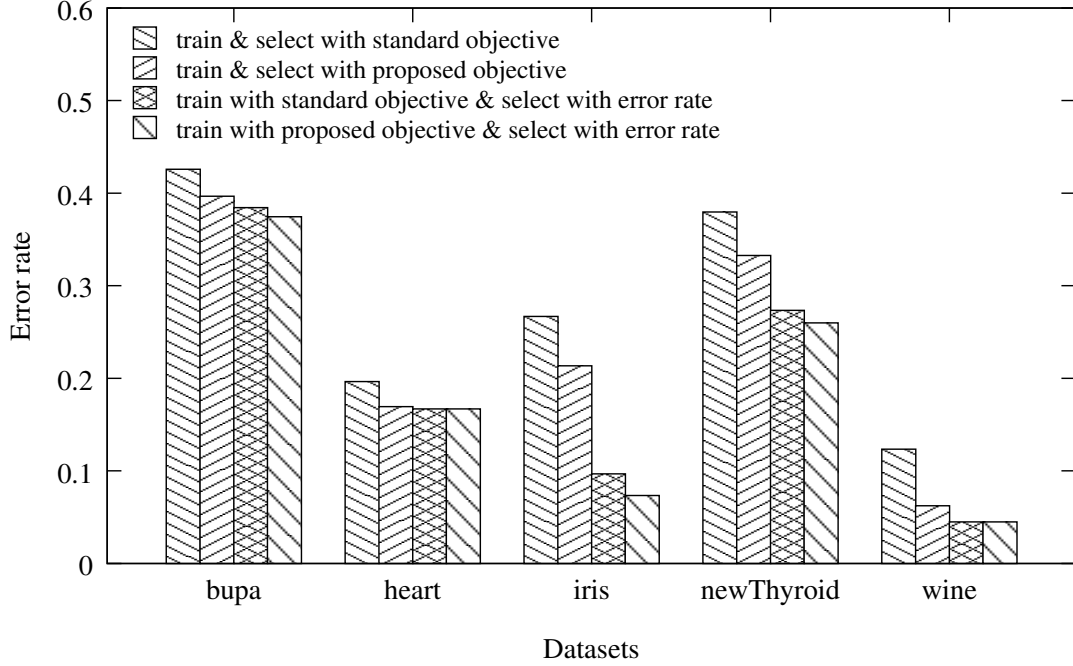


Figure 13: Empirical error rates for classifiers are designed according to the following 4 different scenarios. In scenario (A), classifier training and selection is carried out with the standard SVM objective. In scenario (B), classifier training and selection is carried out with the proposed regularized objective. In scenario (C), classifier training is performed with the standard SVM objective and classifier selection is carried out based on the empirical error-rate. In scenario (D), classifier training is performed with proposed the regularized objective and classifier selection is carried out based on the empirical error-rate. The ten-fold cross-validation experimental results lead us to claim that the proposed regularized objective used by the classifiers in scenarios (B) and (D) has lower empirical error rates than the standard SVM objective used by the classifiers in scenarios (A) and (C).

Three classes of two-dimensional data with 50 samples per class are generated for a total of 150 samples of training patterns. The scatter plot for the generated data is shown in Figure 14. Each class has only one mixture with the same mixture weight but different mean values across the classes.

Figures 15 and 16 show the standard SVM and proposed optimization objective values along with their corresponding empirical error rates for 13 trade-off parameters $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. Figure 15 and Figure 16 show the model selection procedure based on the standard SVM and the proposed variability regularized objective values.

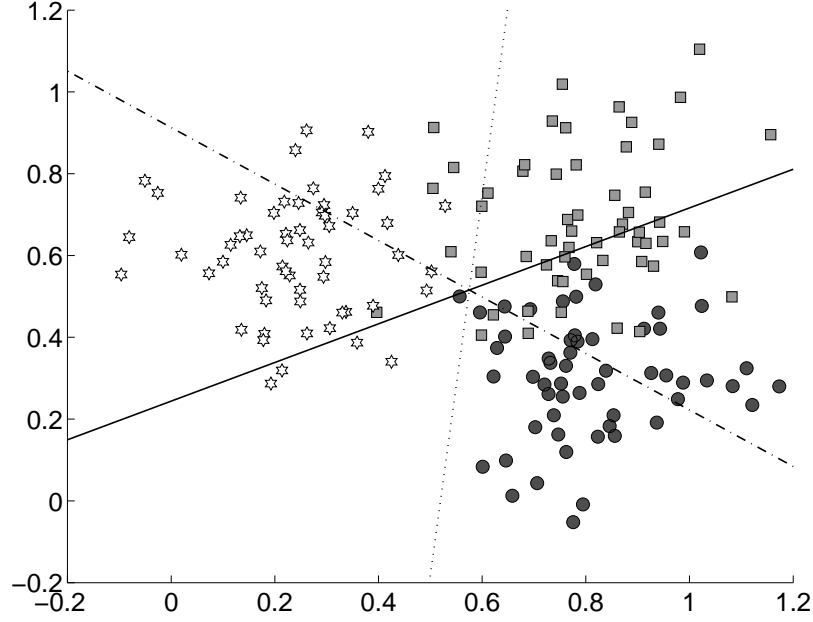


Figure 14: Scatter plot for the 3-class dataset along with decision boundaries for a margin-based classifier.

We want to identify the best trade-off value among the 13 different trade-off parameters $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. In both figures, we list two quantities: the optimization objective values and the empirical error rates.

We observe from Figure 15 that model selection using the standard SVM criterion chooses $\mathfrak{C} = 2^{-8}$, and that the empirical error rate is 0.18. This indicates that the standard SVM optimization objective value is not directly related to the empirical error rate and that a small objective value is not an indicator of a low empirical error rate. We observe from Figure 16 that the model selection using the proposed regularized criterion chooses $\mathfrak{C} = 2^{-8}$ and that the empirical error rate is 0.08. This indicates that the proposed criterion attempts to minimize the variability of the empirical error rate across different trade-off parameters and that a small objective value indicates either a low empirical error rate or low variability of the empirical error rates.

In summary, the design of a high-performance classifier is mainly concerned with the generalization of the classifier to the test data. Conventional classifier designs

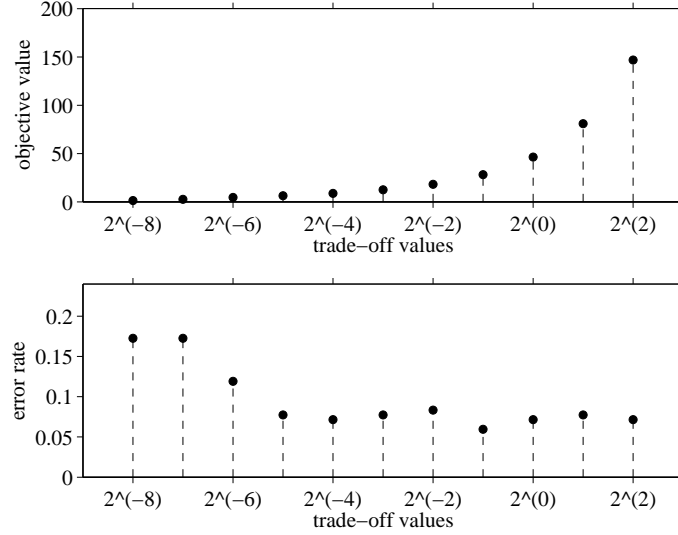


Figure 15: Model selection based on standard regularized SVM objective values for 13 trade-off parameter $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. The objective values are plotted at the top, and the corresponding empirical error rates are plotted at the bottom. Model selection using the standard SVM criterion chooses $\mathfrak{C} = 2^{-8}$ with the empirical error rate of 0.18. This indicates that the standard SVM optimization objective value is not directly related to the empirical error rate and that a small objective value does not correspond to a low empirical error rate.

indirectly address the generalization issue, for example, by using the number of parameters in the model or by using the VC dimension of the classifier’s functional form. In this study, we approach the generalization issue more directly in terms of the variation of the performance measure. Our claim is that the classifier can generalize better to the test data if there is small variation of the performance measures, which is represented by the empirical error rate. To verify this idea, we develop a regularization procedure for a hyperplane classifier using the error margins to approximate the classification error. To address the generalization issue, our regularization procedure is based on the variation of the error margins rather than the VC dimension of the classifier. Although our experimental results show promising results, formal investigations are necessary to verify the use of this regularization term. The variation of the error margins is not exactly equivalent to the variation of the empirical error rates because the former is concerned with the variation of the error margin for

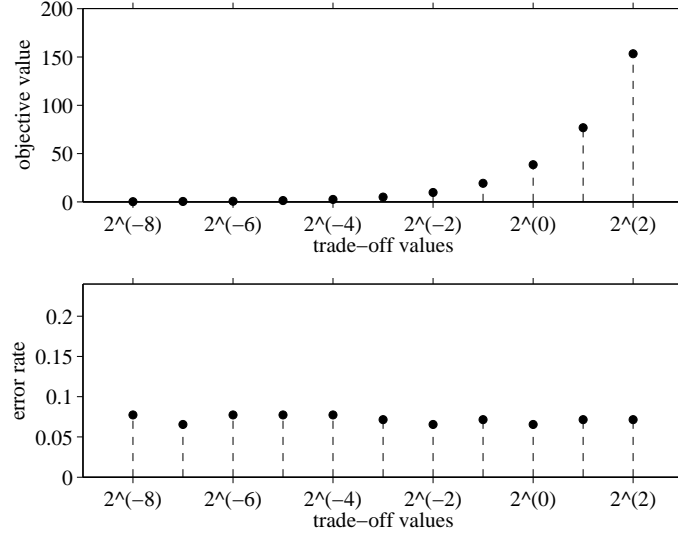


Figure 16: Model selection based on the proposed regularized objective values for 13 trade-off parameter $\mathfrak{C} = [2^{-8}, 2^{-6}, \dots, 2^2]$. The objective values are plotted at the top, and the corresponding empirical error rates are plotted at the bottom. Model selection using the proposed regularized criterion chooses $\mathfrak{C} = 2^{-8}$ with the empirical error rate of 0.08. This indicates that the proposed criterion minimizes the variability of the empirical error rate evenly across the trade-off parameters, where a small objective value corresponds to either a low empirical error rate or low variability of the empirical error rate.

a specific dataset, and the latter is concerned with the variation of empirical error rates across different datasets. Our future plan is to address the generalization issue using the resampling or the cross-validation perspective.

2.4 Summary

One of the distinctive characteristics of our MT systems is that each translation example in the database is mapped to a cluster of sensible variations of the source text, which is, in turn, represented by a statistical model. The statistical model is used to assign a class label to an input query text, and the target text corresponding to the assigned class label will be presented as the output target translation. In the pattern recognition field, this kind of task is usually approached by designing a statistical pattern classifier. This chapter focuses on the design of an optimal pattern classifier according to the Bayes decision theory.

Several classifier designs are examined to achieve the optimal classifier design with the expected error rates at the ultimate objective function. The empirical error rates are used to approximate the expected error rates and are also used as the objective functions for all the classifiers proposed in this chapter. Two main challenges in the design of such classifiers are the lack of information regarding the true functional form of the class-conditional density, and the insufficient number of observations available to approximate the parameters of the density function. KDE-based classifiers are examined to tackle the functional form issue. Corrective bandwidth learning is proposed for KDE-based classifiers so that the classifiers are optimized for empirical error-rate objective function. The model selection and regularization procedures are examined to address the insufficient data problem. An incremental mixture selection procedure is developed for mixture-based classifiers. Our experimental results indicate that the developed selection procedure can produce classifiers with fewer mixtures than the conventional mixture-based classifiers. Thus, we believe our classifiers can generalize better to the test data. A novel regularization procedure for a hyperplane classifier is examined to address the generalization issue. Our experimental results show that the resulting hyperplane classifiers have smaller variation in the error margins. Nevertheless, further studies are necessary to confirm the hypothesis that smaller variation in the error margins can lead to the smaller variability of the error-rate performance measure.

CHAPTER III

SUBJECTIVE JUDGMENTS IN THE ERROR-COST LEARNING PROCEDURE

A retrieval-based MT system uses pattern recognition techniques to extract a feature vector from an input query text and to retrieve matched examples during the translation process. The developed MT system represents each translation example using a set of sensible variations, which is then used to obtain a statistical model. The design of retrieval-based MT systems is explained in detail in Chapter 4. In this chapter, we address the task of incorporating subjective judgments into the statistical model, which is expected to further improve the MT performance. This kind of task is usually referred to as an error-cost learning problem in the pattern recognition field or cost-sensitive learning in the machine learning field [29, 32]. The subjective judgment indicates different degrees of subjective preferences or graded levels of importance. Specifically, this chapter examines two decision-theoretic approaches based on the Bayes decision theory for the error-cost learning procedure.

The first decision-theoretic approach is concerned with an active control of interclass confusions using the error-cost learning procedure. The control of interclass confusions is based on the subjective preferences, which is represented in terms of a *cost matrix*. The overall system performance of the proposed procedure is evaluated in terms of a *confusion matrix*. The designer can actively influence the learning procedure by assigning lower costs to bearable confusions and higher costs to unbearable confusions. The developed learning procedure can reduce specific interclass confusions in the confusion matrix by increasing the corresponding costs in the cost

matrix. We experiment with handwritten digit recognition tasks to validate the proposed procedure. Our experimental results indicate a high correlation between the costs in the cost matrix and the corresponding mistakes in the confusion matrix.

The second decision-theoretic approach incorporates the subjective judgments into a novel retrieval performance measure. First, we explain the motivation for the novel performance measure. The performance measure incorporates three different kinds of information: the subjective judgments, the retrieval ranking, and the length of retrieval results. Then, we provide a step-by-step derivation of the error-cost learning procedure for the novel performance measure. Our experimental results on a 20 Newsgroups dataset show that highly relevant examples appear more often at the top of the retrieval list after the developed learning procedure has been carried out.

The rest of this chapter is organized as follows: In Section 3.1, we investigate an active control of interclass confusions by using subjective preferences, which are compactly represented by a cost matrix. In Section 3.2, we propose a novel multi-level performance measure as a way to incorporate subjective judgment into a retrieval model. Finally, Section 3.3 summarizes our study of the error-cost learning procedure to incorporate subjective judgments into a statistical model.

3.1 Active Control of Interclass Confusions

This section examines an active control of interclass confusions using subjective preferences, which are compactly represented by a cost matrix. The system performance can be directly observed from a confusion matrix. The row and column conventions are the same for both the cost matrix and the confusion matrix. The rows of the matrix represent the system decisions, and the columns represent the true observation labels. Ideally, the values of the off-diagonal entries of the confusion matrix are zeros, which means none of the samples have been misclassified. In this study, we

utilize the error-cost learning procedure proposed by Qiang et al. [38] as the controller for interclass confusions. The main characteristic of the learning algorithm is that it offers a framework for combining the cost-sensitive decision rule and the classifier performance, i.e., the minimum cost into a novel cost function so that the system performance can be evaluated and optimized. From our experimental results, we observe that increasing the value of a specific entry in the cost matrix reduces the corresponding mistakes in the confusion matrix [37, 36].

The use of an error-cost learning procedure in practical scenarios can be justified by the following two scenarios. In the first scenario, the error cost is set purely by the designer. For example, in hand-written check recognition for financial transactions, the designer can set a higher cost for mistaking digit one as digit seven than mistaking digit seven as digit one. Another example is the 1-800 and 1-900 call voice dialing applications. Mis-recognizing digit eight as digit nine is far more serious than mis-recognizing digit nine as digit eight due to the nature of these calls. The 1-800 calls are toll-free while 1-900 calls will incur exorbitant service charges for the party making the call.

The second motivation comes from a larger framework of component modeling, an example of which, in a much simplified manner, is the design of prevalent speech recognition systems. Most of the current speech recognition systems for handling large vocabulary continuous speech use the so-called phone(me) models as the basic building block for constructing "word" reference models, which according to the given lexicon, are formed by concatenating phone(me) models. Several researchers have proposed the use of the Minimum Phone Error (MPE) as the training objective for a speech recognition system [85]. However, the Word Error Rate (WER) is still usually considered the performance metric, where the unit of decoding/decision stays at the phone(me) level (with obviously relevant lexical constraints imposed). In many real-world scenarios, errors on some phone(me)s may have more impact on the word error

than others.

The rest of this section is organized as follows: First, we review the formulation of the error-cost learning principle in Section 3.1.1. Section 3.1.2 describes our experiments on the learning principle using the pattern recognition and the hand-written digit recognition tasks.

3.1.1 Minimum Error-Cost Learning Procedure

The theoretical foundation of the error-cost learning principle is the same as that of the error rate learning, which is described in Section 1.2.1. Indeed, the error-cost learning procedure can also be formulated in a similar fashion as the error rate learning procedure. The formulation can be carried out as follows: *(i)* specification of the ultimate objective function, *(ii)* description of the decision rule, *(iii)* formulation of the optimization objective function, *(iv)* incorporation of the decision rules into the optimization performance measure, *(v)* formulation of the differentiable performance measure, and *(vi)* optimization procedure. The general steps in the error-cost learning principle are explained in detail in the remaining portions of this section.

(i) Specification of the ultimate objective function. In the error-cost learning procedure, the classifier performance is also based on the expected loss as defined in (10). For the purpose of completeness, we have rewritten some of the definitions. The conditional risk R for the decision $\mathbb{C}(X) = C_i$ can be defined as follows:

$$R(C_i|X) = \sum_{j=1}^M \epsilon_{ij} P(C_j|X) , \quad (62)$$

where $P(C_j|X)$ is the posterior probability for class $P(C_j|X)$, and ϵ_{ij} indicates the loss incurred when misclassifying observation X from class C_j as class C_i . The expected loss in (10) can be rewritten again as follows:

$$L = \sum_{i=1}^M \int_{\mathcal{X}} \epsilon_{ij} P(C_j, X) dX , \quad (63)$$

where $P(C_j, X)$ denotes the joint distribution function of the class identity and the observations.

(ii) **Description of the decision rule.** Unlike the error rate learning procedure, the expected loss (63) is minimized when the classifier $\mathbb{C}(X)$ implements the following decision rule [31].

$$\begin{aligned}\mathbb{C}(X) &= C_i \text{ if } i = \arg \min_k R(C_k|X) \\ &= \arg \min_k \sum_{j=1}^M \epsilon_{kj} P(C_j|X),\end{aligned}\tag{64}$$

where $P(C_j|X)$ is the posterior probability for class C_j . For the error-cost learning procedure, the loss is greater than zero for an incorrect classification and is zero for a correct classification as opposed to the zero-one loss in (12). The loss in the error-cost learning procedure can be defined as follows:

$$\epsilon_{ij} = \begin{cases} \geq 0 & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}\tag{65}$$

When the error-cost learning principle is used, the decision rule in (64) can be rewritten as

$$\mathbb{C}(X) = C_i \text{ if } i = \arg \min_k \sum_{j=1, j \neq k}^M \epsilon_{kj} P(C_j|X),\tag{66}$$

where X belongs to a class C_j , and $P(C_m|X)$ is the posterior probability of class C_m . This decision rule is usually referred to as the *minimum risk* decision rule. When the cost function is not uniform, the best decision policy is not necessarily the one that achieves *maximum a posteriori* probability. One can easily construct examples to demonstrate the discrepancy between a MAP decision and another decision that attempts to minimize the recognition cost. An important fallout of this discrepancy is that the validity of the conventional distribution estimation approach, which is a logical outcome of the MAP policy, is in doubt. Thus, we set a discriminant function $g_m(X; \Lambda) \geq 0$ to represent the m^{th} class, where $m \in \{1, 2, \dots, M\}$ and Λ are the

parameter sets that define the discriminant functions. The classification decision in (66) can be re-defined as

$$\mathbb{C}(X) = C_i \text{ if } i = \arg \max_k g_k(X; \Lambda). \quad (67)$$

If the true *a posteriori* probability is available, then $g_k(X; \Lambda)$ is defined as a monotonically decreasing function of the conditional risk and is defined as

$$g_k(X; \Lambda) = \exp\{-R(C_k|X)\} = \exp\left\{-\sum_{j=1}^M \epsilon_{kj} P(C_j|X)\right\}. \quad (68)$$

(iii) **Formulation of the optimization objective function.** For the non-uniform error criteria, the optimization objective function is based on the error cost instead of on the error rate as defined in (15). For clarity, let $i_X = \mathbb{C}(X)$ be the identity index as decided by the classifier and j_X be the true identity of pattern X . The cost incurred by a single sample is defined as

$$\ell(X; \Lambda) = \epsilon_{i_X j_X}. \quad (69)$$

Therefore, if the empirical system loss is defined over the realized sample-based costs, the average cost can be defined as follows:

$$\tilde{L}(\Lambda) = \frac{1}{\sum_{m=1}^M N_m} \sum_{n=1}^{N_m} \sum_{m=1}^M \epsilon_{i_X m} \mathbf{1}[x_{mn} \in C_m], \quad (70)$$

where N_m denotes the number of training patterns in class C_m , and $N = \sum_{m=1}^M N_m$ is the number of training patterns in all classes.

(iv) **Incorporation of the decision rules into the optimization performance measure.** For the error-cost learning procedure, the decision rules (67) is embedded into the average error cost (70), which is the performance measure. The optimization performance measure can be written as

$$\tilde{L}(\Lambda) = \frac{1}{N} \sum_{n=1}^N \left\{ \sum_{m=1}^M \sum_{i=1}^M \epsilon_{im} \mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)] \mathbf{1}[x_{mn} \in C_m] \right\}, \quad (71)$$

where the indicator function $\mathbf{1}$ represents the membership of an element in a set. In other words, it assumes the value of one if the argument is true and zero otherwise. The first indicator function $\mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)]$ denotes the decision rule in (67). The second indicator function $\mathbf{1}[x_{mn} \in C_m]$ denotes the class membership, i.e. observation x_{mn} belongs to class C_m .

(v) **Formulation of the differentiable performance measure.** The next challenge is to turn the objective function $\tilde{L}(\Lambda)$ in (71) into a smooth function suitable for optimization. Consider $\tilde{L}(\Lambda) = \sum_{m=1}^M \tilde{L}_m(\Lambda)$, where each L_m is defined as follows:

$$\tilde{L}_m(\Lambda) = \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^M \epsilon_{im} \mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)] \mathbf{1}[x_{mn} \in C_m]. \quad (72)$$

That is, $\tilde{L}_m(\Lambda)$ is the empirical error cost collected over training samples with a class label of C_m . This approximation needs to be made to the summands, which can be accomplished by

$$\sum_{i=1}^M \epsilon_{im} \mathbf{1}[i = \arg \max_k g_k(x_{mn}; \Lambda)] \approx \sum_{i=1}^M \epsilon_{im} \left(\frac{g_i(x_{mn}; \Lambda)}{G(x_{mn}; \Lambda)} \right)^\beta, \quad (73)$$

where $G(x_{mn}; \Lambda) = [\sum_{k=1}^M g_k^\eta(x_{mn}; \Lambda)]^{1/\eta}$. Note that if the design parameter $\eta \rightarrow \infty$ and $\beta \rightarrow \infty$, we obtain the following approximation.

$$\left(\frac{g_i(x_{mn}; \Lambda)}{G(x_{mn}; \Lambda)} \right)^\beta \approx \begin{cases} 1, & \text{if } G(x_{mn}; \Lambda) = \max_k g_k(x_{mn}; \Lambda) \\ 0, & \text{otherwise} \end{cases}. \quad (74)$$

With the use of this smoothing strategy, the loss for individual tokens $x_{mn} \in C_m$ can be defined as

$$\ell(x_{mn}) = \sum_{i=1}^M \epsilon_{im} \left(\frac{g_i(x_{mn}; \Lambda)}{G(x_{mn}; \Lambda)} \right)^\beta. \quad (75)$$

Finally, the smoothed empirical error cost is defined as follows:

$$\hat{L}(\Lambda) \approx \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M \epsilon_{im} \left(\sum_{i=1}^M \frac{g_i(x_{mn}; \Lambda)}{G(x_{mn}; \Lambda)} \right)^\beta \mathbf{1}[x_{mn} \in C_m]. \quad (76)$$

Indeed, the objective function in (76) is more general than that in (20) because the error-cost can be set as any positive number in (76). When the error cost is set to

zero for a correct decision and one for an incorrect decision, the objective function in (76) minimizes the error rate.

(vi) **Optimization procedure.** The optimization procedure for the error-cost learning algorithm is similar to that of the error rate learning described in Section (1.2.1). For the purpose of completeness, the parameter update equation is defined again. Let t denote the epoch index for parameter adjustment upon presentation of a training pattern, and the GPD algorithm iteratively modifies the parameter values according to the following update equation

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon^{(t)} \nabla \ell(X; \Lambda)|_{\Lambda=\Lambda^{(t)}} , \quad (77)$$

where $\varepsilon^{(t)}$ is the learning steps and $\nabla \ell(X; \Lambda)$ is the gradient of the loss from each training data, which is defined in (75).

The error-cost learning procedure is used to control specific inter-class confusions in the remaining portion of this section. Certain inter-class confusions are more tolerable than others according to subjective preferences. These subjective preferences can be determined directly by humans or dictated by the requirements of other procedures in the pattern recognition systems. We propose one way to represent these subjective preferences by encoding the preferences into a positive semi-definite matrix. Each entry in the preference matrix represents an inter-class confusion cost with higher costs indicating intolerable confusions and lower costs indicating tolerable confusions. The confusion costs in this matrix can be directly used in the error-cost learning procedure.

For the learning in this scenario, we set the discriminant function g_i for class C_i in terms of a Gaussian Mixture Model (GMM) with a diagonal covariance matrix, which is defined as follows:

$$g_i(X; \Lambda) = \exp \left(- \sum_{k=1}^K \epsilon_{ik} b_k P(C_k) \right) , \quad (78)$$

where $b_k = p(X|C_k) = \sum_{j=1}^K c_j \mathcal{N}(X; \mu_j, \sigma_j^2)$ is the likelihood function and $P(C_k)$ is

the *a priori* probability of class C_k . Thus, the parameter sets $\Lambda = \{\mu_k, \sigma_k^2, c_k\}_{k=1}^K$ for K equal the number of mixtures.

The notations used are as follows: j is the index of the class identity, k denotes the mixture number, and l indicates the dimension starting from one to D . For clarity, $P(C_i)$, $g_i(X; \Lambda)$ and $G(X; \Lambda)$ are written into P_i , g_i , and G respectively. First, to help the convergence of the learning process, parameter transformation is applied to the mean μ_{jkl} to obtain $\tilde{\mu}_{jkl}$, where $\tilde{\mu}_{jkl} = \frac{\mu_{jkl}}{\sigma_{jkl}}$. The learning process for the mean $\tilde{\mu}_{jkl}$ vector is as follows:

$$\tilde{\mu}_{jkl}^{(t+1)} = \tilde{\mu}_{jkl}^{(t)} - \varepsilon^{(t)} \frac{\partial \ell(x_{jn})}{\partial \tilde{\mu}_{jkl}}, \quad (79)$$

where $\varepsilon^{(t)}$ is the learning step during t iteration, and the gradient is defined as $\frac{\partial \ell(x_{jn})}{\partial \tilde{\mu}_{jkl}} = \frac{\partial \ell(x_{jn})}{\partial b_m} \frac{\partial g_i}{\partial b_j} \frac{\partial b_j}{\partial \mu_{jkl}} \frac{\partial \mu_{jkl}}{\partial \tilde{\mu}_{jkl}}$. The gradient can be computed as follows:

$$\begin{aligned} \frac{\partial \ell(x_{jn})}{\partial b_m} &= \frac{P_m}{\sum_{r=1}^M b_r P_r} \left\{ \left(\frac{-\sum_{k=1}^M \epsilon_{kj} (\epsilon_{km} + \log(g_k)) g_k}{G} \right) + \left(L_n \frac{\sum_{k=1}^M (\epsilon_{km} + \log(g_k)) g_k^\eta}{G^\eta} \right) \right\} \\ \frac{\partial b_j}{\partial \tilde{\mu}_{jkl}} &= c_{jk} (2\pi)^{-D/2} \exp \left\{ -\frac{1}{2} \sum_{l=1}^D \left(\frac{x_l - \mu_{jkl}}{\sigma_{jkl}} \right)^2 \right\} \left(\frac{x_l - \mu_{jkl}}{\sigma_{jkl}} \right) \left(\prod_l \sigma_{jkl} \right)^{-1}, \end{aligned}$$

The new μ_{jkl} is obtained from $\mu_{jkl} = \tilde{\mu}_{jkl} \sigma_{jkl}$ afterwards. Learning for the variance and weight vectors follows the same procedure [38].

The experimental results reported in the next section consist of the following quantities.

- The Empirical System Risk (ESR) defined in (71) is the appropriate figure to report for system performance evaluation on finite test data. The Empirical Smoothed System Risk (ESSR) defined in (76) is the objective function employed during system optimization (training).
- The Average Conditional Risk (ACR) is the conditional risk in (62), except that the *a posteriori* probabilities therein are computed from the models that the system uses, rather than being computed from the true distributions. The average is taken over the given data set.

- The Empirical Classification Error (ECE) rate is the unsmoothed version of the expected error-cost L in (76) based on the classical error count with a zero-one error-cost assignment. The term "empirical" here means that the result is obtained from a given data set, rather than from the expected value based on the true distribution. The ECE rate is the most popular conventional performance metric as it represents an estimate of the error probability.

3.1.2 Experimental Results

This section includes two classification experiments based on the error-cost learning procedure. The first experiment establishes the connections between the cost matrix and the confusion matrix in the error-cost learning algorithm. The second experiment deals with the digit recognition task using the label-difference cost matrices. Both experiments provide insight into how the error-cost learning procedure performs in the real-world applications.

3.1.2.1 Active Control of Inter-Class Confusion

The error-cost learning procedure can be used to control the confusion among classes produced by the recognition system. As a potential diagnostic tool, the confusion matrix is often employed to reveal the performance of a recognition system. The matrix reflects the statistics of the system's decisions with the rows and columns corresponding to the system decisions and the true class labels, respectively. The entries in the confusion matrix are the counts of tokens resulting from the system's decisions with the possibly of being normalized by the total number of tokens tested. For this research, we conducted a series of experiments using the confusion matrix to investigate how inter-class confusion is affected by a non-uniform cost matrix after the application of the active learning procedure [69].

The experiments in this section use automatically generated multi-class multi-dimensional random observations. We generate four classes of two-dimensional data

with 128 samples for each class and a total of 512 samples for the training and testing sets. Each class has the same number of mixtures but has different class *a priori* probabilities. The scatter plot for the classes is shown in Figure 17. All of the mixtures have different covariance matrices and different mixture weights. This is what we refer to as the true models. The conventional or baseline models are based on maximum likelihood density estimations. Compared to conventional error rate classifiers, conventional cost sensitive models have one additional step, which is the application of additional expected error criteria. This additional step is applied before the classification decision is made [29, 32].

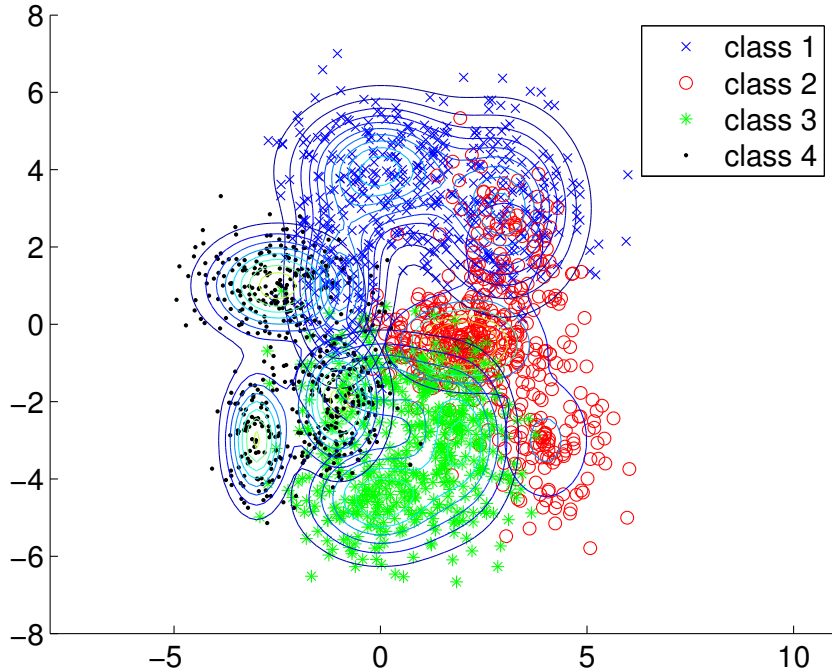


Figure 17: Scatter plot of the true model for the experiments on active-control of interclass confusion.

We start this experiment by using the standard zero-one cost matrices. Table 6 lists the cost matrix and the confusion matrices of the non-cost-sensitive (cost-insensitive) classification. Then, we arbitrarily choose to minimize the misclassification of true class four as class one (the value on row one, column four of the cost

matrix). In other words, the value of two, four, and eight are used as the cost of misclassification. For each cost value, we collect the statistics on the confusion matrix based on thirty trials. Table 7 shows the performance result when the misclassification cost is equal to two. The cost matrix is shown in Table 7(a), and the confusion matrix results of the true, baseline, and cost-optimized models are shown in Tables 7(b), 7(c), and 7(d), respectively. Notice that the cost-optimized models are closer to the ground truth (the true models) than are the baseline models (i.e., $\text{true}[1,4] < \text{cost-optimized}[1,4] < \text{baseline}[1,4]$). The main reason is that the proposed technique uses the cost matrix to optimize the classifier models. However, the baseline models are based only on the conventional density estimation and are not optimized for the cost matrix.

The entry in row one column four is the only difference between the cost matrices used to obtain the result in Table 6 and Table 7. Comparing these two tables, we can observe the impact of the cost matrices on the confusion matrices. The misclassifications from class four into class one decrease if we compare the corresponding models in those two tables. The reason is that all models use the same additional expected error criteria in their decision rule. We also notice that the row one column four entry of the baseline models is lower than that of the cost-optimized models in Table 6. However, the baseline models have a higher error rate compared to the cost-optimized models because the minimization of the error cost with a zero-one cost matrix is equivalent to the minimization of the error rate.

We expected that as the cost value increased, the misclassification between the two classes would decrease. Therefore, we experimented further with the cost values of four and eight. We also expected that whether or not the correct numbers of mixtures were used, the cost-optimized models would still perform better than the baseline models. We compared a model matched with three mixtures and a model mismatched with four mixtures. Table 8 shows the results of the misclassification

Table 6: Confusion matrices of true (b), baseline (c), and cost-optimized (d) models based on cost matrix (a) for cost-insensitive classification.

$$\begin{bmatrix} 0 & 1 & 1 & \boxed{1} \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(a) cost matrix

$$\begin{bmatrix} 107.1 & 14.3 & 0.7 & \boxed{8.4} \\ 9.5 & 104.7 & 11.5 & 0.2 \\ 0.7 & 8.7 & 98.7 & 7.9 \\ 10.6 & 0.3 & 17.1 & 111.4 \end{bmatrix}$$

(b) true model

$$\begin{bmatrix} 103.5 & 16.7 & 1.1 & \boxed{9.3} \\ 11.3 & 97.8 & 11.0 & 0.3 \\ 1.8 & 13.0 & 100.7 & 17.9 \\ 11.5 & 0.4 & 15.2 & 100.4 \end{bmatrix}$$

(c) baseline

$$\begin{bmatrix} 104.7 & 15.2 & 1.1 & \boxed{9.5} \\ 11.1 & 98.2 & 9.8 & 0.4 \\ 1.2 & 14.2 & 102.1 & 17.1 \\ 11.0 & 0.4 & 15.0 & 101.0 \end{bmatrix}$$

(d) cost-optimized

Table 7: Confusion matrices of true (b), baseline (c), and cost-optimized (d) models based on cost matrix (a) for cost-sensitive classification.

$$\begin{bmatrix} 0 & 1 & 1 & \boxed{2} \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(a) cost matrix

$$\begin{bmatrix} 100.5 & 14.2 & 0.1 & \boxed{3.6} \\ 9.5 & 104.7 & 11.5 & 0.3 \\ 0.8 & 8.6 & 98.8 & 7.9 \\ 17.1 & 0.4 & 17.4 & 116.1 \end{bmatrix}$$

(b) true model

$$\begin{bmatrix} 97.3 & 16.4 & 0.5 & \boxed{5.3} \\ 11.3 & 97.8 & 11.0 & 0.3 \\ 2.0 & 13.0 & 100.8 & 18.2 \\ 17.4 & 0.7 & 15.6 & 104.1 \end{bmatrix}$$

(c) baseline

$$\begin{bmatrix} 98.6 & 15.3 & 0.6 & \boxed{4.7} \\ 10.8 & 98.0 & 9.9 & 0.3 \\ 1.8 & 13.9 & 102.3 & 18.4 \\ 16.7 & 0.6 & 15.0 & 104.4 \end{bmatrix}$$

(d) cost-optimized

(class four as class one) over thirty trials. The results show the average number of misclassifications decreases for all techniques as the cost value increases from two to eight since fewer samples are categorized as class one. As expected, the cost-optimized models in general perform better than the baseline models, even the cost-optimized models with the wrong number of mixtures.

We further compared the confusion matrix entries on a trial-by-trial basis. For each of the thirty trial runs, we took the difference between the R1C4 entries in the confusion matrices corresponding both to the baseline system and to the cost-optimized system. We found that the difference is positive if the cost-optimized

system is able to reduce the target type of error; otherwise the difference is negative. Table 9 lists the sign (positive/zero/negative) of the difference. The numbers in the table indicate the number of trial runs that achieved positive reduction, no change, or negative reduction of the target type of error, respectively. In all cases, the number of positive reductions is always higher than that of negative reductions.

In this experiment, we use class-four-to-class-one error to illustrate the active learning procedure for the control of interclass confusions. However, the proposed active learning method will work with any valid cost matrix.

Table 8: The mean (standard deviation) of row one and column four of the multi-class confusion matrix across true models, baseline models, and cost-optimized models.

	3-mixture			4-mixture		
column 1 row 4	2	4	8	2	4	8
true	3.60(1.83)	1.03(0.89)	0.30(0.53)	3.60(1.83)	1.03(0.89)	0.30(0.53)
baseline	5.30(3.36)	2.80(2.44)	1.73(1.95)	6.20(3.31)	3.13(2.43)	1.60(1.75)
cost-optimized	4.70(3.04)	2.37(2.27)	1.37(1.87)	5.63(2.90)	2.53(1.68)	1.27(1.36)

Table 9: The signs of difference between the R1C4 entries of the confusion matrices produced by the baseline and the cost-optimized models.

	3-mixture	4-mixture
	(pos/zero/neg)	(pos/zero/neg)
$e_{14} = 2$	(16/7/7)	(18/6/6)
$e_{14} = 4$	(11/15/4)	(14/11/5)
$e_{14} = 8$	(8/21/1)	(9/20/1)

3.1.2.2 A Handwritten Digit Recognition Task

We further validate our approach using an experiment with a real-world task, i.e. a digit recognition task using a US Postal Service (USPS) dataset. This dataset contains gray-scale handwritten digit images scanned from envelopes by the US Postal Service. The task in this experiment is to correctly identify ten different digits. Thus, the number of class M is equal to ten. The digit images are of size 16×16 , with pixel values in the range of zero and two. Thus, the original feature size is 256,

and the conventional Principal Component Analysis (PCA) technique is used to reduce the feature size using only 70%, 80% and 90% of the PCA variances [31]. The main purpose of this dimension reduction is to demonstrate the stabilities of the proposed algorithm across different feature sizes. The original training set contains 7,291 images, while the testing set contains 2,007 images. We combine the training and testing set and randomly select data for thirty trial runs. For each run, the data are divided into training, validation, and testing sets, each of which has approximately 517 tokens. For each set, the digit class zero to digit class nine has roughly 86, 71, 52, 46, 47, 40, 46, 44, 39, 46 tokens respectively. Figure 18 shows samples of the USPS digit images.



Figure 18: Sample images from the USPS digit dataset for a digit recognition task.

The system performances reported in Table 11 are normalized multi-class confusion matrices based on the average of 30 runs of the experiments. Specifically, those confusion matrices are normalized column-wise to account for different class sizes.

Thus, each entry in the normalized multi-class confusion matrix represents a percentage (a proportion) instead of the number of error counts. In practice, the confusion matrix depends on the cost matrix used. Two different cost matrices are used for the experiments in this section. The first cost matrix assigns confusion costs according to the absolute value of the difference between two class labels. This cost matrix is listed in Table 10(a) and is called a label-difference cost matrix. Table 10(b) lists an exponential-label-difference cost matrix. The confusion cost varies linearly if a label-difference cost matrix in Table 10(a) is used, and varies exponentially if, as in Table 10(b), an exponential-label-difference cost matrix is used.

Table 10: The label-difference cost matrix assigns error-cost based on the absolute value of the difference in the class labels. The exponential-label-difference cost matrix is obtained from taking the exponential of the difference in class labels.

$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 \\ 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 & 512 \\ 2 & 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \\ 4 & 2 & 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 \\ 8 & 4 & 2 & 1 & 2 & 4 & 8 & 16 & 32 & 64 \\ 16 & 8 & 4 & 2 & 1 & 2 & 4 & 8 & 16 & 32 \\ 32 & 16 & 8 & 4 & 2 & 1 & 2 & 4 & 8 & 16 \\ 64 & 32 & 16 & 8 & 4 & 2 & 1 & 2 & 4 & 8 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 & 2 & 4 \\ 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 & 2 \\ 512 & 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{bmatrix}$
--	--

(a) A label-difference cost matrix. (b) An exponential-label-difference cost matrix.

Table 11 lists the normalized confusion matrices for the baseline and the cost-optimized models using the label-difference cost matrix listed in Table 10(a). Each of the normalized confusion matrices was obtained from averaging the confusion matrices of 30 trial runs using the features with 80% of the total PCA variance. Comparing the two confusion matrices, one may observe that the diagonal entries of the cost-optimized confusion matrix have a higher percentage in general than the entries in the baseline confusion matrix. This means that more tokens were classified correctly when the cost-optimized models were used.

Table 11: The normalized multi-class confusion matrices using the baseline (a) and cost-optimized (b) models for features with 80% of the total PCA variances using the label-difference cost matrix.

79.1	0.0	0.3	0.2	0.2	0.6	0.6	0.1	0.2	0.1
0.0	67.5	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0
2.0	0.3	44.5	1.5	2.2	1.1	2.6	1.4	1.5	0.4
0.7	0.2	1.5	38.4	0.0	2.3	0.0	0.1	2.0	0.1
0.4	0.5	2.3	0.4	38.5	1.1	0.8	1.0	1.0	4.0
2.8	0.4	0.9	3.7	0.5	33.0	2.2	0.8	2.0	0.5
1.2	0.5	0.1	0.0	0.6	0.7	39.7	0.0	0.2	0.0
0.0	0.7	0.8	0.1	0.2	0.1	0.0	36.7	0.3	2.0
0.1	0.7	1.4	1.1	0.4	0.6	0.5	0.3	31.7	0.5
0.0	0.4	0.1	0.3	4.5	0.1	0.0	3.6	0.5	38.3

(a) Normalized confusion matrix for the baseline models

81.8	0.2	0.6	0.3	0.2	0.7	0.7	0.1	0.4	0.0
0.0	68.0	0.0	0.0	0.3	0.0	0.0	0.0	0.2	0.0
1.2	0.6	43.3	1.2	1.6	0.7	1.6	0.8	1.2	0.3
0.6	0.2	2.1	39.1	0.2	2.6	0.1	0.1	2.0	0.2
0.3	0.3	2.2	0.4	40.3	1.2	0.6	0.8	0.8	2.4
1.5	0.5	0.9	3.1	0.5	32.3	2.0	0.7	2.0	0.6
0.8	0.3	0.3	0.0	0.6	1.1	41.0	0.0	0.2	0.0
0.0	0.0	1.0	0.4	0.2	0.2	0.0	37.9	0.3	1.9
0.2	0.4	1.3	1.1	1.1	0.6	0.4	0.3	31.8	0.8
0.0	0.2	0.1	0.1	2.4	0.3	0.0	3.4	0.6	39.6

(b) Normalized confusion matrix for the cost-optimized models

Because the confusion matrix is difficult to read when the number of classes becomes large, other performance measures, including Empirical Classification Error (ECE), Average Conditional Risk (ACR), and Empirical System Risk (ESR), are listed in Table 12 (a) for the label-difference cost matrix, and in Table 12 (b) for the exponential-label-difference cost matrix. From Table 12 (a), it can be observed that the ECE, ACR, and ESR are reduced under the cost-optimized system using the label-difference cost matrix. Table 12 (b) shows that the ACR and ESR are reduced but that ECE increases under the cost-optimized system using the exponential-label-difference cost matrix. The reason is that the cost range of the exponential-label-difference cost matrix is much wider than that of the label-difference cost matrix.

With the use of the exponential-label-difference cost matrix, the learning algorithm clearly prefers the reduction in ACR to the reduction in ECE. In other words, users have the freedom to emphasize on ACR by using a cost matrix.

Table 12: The performance measures of the baseline and cost-optimized models using the label-difference and the exponential-label-difference cost matrices.

PCA Variance Percentage	Models	Empirical Classification Error (ECE)	Average Conditional Risk (ACR)	Empirical System Risk (ESR)
70%	baseline	0.15(0.02)	0.49(0.07)	0.17(0.02)
	cost-opt	0.13(0.02)	0.41(0.05)	0.14(0.02)
80%	baseline	0.14(0.02)	0.45(0.08)	0.14(0.02)
	cost-opt	0.12(0.02)	0.38(0.06)	0.11(0.02)
90%	baseline	0.14(0.02)	0.47(0.07)	0.13(0.02)
	cost-opt	0.12(0.02)	0.37(0.05)	0.10(0.02)

(a) Performance measures using the label-difference cost matrix.

PCA Variance Percentage	Models	Empirical Classification Error (ECE)	Average Conditional Risk (ACR)	Empirical System Risk (ESR)
70%	baseline	0.20(0.019)	3.10(0.51)	1.52(0.09)
	cost-opt	0.23(0.037)	2.72(0.30)	1.51(0.13)
80%	baseline	0.18(0.023)	3.04(0.53)	1.40(0.08)
	cost-opt	0.20(0.037)	2.74(0.46)	1.36(0.07)
90%	baseline	0.18(0.02)	3.14(0.52)	1.42(0.10)
	cost-opt	0.21(0.05)	2.93(0.43)	1.35(0.13)

(b) Performance measures using the exponential-label-difference cost matrix.

Because only ten unique costs are available in either the label-difference or the exponential-label-difference cost matrix, the entries of a normalized confusion matrix can be grouped according to their costs. The averages of the resulting confusion percentages are listed in Table 13 (a) for the label-difference cost matrix, and in Table 13 (b) for the exponential label-difference cost matrix. Grouping the confusion matrix entries helps differentiate the confusions distributed among the different costs.

It can be observed that when the label-difference cost matrix is used, and when the features are extracted by keeping 80% of the total PCA variances, the ECE decreases

because the percentages of the entries of the confusion matrix with cost 0 decrease. Our expectation is that the confusion will shift from the high group costs, e.g. group cost 9, to low group cost, e.g. group cost 1. Indeed, more confusions occur in group cost 1 for the cost-optimized models. However, the optimization actually reduces the number of confusions even more, from group cost 5, 6 and 7 than it does from group cost 8 and 9. The possible explanation is that more entries in the confusion matrix with group cost 5, 6 and 7 are present than those in the matrix with group cost 8 and 9. In Table 13 (a), similar tendencies are observed for features extracted by keeping 80% of the PCA variance with the exponential-label-difference cost matrix. For the cost-optimized models, fewer confusions occur in group cost 32, 64 and 128, but even more confusions occur in group cost 2, 4, 8, 16. Nevertheless, the ECE for the exponential-label-difference cost matrix increases because the entries in the confusion matrix with group cost 0 increase in percentage.

One may notice that both the label-difference cost matrix and the exponential-label-difference cost matrices share one important property; the error costs are less expensive for entries closer to the diagonal entries than for those farther from the diagonal entries. In other words, the errors are less expensive for confusing two adjacent classes than for confusing two non-adjacent classes. Based on this property, a confusion matrix may be summarized using a single quantity weighted by its location. We shall name this quantity Weighted Total Confusion (WTC), which is defined as follows:

$$WTC(\tilde{K}) = \sum_{i=1}^M \sum_{j=1}^M (|i - j|) \tilde{K}_{ij}, \quad (80)$$

where \tilde{K} is a normalized confusion matrix, and \tilde{K}_{ij} indicates the i^{th} row and the j^{th} column of a normalized confusion matrix \tilde{K} . The WTCs for the label-difference cost matrix and the exponential-label-difference cost matrix are summarized in Table 14.

Table 13: The average error rate across different costs in the confusion matrix using the label-difference and the exponential-label-difference cost matrices.

PCA Variance Percentage	Models	Error Cost Group									
		0	1	2	3	4	5	6	7	8	9
70%	baseline	15.7	1.4	3.1	1.1	1.0	4.2	1.3	0.4	0.2	0.0
	cost-opt	14.0	1.6	2.7	1.1	1.1	3.0	1.1	0.4	0.3	0.1
80%	baseline	14.5	1.2	2.8	1.2	1.0	3.6	1.3	0.4	0.3	0.1
	cost-opt	13.1	1.4	2.6	1.1	1.1	2.6	1.1	0.3	0.4	0.0
90%	baseline	15.0	1.2	2.8	1.2	1.2	3.8	1.4	0.5	0.3	0.1
	cost-opt	12.9	1.5	2.5	1.2	1.1	2.3	1.0	0.3	0.4	0.2

(a) Average error rates using the label-difference cost matrix.

PCA Variance Percentage	Models	Error Cost Group									
		1	2	4	8	16	32	64	128	256	512
70%	baseline	21.6	3.9	4.0	2.1	1.7	2.7	0.7	0.2	0.0	0.0
	cost-opt	24.4	5.0	4.9	2.4	1.6	1.8	0.5	0.1	0.0	0.0
80%	baseline	18.8	3.2	3.4	2.1	1.5	2.4	0.7	0.2	0.1	0.0
	cost-opt	21.5	3.8	4.6	2.4	1.5	1.7	0.5	0.1	0.2	0.0
90%	baseline	18.9	2.8	3.5	2.0	1.8	2.5	0.8	0.3	0.1	0.0
	cost-opt	21.9	3.7	4.4	2.4	1.6	2.1	0.6	0.2	0.1	0.0

(b) Average error rates using the exponential-label-difference cost matrix.

Table 14: The Weighted Total Confusion (WTC) of the baseline and cost-optimized confusion matrices.

PCA Variance Percentage	$WTC(\tilde{K})$		
	Models	Label Difference	Exponential Label Difference
70%	baseline	5.13(0.70)	5.41(0.63)
	cost-opt	4.37(0.58)	5.43(0.74)
80%	baseline	4.78(0.79)	4.87(0.75)
	cost-opt	4.10(0.60)	5.06(0.99)
90%	baseline	5.04(0.76)	5.09(0.69)
	cost-opt	3.93(0.51)	5.06(0.99)

Using the label-difference cost matrix, the WTCs are obviously lower for the cost-optimized models in comparison with those for the baseline models. This indicates that the error counts have shifted towards the diagonal entries, and, moreover, that the error rate has decreased for the label-difference cost matrix. After all, WTC is

consistent with the label-difference cost matrix. For the exponential-label-difference cost matrix, the increase in the error rate has offset the shifts of the error counts toward the diagonal entries. Therefore, the WTCs are relatively unchanged before and after the optimization using the exponential-label-difference cost matrix. This also indicates that excessive weighting is not advisable in the error-cost learning procedure.

In summary, the set of experiments in this section demonstrate the effectiveness of the proposed error-cost learning procedure in optimizing a real-world dataset based on arbitrary user-specified cost matrices. In general, the overall ACR and ESR are reduced. The reduction in the ECE is a by-product of the reduction of ACR and ESR. Moreover, the improvements in ACR and ESR are not greatly influenced by the feature sizes as we have used features obtained from retaining 70%, 80%, and 90% of the PCA variance.

3.2 Multi-level Relevance Performance Measure

In this section, we propose a novel multi-level retrieval performance measure as a way to incorporate subjective judgments into a statistical model. Performance assessment is critical to the design of a document retrieval system [56]. Conventional retrieval systems often evaluate document relevance in a bi-level manner, i.e., value of zero for irrelevant and value of one for relevant documents. One example of such a bi-level relevance measure is the average precision [84]. Bi-level relevance measures, however, do not reflect the different degrees of relevance of the retrieved documents. A graded relevance scale, however, can reflect how well a retrieved document meets the information needs of the user. For instance, relevance assessment can be based on a tri-level relevance indicator $\{0,1,2\}$ rather than bi-level relevance indicator $\{0,1\}$ to reflect the degree of relevance. Normalized Discounted Cumulative Gain (NDCG), which is a multi-level relevance assessment criterion, was developed to use multi-level

document gain in the assessment of a document retrieval system [56].

In this section, we investigate an optimization strategy for a document retrieval system using a novel multi-level relevance measure as the objective function. The conventional multi-level relevance measure NDCG has two drawbacks. First, the NDCG criterion does not explicitly consider the number of retrieved documents, which may hinder objective assessment of a retrieval system. For example, suppose two different retrieval systems are presented with the same query. The first system retrieves three documents with relevance values of $\{2,1,0\}$, where the value of zero indicates no relevance and the value of two indicates the most relevance. The second system retrieves five documents with relevance values of $\{2,1,0,0,0\}$. The performance of the two retrieval systems would be the same based on the NDCG criterion. However, we believe the first retrieval system outperforms the second one, since the first system retrieves only one irrelevant document, whereas the second system retrieves three irrelevant documents. The second drawback is the mismatch between the gain assignment and the training criterion of the statistical (retrieval) model. In other words, the gain in the NDCG criterion is subjectively assigned by a human, and the statistical (retrieval) model is usually optimized for certain types of errors, such as minimum cost, mean-squared errors, classification errors, or margin errors. Because the degree of irrelevance can be directly formulated in terms of the error cost, we propose optimizing the statistical (retrieval) system using a multi-level performance measure that incorporates both the number of retrieved documents and the degree of irrelevance (cost) as assigned by humans. We name the novel objective function Normalized Penalized Cumulative Cost (NPCC).

Our approach has to address two main issues. The first issue is to define a novel retrieval performance measure NPCC, and the second issue is to derive a differentiable version of the NPCC that can be used as an objective function for optimization. The rest of this section is organized as follows: Section 3.2.1 explains the proposed

NPCC objective function. Section 3.2.2 describes the learning procedure for the proposed objective function. Section 3.2.3 presents the experimental results on the 20 Newsgroups dataset.

3.2.1 Multi-Level Irrelevance Performance Measure

We define a performance measure that systematically penalizes the loss in ranking and relevance. The performance measure serves as an objective function to be optimized by the information retrieval systems. The proposed performance measure is a dissimilarity-based measure that emphasizes the ranking cost and progressively reduces the document value as its rank increases. Let us define the Cumulative Cost (CC) as

$$CC(i) = \sum_{j=1}^i \text{irrel}(q, d_j), \quad (81)$$

where q is the prompted query, d_j is a document retrieved at rank: $j = \text{rank}_q(d_j)$, and i is the index of the i^{th} retrieved document. The degree of irrelevance between the query and the retrieved documents can be computed in a number of ways. For example, if $\text{rel}(q, d)$ represents the degree of relevance between q and a document d , as in NDCG, one simple way to relate relevance and irrelevance judgments is a smooth exponential function, i.e., $\text{irrel}(q, d) = \exp(-\text{rel}(q, d))$, therefore $\text{rel}(q, d) = 0 \mapsto \text{irrel}(q, d) = 1$.

The CC can be extended to an *evaluation measure* that makes a clear distinction between the different ranks, where the ranking errors are included as penalizing factors. We denote the proposed irrelevance evaluation measure as the Penalized Cumulative Cost (PCC), which is defined as follows:

$$PCC(i) = \sum_{j=1}^i \text{irrel}(q, d_j) \times (K - j + 1), \quad (82)$$

where K is the number of documents being retrieved. Note that PCC represents both the irrelevance judgment and the ranking loss.

Similar to Discounted Cumulative Gain (DCG) Criterion [56], the PCC criterion

can also be normalized with respect to the theoretical "worst retrieval vector." Suppose our retrieval system is pre-configured to return 3 documents, i.e., $K = 3$. For a given query document, the gains of the retrieved documents are $\{2,0,1\}$; thus the CC and PCC criteria are $\{0.1,1.1,1.5\}$ and $\{0.4,2.4,2.7\}$ respectively. Because the gain in the worst possible scenario is $\{0,1,2\}$ with a PCC criterion of $\{3.0,3.7,3.8\}$, the Normalized Penalized Cumulative Cost (NPCC) can be obtained by dividing PCC by the normalizing constant 3.8, i.e., $\text{NPCC}=\{0.1,0.6,0.7\}$. Table 15 lists additional examples of how the PCC and NPCC performance measures are computed.

3.2.2 Differentiable Relevance Measure

This section derives a differentiable version of the proposed NPCC performance measure so that it can be used in language model-based document retrieval systems. The idea of using statistical language modeling for document retrieval was popularized by Ponte and Croft [84]. The Maximum Likelihood (ML) is a traditional criterion to estimate each document model. Let $q = [q^{(1)}, q^{(2)}, \dots, q^{(T)}]$ denote the word (or term) sequence of an input query q . Given a text document d , the maximum likelihood estimation of the probability of the t^{th} term in the query q can be written as

$$\hat{P}_{\text{ML}}(q^{(t)}|d) = \frac{\text{count}(q^{(t)}, d)}{\sum_j \text{count}(q^{(j)}, d)} , \quad (83)$$

where $\text{count}(q^{(t)}, d)$ denotes the count of occurrences of term event $q^{(t)}$ in the document d . Because of data sparsity, a large number of terms will have zero-probabilities. One solution to circumvent this problem is to use Jelinek-Mercer (JM) smoothing [57], which linearly interpolates the ML document model with a background model. Based on this smoothing technique, the probability of the t^{th} term in the query q can be written as

$$\tilde{P}_{\text{JM}}(q^{(t)}|d) = \left[\lambda \hat{P}_{\text{ML}}(q^{(t)}|d) + (1 - \lambda) \bar{P}_{\text{BK}}(q^{(t)}|\mathcal{D}) \right] , \quad (84)$$

where λ is an empirical interpolation coefficient, $\bar{P}_{\text{BK}}(q^{(t)}|\mathcal{D})$ is the background collection model, and $\hat{P}_{\text{ML}}(q^{(t)}|d)$ indicates the ML estimation of the term probability.

Table 15: Six examples of multi-level relevance ranking lists demonstrating the difference between the conventional DCG criterion and the proposed PCC criterion. The DCG criterion indicates the value to be maximized and emphasizes the gain value. However, the PCC criterion denotes the cost to be minimized and emphasizes the ranking loss.

System A								
R	G		CG	DCG	NDCG		PCC	NPCC
1	2		2	2.0	0.7		0.4	0.1
2	1		3	3.0	1.0		1.1	0.3
3	0		3	3.0	1.0		2.1	0.6

System B								
R	G		CG	DCG	NDCG		PCC	NPCC
1	2		2	2.0	0.7		0.4	0.1
2	0		2	2.0	0.7		2.4	0.6
3	1		3	2.6	0.9		2.7	0.7

System C								
R	G		CG	DCG	NDCG		PCC	NPCC
1	1		1	1.0	0.3		1.1	0.3
2	2		3	3.0	1.0		1.3	0.4
3	0		3	3.0	1.0		2.3	0.6

System D								
R	G		CG	DCG	NDCG		PCC	NPCC
1	1		1	1.0	0.3		1.1	0.3
2	0		1	1.0	0.3		3.1	0.8
3	2		3	2.3	0.8		3.2	0.8

System E								
R	G		CG	DCG	NDCG		PCC	NPCC
1	0		0	0.0	0.0		3.0	0.8
2	2		2	2.0	0.7		3.3	0.8
3	1		3	2.6	0.9		3.6	0.9

System F								
R	G		CG	DCG	NDCG		PCC	NPCC
1	0		0	0.0	0.0		3.0	0.8
2	1		1	1.0	0.3		3.7	1.0
3	2		3	2.3	0.8		3.9	1.0

Assuming all the documents have equal prior probability of relevance, the ranking is calculated by a likelihood function as follows: [21]:

$$P(q|d_m) = P(\{q^{(1)}, q^{(2)}, \dots, q^{(T)}\}|d_m) = \prod_{t=1}^T P(q^{(t)}|d_m) . \quad (85)$$

The remaining portion of this section outlines the steps to obtain the differentiable NPCC performance measure [38].

• **Retrieval ranking rule.** Our document retrieval system is constructed using a statistical document model instead of a vector space model [87]. Once the set of document models has been estimated, the document retrieval system must determine whether one document is ranked higher than the other document according to the following ranking rule for a given query q .

$$d_k \succ d_m \quad \longleftrightarrow \quad g(q, d_k) > g(q, d_m) , \quad (86)$$

where $g(q, d_m)$ returns the ranking score for the document d_m given the query q and is defined as

$$g(q, d_m) = \prod_{t=1}^T P_{JM}(q^{(t)}|d_m) , \quad (87)$$

where $P_{JM}(q^{(t)}|d_m)$ can be computed from (84). This ranking score function, along with the developed multi-level performance criterion, is used during the optimization of the document models.

Upon retrieving a set of documents for a given query, human supervisors can examine and associate each retrieved document with an irrelevance judgment (cost). A higher cost indicates that the retrieved document is more irrelevant. Thus ideally, documents with lower costs should be retrieved before those with higher costs. Notice, however, that the irrelevance judgment is independent of the rank of the document. The actual rank of a retrieved document is determined by its ranking score, while the irrelevance judgment is decided by a human supervisor.

• **Retrieval performance measure.** The performance measure used in this study is a multi-level irrelevance measure represented by the NPCC. As explained previously,

the retrieved documents are evaluated not only by their ranks but also by their costs. Ideally, we want the cost for all retrieved documents to be as small as possible. The *average cumulative cost* performance measure, which is the sum of the multi-level irrelevance judgments of the training queries, is defined as follows:

$$\overline{CC} = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M \text{irrel}(q_n, d_m), \quad (88)$$

where $\text{irrel}(q_n, d_m)$ is the multi-level irrelevance judgment of retrieving the document d_m for a given query q_n .

The CC value is not a useful performance measure because the CC performance measure is not affected by the ranking of the retrieved results. That is, retrieving a more irrelevant document before a less irrelevant document does not change the CC. In the PCC performance measure, however, documents appearing lower in the retrieval list will be penalized less as the multi-level irrelevance value is increased in proportion to the position of the result. The retrieval list can also be truncated to K elements. The *average penalized cumulative cost*, which is the PCC for all training queries, can be defined as follows:

$$\overline{PCC@K} = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M \text{irrel}(q_n, d_m) \cdot (K - \text{rank}(q_n, d_m) + 1), \quad (89)$$

where $\text{irrel}(q_n, d_m)$ indicates the irrelevance of document d_m , and $\text{rank}(q_n, d_m)$ indicates the rank for the document d_m in the retrieval list of length K .

Normalized PCC (NPCC) can be obtained by dividing PCC by a normalizing factor in such a way that the worst possible ranking list has the value of one. The normalizing factor can be obtained in the following fashion. First, the cost is sorted in a descending order starting from the highest irrelevance score. Now, the higher the ranking is, the lower the irrelevance judgment is. The normalizing constant is found by calculating the PCC using this sorted retrieval list. The NPCC can be obtained using the normalizing factor, and the PCC can be calculated using the unsorted retrieval list. Finally, the NPCC at a given truncation level K for all training queries,

denoted by $\overline{NPCC@K}$, can be defined as follows:

$$\overline{NPCC@K} = \sum_{n=1}^N \frac{Z_{nK}}{N} \sum_{m=1}^M \epsilon(q_n, d_m), \quad (90)$$

where Z_{nK} is a normalizing factor, and $\epsilon(q_n, d_m)$ represents both the cost factor $\text{irrel}(q_n, d_m)$ and the penalizing function $(K - \text{rank}(q_n, d_m) + 1)$.

• **Incorporating the ranking rule into the performance measure.** Our retrieval system is concerned with the minimization of the expected error cost. The performance is based on the expected PCC, assuming K retrieved documents for each query. This expected value can be defined as

$$E[NPCC@K] = E \left[Z_{nK} \sum_{m=1}^M \epsilon(q_n, d_m) \right]. \quad (91)$$

Given a set of training queries, the PCC is computed as

$$\overline{NPCC@K} = \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K \frac{Z_{nK}}{N} \epsilon(q_n, d_m) \cdot 1(d_m \succ \mathcal{D}) \cdot 1(\text{rank}(q_n, d_m) = k), \quad (92)$$

where the first and the second indicator functions are associated with the irrelevance cost factor and penalizing factor respectively. The first indicator function assumes the value of one if the document d_m is retrieved first among all existing documents. The second indicator function indicates the decision of the retrieval system to assign rank k to the document d_m , i.e., $\text{rank}(q_n, d_m) = k$.

The penalizing factor Z_{nK} remains the same for all documents given a query. However, the cost factor depends on the degree of irrelevance between the query and the retrieved document, and the position of the document in the list. For example, the first retrieved document has the position penalty of K , and the second retrieved document has the position penalty of $K - 1$, and so on. If the retrieved document has the largest scoring value, then the cost of retrieving that document has to be fully applied. Otherwise, the cost of retrieving a document will be discounted according to its position in the list and the degree of irrelevance.

The use of two indicator functions is possible because the irrelevance judgment is independent of the rank of a document. Bear in mind that the rank of a document is determined by the scoring function, while the irrelevance judgment is decided by the human supervisor.

• **A differentiable ranking performance measure.** The cost and the penalty factor are represented by

$$\overline{NPCC@K} = \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K \frac{Z_{nK}}{N} \epsilon(q_n, d_m) \left\{ \frac{g(q_n, d_m)}{[\sum_j g(q_n, d_j)^\eta]^\frac{1}{\eta}} \right\}^\gamma \cdot 1(\text{rank}(q_n, d_m) = k). \quad (93)$$

We use a smooth approximation for the indicator function. Notice that as the design parameter η goes to infinity $\eta \rightarrow \infty$ and the design parameter γ goes to infinity $\gamma \rightarrow \infty$, the approximation then becomes

$$\left\{ \frac{g(q_n, d_m)}{[\sum_j g(q_n, d_j)^\eta]^\frac{1}{\eta}} \right\}^\gamma = \begin{cases} 1, & \text{if } g(q_n, d_m) = \max_j g(q_n, d_j) \\ 0, & \text{otherwise,} \end{cases} \quad (94)$$

• **Optimization method for the system cost function.** The expectation operation is calculated over the entire training set, including queries $\{q_n\}_{n=1}^N$ and documents $\{d_m\}_{m=1}^M$. Notably, the error-cost function (93) is considered the continuous-valued representation for the number of rank errors in (92). The document model is iteratively updated by a descent algorithm

$$\Lambda_{t+1}^{NPCC} = \Lambda_t^{NPCC} + \varepsilon_t U_t \nabla \ell(q_n; \Lambda_t^{NPCC}), \quad (95)$$

where U_t is a positive definite matrix used to speed up the convergence rate. Starting from an initial model Λ_0^{NPCC} , parameter optimization is completed when the convergence condition is met. The gradient in (95) is computed as follows:

$$\nabla \ell(q_n; \Lambda_t^{NPCC}) = \frac{\partial \ell(q_n; \Lambda_t^{NPCC})}{\partial g(q_n, d_m)} \frac{\partial g(q_n, d_m)}{\partial \lambda}, \quad (96)$$

where $g(q_n, d_m) = \log P(q_n|d_m)$ indicates the ranking function defined in (84). For simplicity, we use the notations $b_m = P(q_n|d_m)$ and $G = [\sum_j g(q_n, d_j)^\eta]^\frac{1}{\eta}$. Then, the

gradient can be written as follows:

$$\begin{aligned}\frac{\partial \ell_n^{NPCC}}{\partial \log(b_m)} &= \frac{\gamma \epsilon_{nm}}{G} \left(\frac{b_m}{G}\right)^{\gamma-1} - \sum_r \gamma \epsilon_{nr} \frac{b_m^{\eta-1}}{G^\eta} \left(\frac{b_r}{G}\right)^\gamma, \\ \frac{\partial \log(b_m)}{\partial \lambda} &= \sum_{t=1}^T \frac{\hat{P}_{ML}(q_n^{(t)}|d_m) - \bar{P}_{BK}(q_n^{(t)}|\mathcal{D})}{\lambda \hat{P}_{ML}(q_n^{(t)}|d_m) + (1-\lambda) \bar{P}_{BK}(q_n^{(t)}|\mathcal{D})},\end{aligned}\quad (97)$$

where $\hat{P}_{ML}(q_n^{(t)}|d_m)$ indicates the ML estimation of the term probability as defined in (83), and $\bar{P}_{BK}(q_n^{(t)}|\mathcal{D})$ is the background collection model. As indicated in (97), the update of the language model is proportional to the difference between the retrieval score of the current document and that of the background language model. The estimated language model minimizes the ranking cost and attempts to reduce the ranking errors for a retrieval application.

3.2.3 Experimental Results

To evaluate the effectiveness of our proposed multi-level irrelevance-based retrieval system, we have evaluated our approach in a document retrieval application. The experiments are conducted on using the 20 Newsgroups collection [66]. This public-domain collection contains approximately 18,828 documents and 61,188 words, and is partitioned evenly across 20 different categories with approximately 1,000 documents per category. Categories in the 20 Newsgroups dataset are listed in Table 16. The sub-categories of a top-category are split into three subsets: training, validation, and testing datasets.

Categories in the 20 Newsgroups dataset are arranged in a hierarchical structure. The database includes six major categories : computer, science, recreation, talk, religion, and miscellaneous category. Each of them can be grouped into a total of 20 sub-categories. For example, the computer category can be separated into five different sub-categories such as *graphics*, *window OS*, *X window*, *IBM hardware*, and *Macintosh hardware*. We adopted the top two hierarchical levels to designate the relevance judgments. A relevance value of two is assigned to documents from the

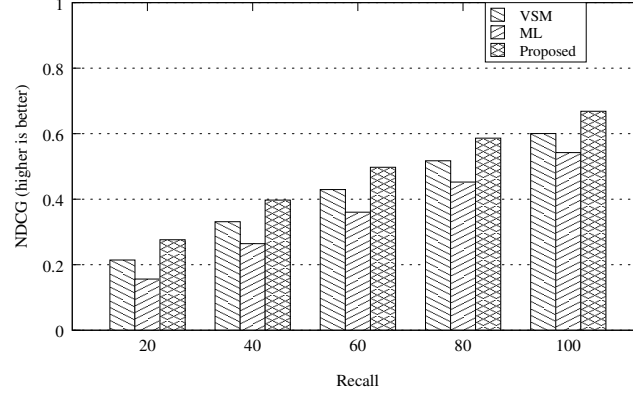
Table 16: Degree of relevance for the 20 Newsgroups dataset is assigned based on the following tri-relevance measure: a relevance of two for documents in the same sub-categories, a relevance of one for documents in the same major category, and a relevance of zero otherwise.

20 Newsgroups	
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey
sci.crypt sci.electronics sci.med sci.space misc.forsale	talk.religion.misc alt.atheism soc.religion.christian
	talk.politics.misc talk.politics.guns talk.politics.mideast

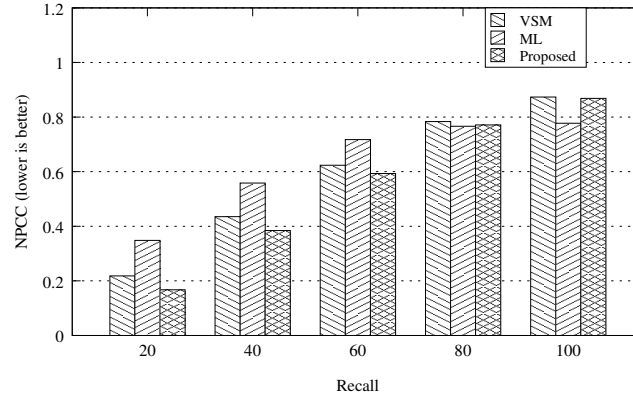
same sub-category. A relevance value of one is assigned to documents from the same major category.

The proposed approach is compared with the traditional retrieval techniques such as the Vector Space Model (VSM) and the Maximum Likelihood (ML) language model-based retrieval systems. The VSM baseline method represents the query and documents in vectors using the product of Term Frequency (TF) and Inverse Document Frequency (IDF) [87]. The ranking function is the cosine similarity. The ML baseline model uses the conventional language-model based document retrieval [84], where the ML estimate of each document model is obtained using uni-grams. Each of the document models is smoothed with the JM interpolation [117]. Figures 19(a), (b) and (c) show the NDCG, NPCC and precision performance measure, respectively, for the 20 Newsgroups dataset on several recall operating points, i.e., operating at 20%, 40%, 60%, 80% and 100% recall points.

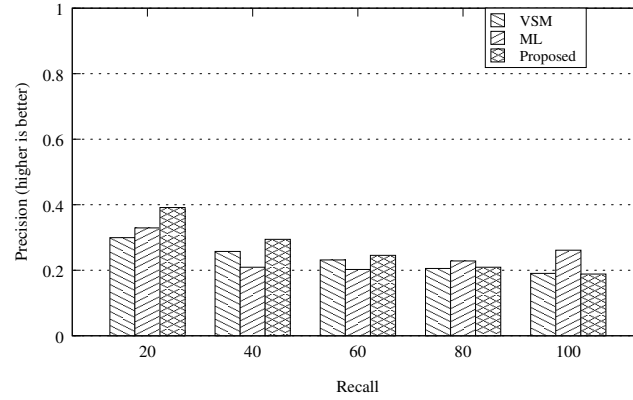
Figure 19(a) shows that the proposed approach consistently outperforms VSM and ML retrieval systems across all recall operating points. Although the VSM system also perform better than the ML approach in terms of the NDCG criterion, the proposed approach achieves better gain than the VSM retrieval system. Figure 19(b) shows the



(a) The plot of the discounted cumulative gain (DCG) across different recall points.



(b) The plot of the penalized cumulative cost (PCC) across different recall points.



(c) The plot of the precision across different recall points.

Figure 19: Performance measures of three different retrieval systems across various recall points on the 20 Newsgroups dataset. The DCGs of the proposed approach are higher than others across all recall points. The PCCs of the proposed approach are lower at low recall points and become similar to other approaches at higher recall points. The higher precision of the proposed approach is the indirect results of the proposed learning algorithm.

NPCC criterion, which is used as the optimization objective function. Notice that the proposed approach consistently has lower cost than the VSM and ML retrieval systems. The cost is noticeably lower at the 20% operating point, and the cost is less noticeable at 100% recall operating point. Figure 19(c) shows that the precision of the proposed approach is the highest when the recall is 20% and slowly decreases for higher recall. This is due to the penalizing factor in the NPCC criterion that penalizes a long retrieval list. This behavior is desirable when a shorter retrieval list is preferred over a longer retrieval list. Compared to the VSM approach, the precision of the ML approach is higher at 20%, 80%, and 100%, and lower at 40% and 60% recall operating points. This indicates that the ML and VSM approaches are comparable in terms of precision performance measures.

In summary, we conclude that the use of PCC is important as the optimization objective function because it improves the retrieval system for both DCG and precision performance measure. To accomplish this scenario, the PCC incorporates the retrieval ranking and user judgments into the retrieval performance measure. A smooth version of the PCC objective function can be formulated and used to optimize the statistical language model.

3.3 Summary

This chapter investigates the incorporation of subjective judgments into a statistical model using the error-cost learning procedure. The developed learning procedure can be used for both pattern recognition and retrieval-based MT systems. In the context of pattern recognition systems, the statistical model are constructed for each category and the subjective judgments indicate the preferences for one type rather than the other type of errors. In the context of retrieval-based MT systems, this statistical model represents a set of sensible variations for each original translation example and the subjective judgments are represented by graded levels of relevances.

In other words, two different approaches to incorporate the subjective judgments are examined in this chapter.

The first approach is to use the cost matrix to compactly represent the subjective preference and the confusion matrix to evaluate the system performance. The row and column conventions are the same for both the cost matrix and the confusion matrix, with the rows of the matrix representing the system decisions and the columns representing the true observation labels. While each entry in the cost matrix indicates the cost of misclassifying from one class and another class, each entry in the confusion matrix indicates the actual number of confusions between the two classes in the training data. The proposed error-cost learning procedure can actively influence specific entries of the confusion matrix by changing the corresponding entries in the cost matrix. This property allows designers to actively influence the statistical models based on the designers preferences.

The second approach incorporates subjective judgments into a novel multilevel irrelevant performance measure. In addition to the subjective judgments, the performance measure also incorporates information on the retrieval ranking and information on the length of retrieval list. By evaluating the novel performance measure, one can determine the merit of using a specific length of the retrieval list. When this novel performance measure is used, the overall performance of the statistical model can be evaluated and optimized according to the subjective preferences. In other words, the designers can directly influence the retrieval model so that more relevant examples appear at the top of the retrieval list.

CHAPTER IV

RETRIEVAL-BASED MACHINE TRANSLATION SYSTEMS

The MT system developed in this dissertation uses statistical pattern recognition techniques during the translation process. The translation task is defined as the preservation of meaning when a notion is rendered in a sequence of words from one language to another. The translation task requires understanding of linguistic rules as well as linguistic exceptions, i.e., cases where rules do not apply, in both source and target languages. Traditional MT systems rely on an extensive parsing strategy to decode the linguistic rules, and use a knowledge base to encode linguistic exceptions. However, the interactions among various linguistic exceptions become harder to analyze as the translation system becomes larger [102, 103]. To provide information on linguistic exceptions for an MT system, Nagao proposed using real translation examples instead of using a manually-crafted knowledge base [77]. This design strategy is now known as the Example-Based Machine Translation (EBMT) principle.

Our developed retrieval-based MT system is a variation of the EBMT principle, which, in turn, is an extension of the conventional rule-based MT paradigm. A main distinction of the proposed retrieval-based MT system is the use of a sentence-level translation unit, while the EBMT principle can be applied at different levels of abstractions, e.g., word, phrase, or sentence-level abstractions. The choice of the translation unit determines the necessary components of an MT system and dictates the compromises among different qualities of an MT system. The advantage of using a sentence-level translation unit is that the boundary of a sentence is explicitly defined and the semantic or meaning of both the source and target sentences is precise. This

advantage is particularly useful in domain-specific applications, which usually involve short sentences. For the translation of an input text, source language parsing and target language recombination are optional in a sentence-level MT system. On the other hand, using word-level translation units usually requires a lot of effort in source language parsing and target language recombination.

A main drawback to using a sentential translation unit is the limited coverage, i.e., the difficulty of finding an exact match between a user query and sentences in a database. To extend the coverage of a retrieval-based MT system, this study investigates the introduction of a set of sensible variations for each translation example in the database. Ideally, each sensible variation is obtained by rewriting (paraphrasing) the source sentence in the database. Paraphrasing, however, is a very labor-intensive and time-consuming task. In this study, we use a computationally efficient procedure to generate sensible variations of the source sentences. Roughly speaking, our generation procedure can be described as follows: First, we construct a word vocabulary list from the source sentences and convert the source sentences into feature vectors. Second, the word-to-word associations are identified from the vocabulary list using an electronic dictionary (WordNet [33]) and organized into an association matrix. The association matrix and the source feature vectors are used to compute a background feature vector. Then, a topic model is constructed using the source feature vectors and the background feature vectors. Using the topic model, each source feature vector can be represented as a mixture of topic distributions. Finally, a set of sensible variations for each source feature vector is generated by sampling from the mixture of topic distributions. The developed generation procedure improves our retrieval-based MT system through better feature representations. Our experimental results show that the retrieval-based MT system is competitive with conventional MT systems for domain-specific translation tasks.

The rest of this chapter is organized as follows: Section 4.1 explains the compromises in the design of MT systems. Section 4.2 describes several design choices in constructing our retrieval-based MT system. We also discuss the use of an electronic dictionary (WordNet) to obtain better coverage for our retrieval-based MT system. Section 4.3 shows the experimental results of the developed retrieval-based MT system. Section 4.4 summarizes our study on the retrieval-based MT system.

4.1 Design Compromises in Machine Translation Systems

Conventional MT systems are usually designed based on a pre-specified paradigm such as statistical, example-based, and rule-based translation paradigms [55]. Each of these paradigms has its own advantages and disadvantages. Many MT researchers believe that next generation MT systems will use hybrid paradigms, for example, a hybrid of statistical and example-based paradigms, the hybrid of statistical and rule-based paradigms, or a hybrid of rule-based or example-based paradigms [1, 19, 94, 100]. The main challenge is to determine which aspects of an MT system are best approached by the statistical paradigm, which by the example-based paradigm, and which by rule-based paradigm. In addition, many challenging issues arise when designing a hybrid system, such as how to find the right combination of each component, how to integrate different methods, and how to evaluate the success of MT systems. To distinguish among different designs of an MT system, researchers have identified three important factors as illustrated in Figure 20 [16, 112].

Figure 20 depicts various MT models according to the degree of example-based, compositional, and statistical techniques [112]. The x -axis represents the degree to which the rule generalization/abstraction is performed on the training examples to obtain translation templates. Scheme-based MT systems mainly use translation templates and example-based MT systems mainly use translation examples. The y -axis represents the degree to which compositional rules are required. In a loose sense,

compositional systems are predicated on the belief that the meaning of a long expression is a function of the meaning of its components and the way they are combined syntactically. Lexical systems require simpler compositional rules than collocational (phrasal) systems. MT systems with alignment and semantic models require more sophisticated compositional rules than collocational systems. The z -axis represents the degree to which statistics and statistical inference are used. In traditional machine translation systems, logical (set theoretic) rules are usually used instead of statistical rules. Note that statistical models are inherently logical models because the probability theory can be derived from the set theory. However, the converse does not hold — not all logical models are statistical models.

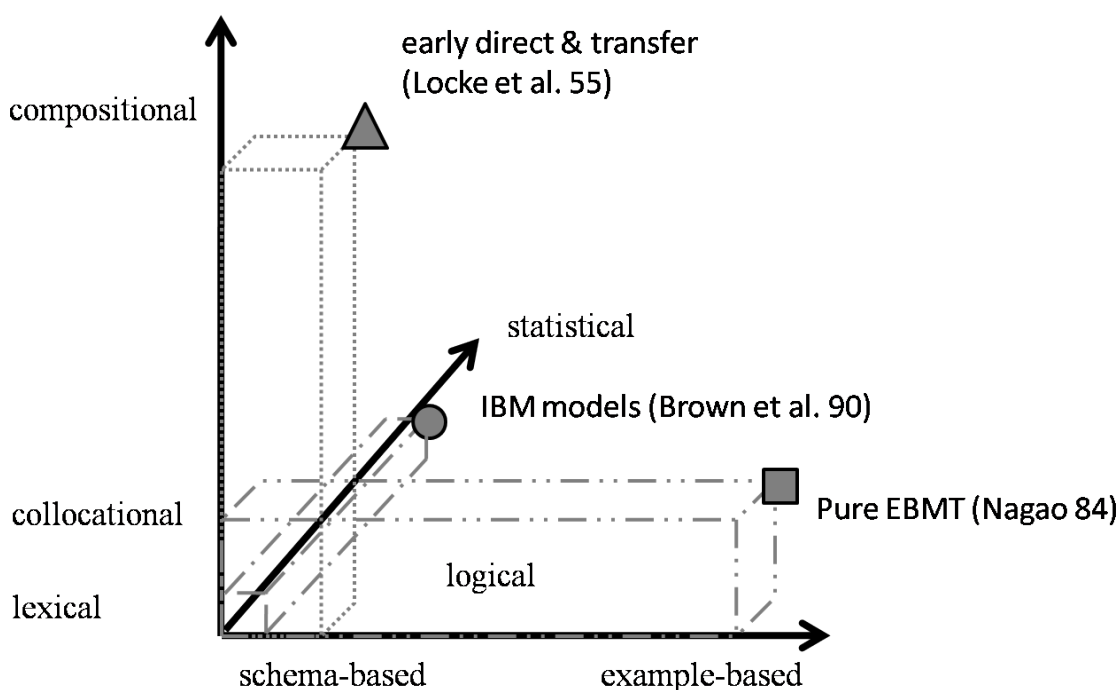


Figure 20: The space of machine translation models

The degree of each factor used in MT designs varies from one system to another depending on the ultimate applications of the MT systems [16]. A traditional word-by-word translation system can be positioned at the origin of the three-dimensional space because it employs schema-based translation with logical operation on lexical

word forms. The original IBM source-channel model used a very simple word-to-word lexical translation model without incorporating translation examples. Thus, the original IBM model is a statistical, lexical, and schema-based MT system [11]. Nagao's original EBMT system was characterized by the use of a large library of translation examples for lexical collocations [77]. Thus, Nagao's original EBMT system is a collocational example-based MT system.

The three-dimensional model space provides a clear grouping of existing MT systems. However, the model space may give the wrong impression that there is only one optimal design of an MT system [17]. The design of an MT system varies in practice according to the system's intended applications such as for web browsing or for law translations. Moreover, designing an MT system involves other issues such as the organization of the database and the choice of online or offline processing. The three-dimensional model space conveys several important messages. First, the design of an MT system involves making compromises, e.g., using schema-based, example-based, or hybrid-based systems. Second, the choice of translation units affects the choice of the compositional rules and, thus, is a major design factor in an MT system. The three-dimensional model space does not indicate what the challenges of converting from one paradigm to another are or how the translation qualities are influenced by the length of the translation units. In the remaining portion of this section, we discuss the compromises among translation qualities affected by the length of the translation units.

Compromise is inevitable in the design of an MT system and determines the quality of the final MT system. The quality of an MT system can be differentiated into internal and external quality requirements [18]. The external quality requirements are represented by two independent factors, which are defined as follows:

Coverage indicates the variety of topics or subjects the system can handle. A low coverage system usually deals with restricted topics with specific terminology,

and a high coverage system can handle a great variety of subjects.

Reliability refers to the degree of success of the translation results judged by human users because any MT system will eventually be used by people. Reliable MT has to fulfill the end users' expectations. Typical user expectations vary from one case to another. For example, *informative* translation allows the user to understand more or less the content of the source text. *Literal* translation provides a target text in a correct grammatical form. *Reliable* translation is semantically, idiomatically, and stylistically correct. *User-oriented* translation should be correct from the standpoint of a pre-specified user.

An MT system with restricted coverage and low reliability is generally considered not useful. MT systems with good coverage and lower reliability are acceptable for leisure purposes such as for internet browsing. MT systems with restricted coverage and high reliability are required for demanding tasks such as law translation. Although a system with good coverage and high reliability is desirable, such a system is still not feasible at the current time. Reliable systems usually require long translation units, which decrease the system coverage. On the other hand, good coverage is usually achieved through short translation units [18]. Identifying short translation units is usually technically difficult because it depends on stringent parsing technology to uncover the language structure.

Automatic MT evaluation is usually based on the BiLingual Evaluation Understudy (BLEU) metric, which is one of the first metrics to achieve a high correlation with human judgments of quality [81]. The BLEU metric and its variants, i.e., NIST, ROUGE, METEOR rely on the comparison of n -grams between the MT outputs and the reference translations. An advantage of using the BLEU metric lies mainly in the absence of human involvement while giving clear quantitative results. Several researchers have warned against over-reliance on the BLEU metric [13] in part because an increase in BLEU scores does not necessarily mean improvement in human

judgment. Nevertheless, the use of the BLEU metric is appropriate for tracking incremental changes in a single system with similar translation strategies [15]. Our work in this dissertation uses the BLEU metric for automatic MT evaluations.

The BLEU metric is the product of a *brevity penalty* and a *precision score* [81]. The *brevity penalty* penalizes shorter translations compared to reference translations in proportion to the comparative brevity. The brevity penalty BP is defined as

$$BP = \begin{cases} 1 & \text{if } L_{ref} < L_{sys} \\ \exp\left(1 - \frac{L_{ref}}{L_{sys}}\right) & \text{otherwise} \end{cases},$$

where L_{sys} are the length of the translation output and L_{ref} is the length of the closest reference translation to the system's output. The *precision score* is derived from counting the number of N -gram matches between the candidate translation and the reference translations. Translation that is identical to any of the reference translations gives a score of one. The n -gram precision is defined as

$$\text{prec}_n = \frac{\sum_{i=1}^I \sum_{n\text{-gram} \in s_i} \text{count}(n\text{-gram})}{\sum_{i=1}^I \sum_{n\text{-gram} \in s_i} \text{count}_{sys}(n\text{-gram})},$$

where $\text{count}(n\text{-gram})$ is the number of n -gram found both in the systems output and in the corresponding reference translations, and $\text{count}_{sys}(n\text{-gram})$ is the number of n -gram found in the system's output. The BLEU metric is defined as follows:

$$BLEU = BP \times \exp\left(\sum_{n=1}^N \frac{\log(\text{prec}_n)}{N}\right), \quad (98)$$

where N is the maximum n -gram size considered, BP is a brevity penalty, and prec_n is the n -gram precision. The range of BLEU score is between zero and one measuring the statistical closeness to a set of reference translations. The BLEU metric is basically the geometric mean of the n -gram co-occurrences between the system's output and the set of reference translations.

MT developers are concerned with the internal quality requirements in addition to the external quality requirements. The internal quality requirements are separately evaluated in source and target texts and can be defined as follows:

Precision indicates the degree of matching between the source text and translation units stored in the system. Precision is usually computed using a set of pre-specified reference texts.

Adaptability indicates the capability of translation units to properly fit into a target text. For short translation units, additional preprocessing steps are necessary to make the final text reliable because a concatenation of a number of sub-sentence translations does not necessarily give a valid translation. Fewer processing steps are required to fit longer translation units into the final target text.

Convolved relations between internal and external quality requirements dictate the design of an MT system. Different designs are required depending on whether higher reliability or broader coverage is desirable. Roughly speaking, the quality of an MT system can be defined as "Coverage * Reliability = \mathfrak{R} " [8], where \mathfrak{R} is a factor that depends on the MT technology and the amount of effort one is willing to invest. Advancements in computing technology or the availability of a large number of resources can result in broader coverage and a more reliable MT system. For a fixed \mathfrak{R} , one can expect either broader coverage and lower reliability MT systems, or narrower coverage and higher reliability MT systems.

In summary, many researchers acknowledge that the design of an MT system involves various design choices and compromises among different quality requirements [16]. Table 17 summarizes the compromises among external and internal qualities due to the lengths of translation units. The choice of translation units determines the requirement of compositional rules to obtain a reliable target text. Short translation units reduce the concern over the coverage of the source text and improve the adaptability of the target text. However, short translation units make it difficult to achieve high reliability of the target text and high precision of the source text. On the other hand, using long translation units improves the reliability of the target text and the precision of the source text because of the uniqueness of the expression retrieved.

However, long translation units deteriorate the coverage of the source text and the adaptability of the target text.

Table 17: Compromises among external and internal qualities due to the lengths of translation units

Lengths of Translation Units	External Qualities		Internal Qualities	
	Coverage	Reliability	Precision	Adaptability
short	good	low	low	high
long	restricted	high	high	low

4.2 *Retrieval-Based Machine Translation System*

This dissertation focuses on the design of an MT system using statistical pattern recognition techniques, which we refer to as a retrieval-based MT system. An overview of different MT paradigms such as the RBMT, EBMT, and SMT paradigms is provided in Section 1.1.3. This section describes the retrieval-based MT system in detail, emphasizing the differences between the proposed retrieval-based MT system and the conventional EBMT systems. First, we explain the defining characteristic of an EBMT system. Then, we describe the design choices and explain the design compromises in our retrieval-based MT system. Finally, we propose a solution to deal with the main weakness of a retrieval-based MT system, which is identified as limited coverage.

Although the EBMT paradigm is widely used in practice [54], the precise specification of what constitutes an EBMT system remains a debatable topic. According to Somers [96], the condition necessary for a system to be regarded as an EBMT system is that it be “primarily example-based”, i.e. mostly data-driven as opposed to theory-driven. Somers envisions the EBMT paradigm as complementing, enhancing, and, sometimes, replacing the RBMT paradigm. In other words, some phenomena may be suited to EBMT while others are better tackled by RBMT [103]. Turcato and Popowich [105] compare a variety of EBMT systems to RBMT systems and conclude that almost all of the techniques used in EBMT are also used in RBMT. The

main distinction is the “translation by analogy” principle as originally suggested by Nagao [77]. Translation by analogy implies that we cannot *a priori* know which parts in an example are relevant for the translation of a new sentence [17]. If this is the case, any preprocessing and decomposition of the training examples will eventually make EBMT systems appear to resemble traditional RBMT systems. According to Hutchins [54], the main step in an EBMT system is the matching of source-language fragments and extraction of equivalent target-language fragments as partial potential translations. Neither the structure of the fragments nor the preprocessing step is the defining criterion for an EBMT system. The defining procedure is the matching step because the matching procedure is carried out with reference to paired source and target sentences [17].

This dissertation uses the EBMT paradigm as originally outlined by Nagao [95] and is illustrated in Figure 21. A typical EBMT system consists of the following three main steps:

1. **The matching step** matches the input query sentence against a set of translation examples. The matching procedure is important because the matching results are used by the subsequent steps in an EBMT. This step is equivalent to the *analysis step* in a conventional RBMT system.
2. **The alignment step** identifies the corresponding translation fragments for both the source and target text. Once the relevant examples of the source text are retrieved, the corresponding fragment of the target text can also be retrieved. This step is equivalent to the *transfer step* in a conventional RBMT system.
3. **The recombination step** combines the translation fragments to produce a target text and also puts the finishing touches on the output target text. This step is equivalent to the *generation step* in a conventional RBMT system.

Each step in EBMT paradigms has its own challenges. For example, coverage is the main issue in the matching step. Determining the boundaries of a translation unit is challenging in the alignment step [20]. The challenging issue in the recombination step is boundary friction problems, which are concerned with joining and smoothing of the translation units (fragments) in the target text [12].

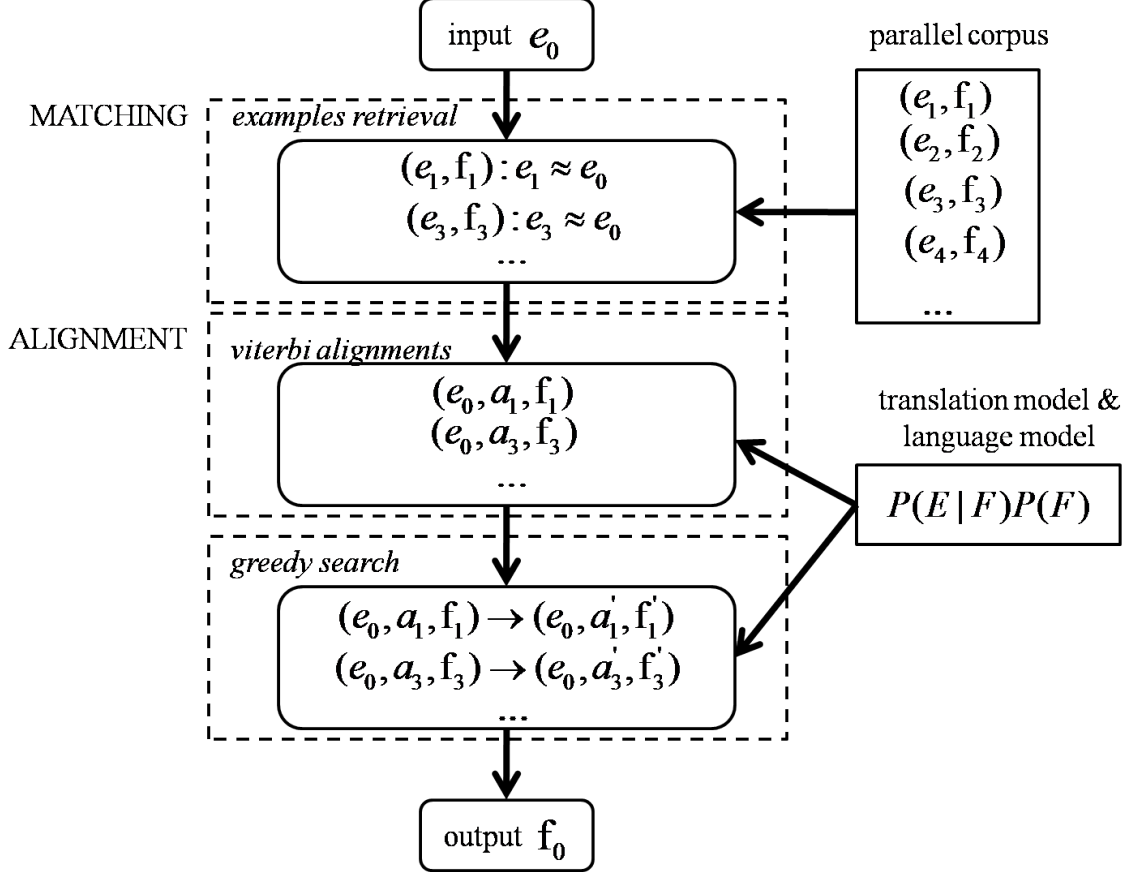


Figure 21: Three major steps in a conventional EBMT system. The retrieval-based MT steps are shown in lower-case; those for EBMT are in upper-case [109].

Our retrieval-based MT system is actually a variation of an EBMT system with the main distinction being the choice of the “translation unit”. Figure 22 illustrates the five levels of syntactic structures: word, phrase, clause, sentence, discourse. The main advantage of using sentential units is that the semantic unit(meaning) is well preserved within the boundaries [101]. The main drawback of using sentential units

is that the coverage of each unit is narrow. Many researchers believe that the potential of the EBMT paradigm can be fully exploited if a "sub-sentence" is used as the translation unit [89]. The main drawback of using sub-sentential translation units is that incorrect identification of the sub-sentence boundary result in low translation quality for the whole sentence [78]. Moreover, boundary determination is more complicated in translation problems with sub-sentence units because accurate boundary identifications have to be determined in both the target and source texts [62].



Figure 22: Five levels of syntactic structures: word, phrase, clause, sentence, discourse.

Similarity measures for an MT system are usually chosen based on the text patterns they are applied to. For example, word-based metrics compare individual words of the two sentences in terms of their synonyms, hyponyms, or antonyms [78]. Although the word-based metrics are the most popular, other approaches include syntax based metrics [113] and character-based metrics [88]. A syntax-based metric tries to capture the similarity of two sentences at the syntax level. A character-based metric is usually applied to Chinese, Japanese and Korean languages. Hybrid-based metrics have also been applied. These metrics combine the strength of the word, syntax, and character-based metric [40]. Our retrieval-based MT system converts the sentential translation units into word occurrence vectors and computes the "cosine similarity" between two feature vectors.

The coverage of sentential translation units is narrower than the coverage of sub-sentential translation units. To improve the coverage in our sentential translation system, we incorporate information from a dictionary into the EBMT system by addressing two main issues: (i) One word may have multiple variations (e.g., work, works, and worked), and multiple words may have the same meaning (e.g., house, mansion and manor). (ii) A procedure is needed to discriminate one sentence from another. As far as issue (i) is concerned, the obvious choice is to obtain synonymous word forms from a lexical knowledge source such as the electronic dictionary WordNet because the WordNet dictionary is organized into synonym sets. To address issue (ii), a value or weight can be assigned to each word even though the translation system operates at the sentence level not at the phrase or word level. The challenging issue is to determine the exact value to assign so that the discrimination between sentences can be carried out.

Our retrieval-based MT system uses the PLSA procedure to generate sensible variations for each original example. Once sensible variations for each document in the original collection $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ have been generated, the classification and retrieval procedures developed in Chapter 2 and Chapter 3 respectively, can be used in the retrieval-based MT system. At a high-level, the document generation procedure involves the following four steps:

1. Construct a term-document matrix and a word list (dictionary) from the original source collection.
2. Construct a term-to-term association matrix from an electronic dictionary (WordNet).
3. Identify the topic for each source document in the original source collection.
4. Sample sensible variations for each source document according to the topics identified in each source document.

The first step is straightforward and can be carried out using the standard procedures in natural language processing as described in Section 1.1.1. The rest of this section describes only step 2, step 3, and step 4 in detail. The construction of the term-to-term association matrix is based on the electronic dictionary WordNet. The term-to-term association matrix is used to initialize a background feature vector, which is used to generate (identify) the topic for each source document in the collection. The identified topics are used to generate additional variations of the original source document.

4.2.1 Construction of a term-to-term association matrix

The main distinction of our document generation procedure lies in the use of background knowledge from an electronic dictionary (WordNet [33]). Background knowledge from an electronic dictionary is represented by a term-to-term association matrix, which is used to increase the overlaps between two feature vectors. Many pattern recognition techniques are not successful in dealing with sparse feature vectors because conventional similarity measures require substantial overlaps in the feature vectors. Two main approaches addressing the problems raised by sparse feature vectors are as follows: First, we define new semantic similarity functions by means of external knowledge sources without changing the underlying document representation [9, 74, 75]. Second, we expand the feature vectors prior to using the conventional document similarity functions [4, 41, 52]. Our document generation procedure follows the latter approach.

Our document generation procedure expands each of the source documents using a term-to-term association matrix \mathbf{A} . Because the association matrix \mathbf{A} typically encodes pairwise term similarities, the mapping $\phi(\mathbf{e}_i) = \mathbf{A} \mathbf{e}_i$ allows us to represent each document not only by its original terms but also by the terms that are related to each of them. For example, if a classification algorithm uses an inner-product similarity measure $\text{sim}(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{e}_i' \mathbf{e}_j$, the similarity function using the association

matrix \mathbf{A} becomes $\text{sim}(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{e}_i' \mathbf{A}' \mathbf{A} \mathbf{e}_j$, where \mathbf{A}' indicates the transpose of the matrix \mathbf{A} . By varying the matrix \mathbf{A} one can obtain different transformations of the document feature space.

Term-to-term associations can be computed using various methods such as linguistic analysis, semantic term relationships, and statistical term co-occurrence [74, 82]. In our document generation procedure, the association matrix \mathbf{A} is computed using an electronic dictionary (WordNet) and the Leacock & Chodorow similarity measure, which is defined as

$$\mathbf{a}_{ij} = -\log \left(\frac{\text{length}(\mathbf{t}_i, \mathbf{t}_j)}{2 \times \text{max_length}} \right), \quad (99)$$

where $\text{length}(\mathbf{t}_i, \mathbf{t}_j)$ is the length of the shortest path between two synonym sets $(\mathbf{t}_i, \mathbf{t}_j)$ using a node counting procedure, max_length is a constant indicating the maximum depth of the WordNet taxonomy, and \mathbf{a}_{ij} denotes the i^{th} row and j^{th} column of the term-to-term association matrix \mathbf{A} . The term-to-term association matrix $\mathbf{A} = \{\mathbf{a}_{ij}\}_{i,j=1}^N$ is used to construct a background feature vector, which is defined as follows:

$$p_{\text{BK}}(\mathbf{t}_i | \mathcal{E}) = \frac{1}{N} \sum_{n=1}^N \sum_{j=1}^T \mathbf{a}_{ij} \mathbf{e}_n^{(j)}, \quad (100)$$

where $\mathbf{e}_n^{(j)}$ indicates the j^{th} term in the n^{th} source document \mathbf{e}_n , and \mathcal{E} denotes the source document collection $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$.

4.2.2 Identification of document topics

A document generation procedure is carried out by sampling from a mixture of multinomial distributions. Each document in the collection is assumed to be a mixture of multinomial distributions, and the mixture coefficients are estimated using a modification of the PLSA technique introduced in Section 1.1.1 so that it can take into account the background knowledge obtained from WordNet.

Specifically, let $\{z_1, \dots, z_k\}$ be K topic unigram language models (i.e., word distributions) and z_B be a background topic model for the whole collection \mathcal{E} . A word

\mathbf{t} in a document \mathbf{e} is regarded as a sample of the following mixture model [118, 73].

$$p(\mathbf{t}_i, \mathbf{e}_j) = \lambda_{\text{BK}} P_{\text{BK}}(\mathbf{t}_i | \mathcal{E}) + (1 - \lambda_{\text{BK}}) \sum_{k=1}^K p(\mathbf{t}_i | z_k) p(z_k | \mathbf{e}_j) , \quad (101)$$

where $P_{\text{BK}}(\mathbf{t}_i | \mathcal{E})$ is obtained from (100), and λ_{BK} is set empirically and indicates the amount of "background knowledge" common to all source texts. λ_{BK} and $P_{\text{BK}}(\mathbf{t}_i | \mathcal{E})$ are assumed to be available. The objective is to estimate the parameter set $\Lambda = \{P(\mathbf{t}_i | z_k), P(z_k | \mathbf{e}_j)\}$. We are going to use EM algorithm, similarly to the way it is used in the conventional PLSA technique, to maximize the the log-likelihood of the document collection \mathcal{D} , which is defined as

$$\log p(\mathcal{D} | \Lambda) = \sum_{i=1}^T \sum_{j=1}^N \text{count}(\mathbf{t}_i, \mathbf{e}_j) \log p(\mathbf{t}_i, \mathbf{e}_j) . \quad (102)$$

This formulation introduces two types of hidden variables for each word in the vocabulary. The first type of hidden variable indicates that the word \mathbf{t}_i in the source document \mathbf{e}_j is generated with the background model, i.e. $z_B = 1$. The second type of hidden variable indicates that the word \mathbf{t}_i in the source document \mathbf{e}_j is not generated with the background model, i.e. $z_B = 0$, but is instead generated using topic z_k for $k \in \{1, \dots, K\}$. The E-step in the EM algorithm is defined as

$$p^{(n)}(z_k | z_B = 0, \mathbf{t}_i, \mathbf{e}_j) = \frac{p^{(n)}(\mathbf{t}_i | z_k) p^{(n)}(z_k | \mathbf{e}_j)}{\sum_{l=1}^K p^{(n)}(\mathbf{t}_i | z_l) p^{(n)}(z_l | \mathbf{e}_j)} ,$$

$$p^{(n)}(z_B = 1 | \mathbf{t}_i, \mathbf{e}_j) = \frac{\lambda_{\text{BK}} p_{\text{BK}}(\mathbf{t}_i | \mathcal{E})}{\lambda_{\text{BK}} p_{\text{BK}}(\mathbf{t}_i | \mathcal{E}) + (1 - \lambda_{\text{BK}}) \sum_{k=1}^K p^{(n)}(\mathbf{t}_i | z_k) p^{(n)}(z_k | \mathbf{e}_j)} .$$

In the E-step, we are actually estimating the distribution of the hidden variables. A word could be separated into several fractions, with each fraction generated from a topic model or background model. This distribution is simple to compute: All that must be figured out is, in the likelihood of a word, how much proportion is contributed by the background model or by the topic z_k if the proportion is not contributed by the background model. Notice that the background feature vector $p_{\text{BK}}(\mathbf{t}_i | \mathcal{E})$ does not change during the EM algorithm because it represents the information that is

common to all source documents. The M-step in the EM algorithm is defined as

$$p^{(n+1)}(z_k|\mathbf{e}_j) = \frac{\sum_{i=1}^T \text{count}(\mathbf{t}_i, \mathbf{e}_j) [1 - p^{(n)}(z_B = 1|\mathbf{t}_i, \mathbf{e}_j)] p^{(n)}(z_k|\mathbf{t}_i, \mathbf{e}_j)}{\sum_{l=1}^K \sum_{i=1}^T \text{count}(\mathbf{t}_i, \mathbf{e}_j) [1 - p^{(n)}(z_B = 1|\mathbf{t}_i, \mathbf{e}_j)] p^{(n)}(z_l|\mathbf{t}_i, \mathbf{e}_j)} ,$$

$$p^{(n+1)}(\mathbf{t}_i|z_k) = \frac{\sum_{j=1}^N \text{count}(\mathbf{t}_i, \mathbf{e}_j) [1 - p^{(n)}(z_B = 1|\mathbf{t}_i, \mathbf{e}_j)] p^{(n)}(z_k|\mathbf{t}_i, \mathbf{e}_j)}{\sum_{i=1}^T \sum_{j=1}^N \text{count}(\mathbf{t}_i, \mathbf{e}_j) [1 - p^{(n)}(z_B = 1|\mathbf{t}_i, \mathbf{e}_j)] p^{(n)}(z_k|\mathbf{t}_i, \mathbf{e}_j)} ,$$

The M-step essentially aggregates such fractions to estimate a new set of values for Λ .

To estimate the topic weights for a document $P(z_k|\mathbf{e}_j)$, we simply aggregate all the fractions of words generated by topic z_k in document \mathbf{e}_j and normalize $\{P(z_k|\mathbf{e}_j)\}$ to make $\sum_{k=1}^K P(z_k|\mathbf{e}_j) = 1$. After performing the EM algorithm, we obtain the parameter set $\{P(\mathbf{t}_i|z_k), P(z_k|\mathbf{e}_j)\}_{k=1}^K$.

4.2.3 Generation of sensible variations for each source feature vector

Additional variations of the original translation examples $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ are generated by sampling from the topic distributions that are automatically discovered using the PLSA technique. The purpose of generating these additional variations is to improve the coverage of our retrieval-based MT system. In the PLSA technique, document \mathbf{e}_1 is represented as a mixture of topic distributions, which are defined to be conditionally independent multinomial distributions. Our generation process uses the asymmetric formulation of the PLSA technique, in which the co-occurrence of the word and document is modeled as

$$p(\mathbf{t}_i, \mathbf{e}_j) = p(\mathbf{e}_j) \sum_{k=1}^K p(\mathbf{t}_i|z_k) p(z_k|\mathbf{e}_j) , \quad (103)$$

where, for each document in the collection, a latent class z_k is chosen conditionally based on the document according to $p(z_k|\mathbf{e}_j)$ and a word is then generated from that class according to $p(\mathbf{t}_i|z_k)$. Alternative formulation of the PLSA technique includes a symmetric formulation, in which the co-occurrence of the word and document is modeled as

$$p(\mathbf{t}_i, \mathbf{e}_j) = \sum_{k=1}^K p(\mathbf{t}_i|z_k) p(\mathbf{e}_j|z_k) p(z_k) , \quad (104)$$

where the term \mathbf{t}_i and \mathbf{e}_j are assumed to be generated from the latent class z_k in similar ways using the conditional probabilities $p(\mathbf{t}_i|z_k)$ and $p(\mathbf{e}_j|z_k)$. Both symmetric and asymmetric formulations can be used to discover the topic model of a document collection [51].

Our document generation procedure assumes that each document in the collection is equally important and, thus, are more suitable to be modeled using the asymmetric formulation as defined in (103). The document generation procedure based on the asymmetric PLSA technique proceeds as follows:

1. Specify a source document \mathbf{e}_j to work on.
2. Choose a topic z_k according to the distribution $p(z_k|\mathbf{e}_j)$.
3. Choose a term \mathbf{t}_i according to the distribution $p(\mathbf{t}_i|z_k)$.

Imagine someone wants to write an article. He will first recall a previously read article from his memory. This corresponds to the first step in the document generation procedure, in which we identify the source document \mathbf{e}_j in \mathcal{E} . After that, he decides the relevant topics in the document according to the distribution $P(z_k|\mathbf{e}_j)$. Once the topic decision z_k has been made, he then chooses a term according to the distribution $P(\mathbf{t}_i|z_k)$. This three-step generation procedure is repeated for all the documents in the collection \mathcal{E} .

4.3 Experimental Results

Our experiments are geared toward domain-specific rather than general-purpose translation tasks. Domain-specific translation tasks are not as trivial as one might suspect as they are usually targeted at “exotic” languages with only a limited amount of training data available. A dataset containing parallel English and Indonesian sentences is manually constructed. For the experiments, the dataset contains 120 different semantic categories with 564 different expressions. On average, 4.7 different expressions

are available for each semantic category. The categories are selected based on a hotel reservation task. Table 18 lists a subset of the categories in the database including inquiries about reservation, billing, room amenities, and hotel amenities. Table 19 lists eight sample sentences pertaining to two different categories in the dataset. The first category is about getting a cheaper room, and the second category is about the hotel bill. The main distinction of a domain-specific translation is that all English sentences in the same category will be translated to the same Indonesian sentence because all English sentences in the same category convey the same semantic message.

Table 18: Semantic categories for the hotel reservation task in the database

reservation	bill	room amenities
reservation confirmation	prepare the bill	room bed
reservation cancellation	pay for the bill	room key
reservation extension	dispute the bill	room temperature
reservation changes	explain the bill	room lightning
reservation		room television
hotel amenities	telephone calls	misc
hot-tub	wake-up call	smoke detectors
internet access	call for taxi	fire exits
parking area	call for shuttle	checkout time
meeting room	call for security	
swimming pool		
laundry room		

For the proposed retrieval-based MT system, the pre-processing techniques explained in Section 1.1.1 are used to extract the semantic features from the source sentences (documents in the context of information retrieval). The document generation procedure described in Section 4.2 is used to obtain additional source sentences and to expand the coverage of our MT system. These additional source sentences are clustered with the original source sentences as the centroid using a conventional k-means clustering algorithm. Each cluster represents one document class for the classifier. The PLSA technique is used to reduce the dimension of the feature vectors, and the length of the feature vector is identified using validation datasets. Figure 23

lists the BLEU scores for our MT system across four different generation factors. The length of the feature vectors or the PLSA dimension is set to 100 for all classifiers reported in Figure 23.

To estimate the performance measure in the experiments, we randomly construct ten different subsets from the original dataset. The first five datasets are used as training datasets and the other five subsets as test datasets. Each subset consists of 250 parallel English-Indonesian sentences. Performance assessment is based on the average of these four subsets, and the performance measure is based on the BLEU metric as defined in (98). The x-axis in Figure 23 indicates the number of additional feature vectors that are generated for each source feature vector. The original 250 feature vectors are grouped together with additional 1250, 2500, 3750, and 5000 feature vectors corresponding to the generation factors of 5, 10, 15 and 20. The performance of our classifier is computed using test feature vectors of size 1500, 2750, 4000, and 5250, respectively.

Table 19: Examples of parallel English-Indonesian sentences used to train a domain-specific translation system for a hotel reservation purpose.

i want a cheaper room	=> saya ingin kamar yang lebih murah
a cheaper room please	=> saya ingin kamar yang lebih murah
cheaper room is wanted	=> saya ingin kamar yang lebih murah
do you have a cheaper room	=> saya ingin kamar yang lebih murah
i think that my bill has a mistake	=> saya merasa bahwa tagihan saya memiliki kesalahan
i think that a mistake has been made in my bill	=> saya merasa bahwa tagihan saya memiliki kesalahan
i think that there has been a mistake in my bill	=> saya merasa bahwa tagihan saya memiliki kesalahan
my bill has a mistake i think	=> saya merasa bahwa tagihan saya memiliki kesalahan

Three benchmark MT systems are used in this experiment. The first system is a

statistical MT system ¹, the second system is the Microsoft Bing Translator system ², and the third system is the Google Translate system ³. For the Google and Microsoft MT systems, we obtain Indonesian translation by directly supplying the test sentences to the systems and compare the translated sentences with the predefined sentences. The baseline statistical MT system is based on the procedure outlined in Figure 5. For the baseline, we use a five-gram language model that is constructed using “SRILM” [98], a phrase model that is extracted using “Thot” [80], and a beam-search decoder that is implemented in “Moses” [50]. We experiment with phrase lengths of three and five words in the phrase model. A shorter phrase length prevents the alignment algorithm from using the whole sentence as a phrase. This restriction usually results in lower translation qualities for the SMT system. The performance of the baseline statistical MT system is lower than those of the Google and Microsoft translation systems. Both Google and Microsoft translation systems use more training data than the baseline statistical MT system.

Figure 23 summarizes the results of using five different classification schemes in our retrieval-based MT system, including a nearest neighbor scheme and four margin-based classifiers. The four margin-based classifiers include the ‘one-versus-one’ binary, ‘one-versus-all’ binary, conventional multi-class, variability-regularized multi-class margin-based classifiers developed in Section 2.3. We observe that multi-class classifiers perform better than the binary margin-based classifiers. This is perhaps because multiple categories are available in our translation task. However, the multi-class classifiers require more advanced optimization procedures than the binary margin-based classifiers. The performance of variability-regularized multi-class classifiers is better than that of the conventional multi-class classifiers when ten additional feature vectors are generated for each source feature vector (a generation

¹<http://www.statmt.org/moses/>

²<http://www.microsofttranslator.com/>

³<http://translate.google.com>

factor of ten). Many margin-based classifiers have better performance than that of the Google and Microsoft MT systems. This is perhaps because our system is designed for domain-specific topics, whereas Google and Microsoft MT systems were designed to handle any kind of text.

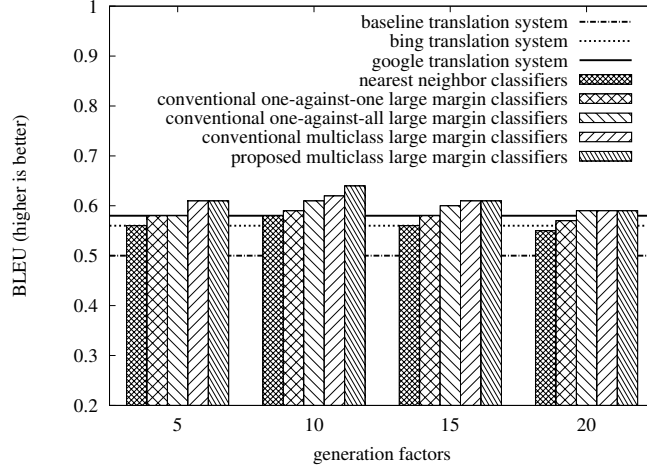
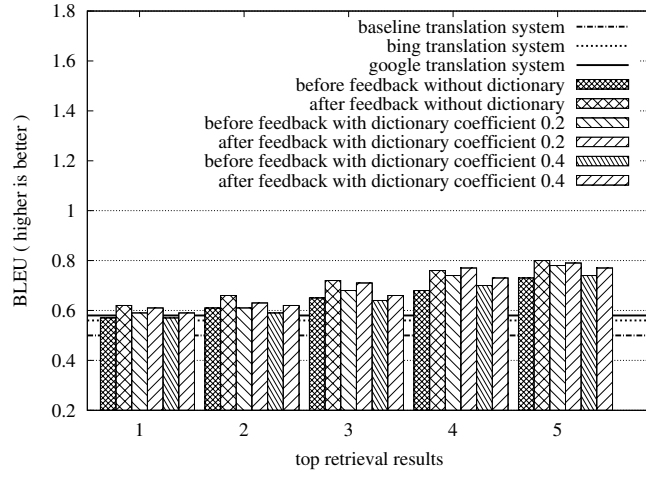


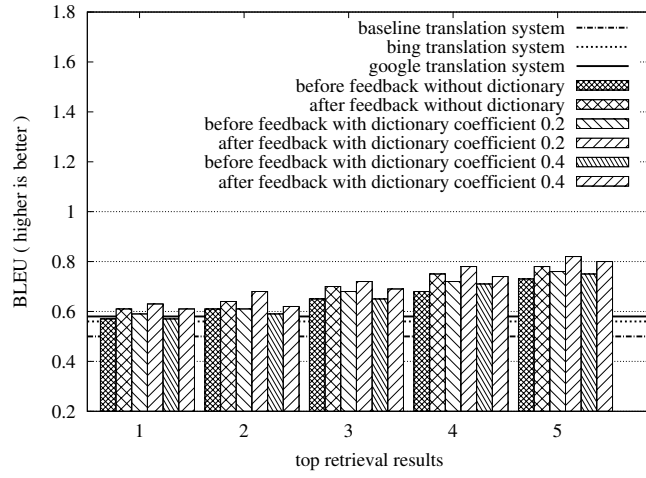
Figure 23: Performance of retrieval-based MT systems using different classification schemes is computed across different generation factors. The most effective generation factor for the proposed MT system is ten, which means ten additional feature vectors are generated for each source feature vector in the database.

Additional experiments were conducted to obtain more insights for the translation task. Specifically, we compute the BLEU score when the translation decision is obtained the top one, two, three, fourth, and fifth decisions of a retrieval model. In other words, our translation procedure is viewed as a retrieval task rather than as a classification task. The main difference between the text classification and retrieval tasks is that the former is concerned with the assignment of different categories to a sentence (document), while the latter task is concerned with the degree of similarity between two sentences. Figure 24 summarizes the results of using background information (WordNet) on the retrieval-based MT system.

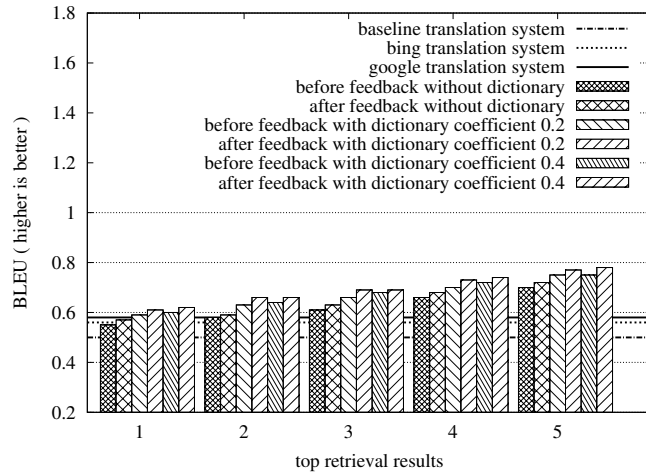
This experiment uses the language-model retrieval model developed in Section 3.2. First, the document generation procedure described in Section 4.2 is used to expand the coverage of our MT system. Then, background knowledge (WordNet) is



(a) Five variations of each original source document are generated.



(b) Ten variations of each original source document are generated.



(c) Fifteen variations of each source document are generated.

Figure 24: Comparison of BLEU scores of the four MT systems for different generation factors of five, ten, and fifteen.

incorporated during the document generation procedure. After that, language models are estimated using both the original source sentences and additional variations of the source sentences. The experiment uses three relevance levels: a relevance level of zero, one, and two. The language model parameters are optimized for NPCC performance measure, which is defined in Section 3.2. From the database, we identify 20 major categories and 120 sub-categories as illustrated in Table 18. Subjective judgments are specified to have the value of one for sentences belong to the same major categories, to have the value of two for sentences belong to the same sub-category, and zero otherwise.

Figure 24(a), (b), and (c) lists the BLEU metrics across the three different generation factors of five, ten, and fifteen. The experiments detailed in Figure 24(a), (b), and (c) use the same baseline systems as those in Figure 23. The performance on the cross-validation dataset can be used to identify the best generation coefficient to incorporate the electronic dictionary (WordNet). From the experimental results, we recommend using the generation factor of ten for this dataset. For experiments on a specific generation factor, the BLEU metrics are computed for retrieval lists of different length, e.g., retrieval lists of length one, two, three, four, and five. The calculation of the BLEU metrics are modified slightly in retrieval-based machine-translation systems because a list of possible translation sentences are available as the outputs. If the correct translation is present in the retrieved list, the BLEU metric is computed based on the correct target text. If the correct translation is not present in the list, the BLEU metric is computed based on the first choice of the retrieved translation. For all generation factors, the BLEU scores improve after the use of subjective judgements. Obviously, the BLEU performances increase if the retrieval list become longer. Using a retrieval-based approach, our MT system has the advantage of obtaining meaningful output sentences for every query sentence because those output sentences are retrieved directly from the database.

4.4 *Summary*

This dissertation explores the use of pattern recognition principles to improve an example-based MT system. In a few words, an EBMT performs translation by searching similar translation examples in the database and retrieving a target sentence corresponding to the most similar source example. A translation of the input sentence can be generated by combining relevant (retrieved) translation examples in an appropriate way. In this dissertation, pattern recognition techniques are used to obtain the associated translation sentences. The developed MT system is named a retrieval-based MT system. A main distinction of our retrieval-based MT system is the use of the sentential translation unit. The main advantage of this design choice is that the semantic or meaning of the source and target sentences is precisely defined. The main disadvantage of this design choice is that it provides limited coverage of long (sentential) translation units.

One effective way to improve the coverage of a retrieval-based MT system is through paraphrasing or rewriting a source text using different words in the same language. However, paraphrasing is a very labor-intensive and time-consuming task. Thus, pattern recognition techniques are used to improve the coverage of the MT system by generating sensible variations of source feature vectors. The document generation procedure utilizes a topic model and an electronic dictionary (WordNet) to identify the specific and common topics among different source sentences. Our experiments were carried out using a parallel corpus with a database related to hotel reservations in a travel domain. Various classification/retrieval schemes can be used in a retrieval-based MT system based on both original and generated documents. Our experimental results demonstrate the superiority of the proposed retrieval-based MT system over the baseline statistical MT system and over commercially developed systems in domain-specific translation tasks.

CHAPTER V

CONCLUSION AND FUTURE WORK

An ideal Machine Translation (MT) system should take into account of both empirical data and linguistic features. At present, how to efficiently and effectively construct an ideal MT system remains an open research problem mainly because an ideal MT system requires various forms of knowledge such as knowledge about the corresponding meanings of words in the two languages, knowledge about the syntactic constraints of each language, and semantic and pragmatic knowledge. These various forms of knowledge are necessary to resolve the ambiguities of natural languages that exist at various levels of syntactic structures: word, phrase, clause, sentence and discourse. Using a shorter translation unit such as a word-level translation unit may provide greater coverage in the source language and higher adaptability in the target language. However, combining shorter translation units in the target language may change the meaning (semantic) of the original text unless advanced compositional techniques are used in the MT system. On the other hand, longer translation units such as a sentential translation unit may ensure meaningful translations in the target language. However, coverage of the longer translation unit is limited because exact matches of longer translation units are rarely encountered in practice.

Recent excitement in MT research has been inspired by the revival of data-driven approaches, such as example-based and statistical-based approaches (Chapter 1). Although the differences between example-based and statistical approaches are constantly under debate, these two approaches emphasize inherently different aspects of MT problems. In an example-based approach, a large database of translation

examples is collected and kept in the MT system. Translating an input query text requires the example-based MT approach to compare the input text to the similar texts in the database. A translation of the input sentence can be generated by combining relevant (retrieved) translation examples in a word-to-word (or phrase-to-phrase) fashion. The main distinction of the statistical-based approach is the assignment of statistical parameters to all translation features including the translation model and language models, as well as other data-driven features. Although statistical-based approaches were originally developed to operate as word-to-word translation units, many statistical-based MT systems can be operate based on linguistically-inspired features such as phrase (or clause) translation units. Nevertheless, these data-driven (empirical) approaches are relatively knowledge poor with respect to linguistic analysis. We have developed a variation of an example-based MT system that takes into account both empirical data and linguistic features.

This dissertation has developed statistical pattern classifiers to automatically organize the empirical data into different semantic categories. Manual assignment of semantic categories is tedious and error-prone. Chapter 2 demonstrates how several high performance pattern classifiers are developed by addressing the generalization capability of a classifier. The generalization capability of a classifier is concerned with the classifier’s ability to perform well on unseen test data. Traditionally, the generalization capability of a classifier design is indirectly addressed using the Occams Razor principle, which favors “simpler” classifiers (a lower number of model parameters) for a given training dataset. We argue that fewer model parameters may not result in the best classifier design. Instead, the objective should be to minimize empirical error. First, we investigate a discriminative model selection technique for mixture-based classifiers. Then, we develop a novel variability regularization procedure for margin-based classifiers. Our work in this chapter provides ways to directly improve the performance of a pattern classifier using the empirical (training) data.

Incorporating human preferences (judgments) into MT systems is important. For example, when translating a medical document, one may assign higher cost for confusing a fatal disease with a mild disease than confusing a mild with a fatal disease because the former is more likely to cause the loss of life. However, the incorporation of subjective judgments has not been widely studied in MT systems. We have developed a non-uniform learning procedure to incorporate subjective judgments and preferences into a statistical-based classifier (Chapter 3). The subjective judgments and preferences can be represented in terms of a cost matrix or in terms of relevant judgments. The main distinction of the developed framework is that the resulting statistical-based classifier can be directly optimized based on the cost-matrix or the relevant judgments. This strategy provides more direct connections among the statistical models, the subjective preferences, and the ultimate performance measure. We have verified the developed statistical framework in natural language processing. In addition, the framework is general enough to be applied in other classification tasks, such as handwritten digit recognitions and biological applications, etc.

The design of our retrieval-based MT system is addressed in detail in Chapter 4. Two distinctive characteristics of the developed MT system are the use of a sentence-level translation unit and the use an electronic dictionary to improve the coverage of the source database. The use of a sentence-level translation unit reduces the ambiguity of the translation unit in both the source and target sentences. The main drawback of this approach is the limited coverage of the sentence-level translation unit. We address this issue by using background knowledge as found in an electronic dictionary to enrich existing translation examples in the database. The developed classifier design techniques in Chapter 2 and the non-uniform learning procedure in Chapter 3 are used to obtain matching examples in the database. Our experiments show promising results of the developed retrieval-based MT system in comparison with conventional statistical-based MT system.

5.1 *Summaries and Contributions*

This dissertation uses MT systems as the main application to demonstrate our contribution in pattern recognition techniques. Indeed, the developed pattern recognition techniques can be applied in various application domains. The contributions of this dissertation can be summarized as follows:

1. We presented novel model selection and model regularization algorithms for classifier designs from a limited number of training data.
 - We developed a variability regularization strategy for hyperplane classifiers.
 - We provided an algorithm to enable the use of discriminative model selection in mixture-based classifiers.
2. We enabled the incorporation of human preference and judgment into the statistical models through the use of error-cost learning procedure.
 - We formulated a novel multi-level irrelevance performance measure to incorporate human judgments into an information retrieval application.
 - We provided a systematic way to represent human preferences in a cost matrix and a way to use that subjective information to actively control inter-class confusions.
3. We quantitatively validated our novel pattern recognition techniques in *matching procedures* in a retrieval-based MT system. Note that our objective in this language translation system is illustrative of our novel statistical pattern recognition techniques.

The following is a list of refereed publications produced by the work in this dissertation from about 2008.

- **Mansjur, D.S.**, Wada T. and Juang, B.H., “ Variability Regularization in Large-Margin Classification,” in *the 33rd International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1956-1959, Prague, Czech Republic, May 2011.
- Fu, Q., **Mansjur, D.S.**, and Juang, B.H., ”Empirical System Learning for Statistical Pattern Recognition With Non-Uniform Error Criteria,” *IEEE Transactions on Signal Processing*, vol 58, pp. 4621-4633, September 2010.
- **Mansjur, D.S.**, Wada T. and Juang, B.H., “ Using Kernel Density Classifier with Topic Model and Cost Minimization Procedure for Automatic Text Categorization”, in *the 10th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1086-1090, Barcelona, Spain, July 2009.
- **Mansjur, D.S.**, and Juang, B.H., “ Improving Kernel Density Classifier Using Corrective Bandwidth Learning with Smooth Error Loss Function”, in *the 7th International Conference on Machine Learning and Applications (ICMLA)*, pp. 161-167, San Diego, CA, USA, December 2008.
- **Mansjur, D.S.**, and Juang, B.H., “ Incremental Learning of Mixture Models for Simultaneous Estimation of Class Distribution and Inter-Class Decision Boundaries”, in *the 19th International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA, December 2008.
- **Mansjur, D.S.**, Fu, Q. and Juang, B.H., “ Utilizing Non-Uniform Cost Learning for Active Control of Inter-Class Confusion”, in *the 19th International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA, December 2008.
- Fu, Q., **Mansjur, D.S.**, and Juang, B.H., “ Non-Uniform error criteria for automatic pattern and speech recognition”, in *the 33rd International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1853-1856, Las

Vegas, NV, USA, April 2008.

- **Mansjur, D.S.**, and Juang, B.H., “Multi-level Relevance Performance Measure for Language Modeling and Document Retrieval”, Under preparation.

5.2 Avenues of Future Research

More work can be done to refine and extend the developed retrieval-based MT system.

Several promising research directions based on this work are as follows:

- Richer and more robust representations of examples/documents are crucial for a high-performance retrieval-based MT system. Instead of using a straightforward word occurrence representation, a representation using a complex network of words, namely a graphical model, can be used to represent the statistical dependencies among a set of words. The main drawback of using this richer representation is the computational cost.
- Many classifier design principles can benefit from the developed variability regularization strategy. Currently, the developed regularization strategy is applied only to margin-based classifiers but margin error is in reality only an approximation of the error count in the empirical error rate. Thus, a promising research direction is to extend the developed strategy to regularize empirical error rates directly.
- Extension of the error-cost learning to different kinds of performance measures is definitely worthwhile. This dissertation has provided two possible applications of the error-cost learning to incorporate subjective preference and judgment. In reality, many other applications of error-cost learning are possible.

In conclusion, we feel that MT researchers have not completely exploited the strength of existing pattern recognition techniques. Similarly, pattern recognition researchers seem to neglect the challenges in MT problems. Increasing the interaction

between these two fields can definitely result in many fruitful results in both fields. We hope that the work in this dissertation contributes to both the pattern recognition and the machine translation fields.

REFERENCES

- [1] AKAMINE, S., FURUSE, O., and IIDA, H., “Integration of example-based transfer and rule-based generation,” in *Proceedings of the fourth conference on Applied natural language processing*, pp. 196–197, Association for Computational Linguistics, 1994.
- [2] ARAMAKI, E., IMAI, T., MIYO, K., and OHE, K., “Support vector machine based orthographic disambiguation,” in *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 21–30, Citeseer, 2007.
- [3] ASUNCION, A. and NEWMAN, D., “UCI machine learning repository,” 2007.
- [4] BANERJEE, S., “Improving text classification accuracy using topic modeling over an additional corpus,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 867–868, ACM, 2008.
- [5] BIEM, A., HA, J.-Y., and SUBRAHMONIA, J., “A Bayesian model selection criterion for HMM topology optimization ,” *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I–989–I–992 vol.1, 2002.
- [6] BLEI, D., NG, A., and JORDAN, M., “Latent Dirichlet Allocation,” in *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, MIT Press Cambridge, MA, USA, 2003.
- [7] BLUM, J. R., “Multidimensional stochastic approximation methods,” *The Annals of Mathematical Statistics*, vol. 25, pp. 737–744, April 1954.
- [8] BOITET, C., “A research perspective on how to democratize machine translation and translation aids aiming at high quality final output,” in *Proc. MT Summit VII, Singapore*, pp. 13–17, Citeseer, 1999.
- [9] BOLLEGALA, D., MATSUO, Y., and ISHIZUKA, M., “Measuring semantic similarity between words using web search engines,” in *Proceedings of WWW*, vol. 7, pp. 757–786, 2007.
- [10] BOSER, B., GUYON, I., and VAPNIK, V., “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

- [11] BROWN, P., DELLA PIETRA, S., DELLA PIETRA, V., JELINEK, F., LAFERTY, J., MERCER, R., and ROOSSIN, P., “A statistical approach to machine translation,” *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.
- [12] BROWN, R., “Automated generalization of translation examples,” in *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pp. 125–131, Association for Computational Linguistics, 2000.
- [13] BURCH, C. C., *Paraphrasing and Translation*. PhD thesis, University of Edinburgh, 2007.
- [14] BURGESS, C., “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [15] CALLISON-BURCH, C., OSBORNE, M., and KOEHN, P., “Re-evaluating the role of BLEU in machine translation research,” in *11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 249–256, Citeseer, 2006.
- [16] CARL, M., “A model of competence for corpus-based machine translation,” *Proceedings of the 18th conference on Computational linguistics*, vol. 2, pp. 997–1001, 2000.
- [17] CARL, M., “A system-theoretical view of EBMT,” *Machine Translation*, vol. 19, no. 3, pp. 229–249, 2005.
- [18] CARL, M., PEASE, C., IOMDIN, L., and STREITER, O., “Towards a dynamic linkage of example-based and rule-based machine translation,” *Machine Translation*, vol. 15, no. 3, pp. 223–257, 2000.
- [19] CARL, M., PEASE, C., and STREITER, O., “Linking Example-Based and Rule-Based Machine Translation,” in *The seventh Machine Translation Summit*, Citeseer, 1999.
- [20] CARL, M., “Inducing translation grammars from bracketed,” in *Recent advances in example-based machine translation*, pp. 339–361, The Kluwer Academic Publishers, 2003.
- [21] CHIEN, J.-T. and WU, M.-S., “Minimum rank error language modeling,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, pp. 267–76, 2009.
- [22] CHOU, W. and JUANG, B.-H., “Adaptive discriminative learning in pattern recognition,” tech. rep., AT&T Bell Labs, Murray Hill, NJ, 1991.
- [23] COHEN, W. and SINGER, Y., “Context-sensitive learning methods for text categorization,” *ACM Transactions on Information Systems (TOIS)*, vol. 17, no. 2, pp. 141–173, 1999.

- [24] COLLINS, B., *Example-based Machine Translation: an adaptation-guided retrieval approach*. PhD thesis, University of Dublin, Trinity College. Department of Computer Science, 1999.
- [25] CORTES, C., *Prediction of Generalization Ability in Learning Machines*. PhD thesis, University of Rochester, Rochester, New York, 1993.
- [26] COVER, T. and THOMAS, J., *Elements of information theory*. Wiley New York, 1991.
- [27] DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., L, T. K., and HARSHMAN, R., “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.
- [28] DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B., “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [29] DOMINGOS, P., “Metacost: A general method for making classifiers cost-sensitive,” in *Knowledge Discovery and Data Mining*, pp. 155–164, 1999.
- [30] DOOB, J. L., *Stochastic Process*. New York, NY: Wiley, 1953.
- [31] DUDA, O. R., HART, P. E., and STORK, D. G., *Pattern Classification*. John Wiley and Sons, NY, 2001.
- [32] ELKAN, C., “The foundations of cost-sensitive learning,” in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 973–978, 2001.
- [33] FELLBAUM, C., *WordNet: An electronic lexical database*. The MIT press, 1998.
- [34] FRANC, V., *Optimization Algorithms for Kernel Methods*. PhD thesis, Czech Technical University in Prague, 2005.
- [35] FRANK, A. and ASUNCION, A., “UCI machine learning repository,” in *University of California, Irvine, School of Information and Computer Sciences*, 2010.
- [36] FU, Q., *A generalization of the minimum classification error (mce) training method for speech recognition and detection*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2008.
- [37] FU, Q., MANSJUR, D. S., and JUANG, B. H., “Non-uniform error criteria for automatic pattern and speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1853–1856, April 2008.

- [38] FU, Q., MANSJUR, D. S., and JUANG, B. H., “Empirical system learning for statistical pattern recognition with non-uniform error criteria,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 4621 – 4633, September 2010.
- [39] FUKUNAGA, K., *Introduction to Statistical Pattern Recognition*. San Diego, California: Academic Press, 1990.
- [40] FURUSE, O. and IIDA, H., “Cooperation between transfer and analysis in example-based framework,” *Proceedings of the 14th conference on Computational linguistics*, vol. 2, pp. 645–651, 1992.
- [41] GABRILOVICH, E. and MARKOVITCH, S., “Computing semantic relatedness using wikipedia-based explicit semantic analysis,” in *Proceedings of the 20th international joint conference on Artificial intelligence*, pp. 1606–1611, 2007.
- [42] GANGADHARAI, R., BROWN, R., and CARBONELL, J., “Spectral clustering for example based machine translation,” in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pp. 41–44, Association for Computational Linguistics, 2006.
- [43] GEMAN, S., BIENENSTOCK, E., and DOURSAT, R., “Neural networks and the bias/variance dilemma,” *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.
- [44] GOLLER, C., LÖNING, J., WILL, T., and WOLFF, W., “Automatic document classification: A thorough evaluation of various methods,” in *Internationales Symposium für Informationswissenschaft (ISI 2000)*, 2000.
- [45] GRAY, A. G. and MOORE, A. W., “‘n-body’ problems in statistical learning,” in *NIPS*, pp. 521–527, 2000.
- [46] GRUND, B. and POLZEHL, J., “Bias corrected bootstrap bandwidth selection,” tech. rep., University of Minnesota, 1996.
- [47] HÄRDLE, W., *Smoothing Techniques: with Implementation in S*. Springer Verlag, 1991.
- [48] HAYES, P. J., ANDERSON, P. M., NIRENBURG, I. B., and SCHMANDT, L. M., “TCS: A shell for content-based text categorization,” in *IEEE Conference on Artificial Intelligence Applications*, 1990.
- [49] HAYES, P. J. and WEINSTEIN, S. P., “CONSTRUE/TIS: a system for content-based indexing of a database of news stories,” in *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.
- [50] HOANG, H., BIRCH, A., CALLISON-BURCH, C., ZENS, R., AACHEN, R., CONSTANTIN, A., FEDERICO, M., BERTOLDI, N., DYER, C., COWAN, B., SHEN, W., MORAN, C., and BOJAR, O., “Moses: statistical machine translation system.” <http://www.statmt.org/moses/>, 2010.

- [51] HOFMANN, T., “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, ACM Press New York, NY, USA, 1999.
- [52] HOTH, A., STAAB, S., and GERD, S., “Ontologies improve text document clustering,” in *Proceedings of the Third IEEE International Conference on Data Mining, ICDM '03*, (Washington, DC, USA), pp. 541–, IEEE Computer Society, 2003.
- [53] HSU, C. and LIN, C., “A comparison of methods for multiclass support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [54] HUTCHINS, J., “Towards a definition of example-based machine translation,” in *Proceedings of the 2nd Workshop on Example-Based Machine Translation at MT Summit X*, pp. 63–70, Citeseer, 2005.
- [55] HUTCHINS, J. and SOMERS, H. L., *An introduction to machine translation*. Academic Press, 1992.
- [56] JÄRVELIN, K. and KEKÄLÄINEN, J., “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, p. 446, 2002.
- [57] JELINEK, F. and MERCER, R., “Interpolated estimation of markov source parameters from sparse data,” in *Workshop on Pattern Recognition in Practice*, pp. 381–397, North Holland, 1980.
- [58] JIANG, H. and LI, X., “Incorporating training errors for large margin HMMs under semi-definite programming framework,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, 2007.
- [59] JUANG, B.-H., CHOU, W., and LEE, C.-H., “Minimum classification error rate methods for speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.
- [60] JUANG, B.-H., HOU, W., and LEE, C.-H., “Minimum classification error rate methods for speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257–265, May 1997.
- [61] JUANG, B.-H. and KATAGIRI, S., “Discriminative learning for minimum error classification,” *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.
- [62] KAJI, H., KIDA, Y., and MORIMOTO, Y., “Learning translation templates from bilingual text,” in *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pp. 672–678, Association for Computational Linguistics, 1992.

- [63] KIBRIYA, A., FRANK, E., PFAHRINGER, B., and HOLMES, G., “Multinomial naive bayes for text categorization revisited,” in *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, vol. 3339, pp. 488–499, Springer, 2004.
- [64] KOEHN, P., HOANG, H., BIRCH, A., CALLISON-BURCH, C., FEDERICO, M., BERTOLDI, N., COWAN, B., SHEN, W., MORAN, C., ZENS, R., and OTHERS, “Moses: Open source toolkit for statistical machine translation,” *Annual Meeting-Association for Computational Linguistics*, vol. 45, no. 2, p. 2, 2007.
- [65] KOEHN, P., OCH, F., and MARCU, D., “Statistical phrase-based translation,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 48–54, Association for Computational Linguistics Morristown, NJ, USA, 2003.
- [66] LANG, K., “NewsWeeder: Learning to filter netnews,” in *12th International Conference on Machine Learning (ICML95)*, pp. 331–339, 1995.
- [67] LI, J. Q. and BARRON, A. R., “Mixture density estimation,” in *Neural Information Processing Systems*, pp. 279–285, 1999.
- [68] LOFTSGAARDEN, D. O. and QUESENBERY, C. P., “A nonparametric estimate of a multivariate density function,” *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 1049–1051, 1965.
- [69] MANSJUR, D. S., FU, Q., and JUANG, B. H., “Utilizing non-uniform cost learning for active control of inter-class confusion,” in *Proceedings of the 19th International Conference on Pattern Recognition, Tampa, FL, USA*, Dec. 2008.
- [70] MANSJUR, D. S. and JUANG, B. H., “Incremental learning of mixture models for simultaneous estimation of class distribution and inter-class decision boundaries,” in *Proceedings of the 19th International Conference on Pattern Recognition, Tampa, FL, USA*, Dec. 2008.
- [71] MCCALLUM, A. and NIGAM, K., “A comparison of event models for naive bayes text classification,” in *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, 1998.
- [72] MCDERMOTT, E. and KATAGIRI, S., “A new formalization of minimum classification error using a parzen estimate of classification chance,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 713–716, 2003.
- [73] MEI, Q. and ZHAI, C., “Discovering evolutionary theme patterns from text: an exploration of temporal text mining,” in *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 198–207, ACM, 2005.

- [74] MIHALCEA, R., CORLEY, C., and STRAPPARAVA, C., “Corpus-based and knowledge-based measures of text semantic similarity,” in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, p. 775, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [75] MILNE, D. and WITTEN, I., “Learning to link with wikipedia,” in *Proceeding of the 17th ACM conference on Information and knowledge management*, pp. 509–518, ACM, 2008.
- [76] MONTGOMERY, D. and PECK, E., *Introduction to linear regression analysis*. John Wiley, New York, 1992.
- [77] NAGAO, M., “A framework of a mechanical translation between japanese and english by analogy principle,” in *Proc. of the international NATO symposium on Artificial and human intelligence*, (New York, NY, USA), pp. 173–180, Elsevier North-Holland, Inc., 1984.
- [78] NIRENBURG, S., DOMASHNEV, C., and GRANNES, D., “Two approaches to matching in example-based machine translation,” in *Proc. of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93)*, pp. 47–57, Citeseer, 1993.
- [79] OCH, F., TILLMANN, C., and NEY, H., “Improved alignment models for statistical machine translation,” in *In Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 20–28, 1999.
- [80] ORTIZ-MARTÍNEZ, D., GARCÍA-VAREA, I., and CASACUBERTA, F., “Thot: a toolkit to train phrase-based models for statistical machine translation.” <http://thot.sourceforge.net/>, 2010.
- [81] PAPINENI, K., ROUKOS, S., WARD, T., and ZHU, W., “BLEU: a Method for Automatic Evaluation of Machine Translation,” *40th Annual meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.
- [82] PEDERSEN, T. and PATWARDHAN, S., “Wordnet::similarity - measuring the relatedness of concepts,” in *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, (San Jose, CA), pp. 1024–1025, July 2004.
- [83] PENG, F., SCHUURMANS, D., and WANG, S., “Language and task independent text categorization with simple language models,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 110–117, Association for Computational Linguistics Morristown, NJ, USA, 2003.
- [84] PONTE, J. and CROFT, W., “A language modeling approach to information retrieval,” in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 275–281, ACM New York, NY, USA, 1998.

- [85] POVEY, D. and WOODLAND, P. C., “Minimum phone error and i-smoothing for improved discriminative training,” in *International Conference on Acoustics, Speech, and Signal Processing*, (Orlando, FL), pp. 105–108, May 2002.
- [86] ROBBINS, H. and MONRO, S., “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, April 1951.
- [87] SALTON, G. and MCGILL, M., *Introduction to modern information retrieval*. McGraw-Hill New York, 1983.
- [88] SATO, S., “CTM: an example-based translation aid system,” *Proceedings of the 14th conference on Computational linguistics*, vol. 4, pp. 1259–1263, 1992.
- [89] SATO, S. and NAGAO, M., “Toward memory-based translation,” *Proceedings of the 13th conference on Computational linguistics*, vol. 3, pp. 247–252, 1990.
- [90] SCHÖLKOPF, B. and SMOLA, A., *Learning with kernels*. The MIT Press, 2002.
- [91] SEBASTIANI, F., “Machine learning in automated text categorization,” *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.
- [92] SHA, F. and SAUL, L., “Large margin hidden Markov models for automatic speech recognition,” *Advances in Neural Information Processing Systems*, vol. 19, p. 1249, 2007.
- [93] SILVERMAN, B., *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability, London-New York: Chapman and Hall, 1986.
- [94] SMITH, J. and CLARK, S., “EBMT for SMT: A New EBMT-SMT Hybrid,” in *3rd International Workshop on Example-Based Machine Translation*, p. 3, Citeseer, 2009.
- [95] SOMERS, H., “Review article: Example-based machine translation,” *Machine Translation*, vol. 14, no. 2, pp. 113–157, 1999.
- [96] SOMERS, H., “An overview of EBMT,” *Recent advances in example-based machine translation*, pp. 3–57, 2003.
- [97] SOUCY, P. and MINEAU, G., “Beyond TFIDF weighting for text categorization in the vector space model,” in *Proceedings of the Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 1130–1135, 2005.
- [98] STOLCKE, A., “SriLm: the sri language modeling toolkit.” <http://www.speech.sri.com/projects/sriLm/>, 2010.
- [99] STURM, J. F., “Using SeDuMi 1.02, a MATLAB Toolbox for Optimization Over Symmetric Cones,” in *Optimization Methods and Software 11-12*, pp. 625–653, 1999.

- [100] SUMITA, E., AKIBA, Y., DOI, T., FINCH, A., IMAMURA, K., OKUMA, H., PAUL, M., SHIMOHATA, M., and WATANABE, T., “EBMT, SMT, Hybrid and More: ATR spoken language translation system,” in *Proc. of the International Workshop on Spoken Language Translation*, 2004.
- [101] SUMITA, E., AKIBA, Y., DOI, T., FINCH, A., IMAMURA, K., PAUL, M., SHIMOHATA, M., and WATANABE, T., “A corpus-centered approach to spoken language translation,” *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, vol. 2, pp. 171–174, 2003.
- [102] SUMITA, E. and IIDA, H., “Experiments and prospects of example-based machine translation,” in *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pp. 185–192, Association for Computational Linguistics, 1991.
- [103] SUMITA, E., IIDA, H., and KOHYAMA, H., “Translating with examples: A new approach to machine translation,” in *The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, pp. 203–212, 1990.
- [104] THEODORIDIS, S. and KOUTROUMBAS, K., *Pattern Recognition*. Academic Press, 4 ed., 2008.
- [105] TURCATO, D. and POPOWICH, F., “What is example-based machine translation?,” in *Recent Advances in Example-Based Machine Translation*, 2003.
- [106] VAPNIK, V., *The nature of statistical learning theory*. Springer Verlag, 1995.
- [107] VAPNIK, V., *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.
- [108] VLASSIS, N. A. and LIKAS, A., “A greedy em algorithm for gaussian mixture learning,” *Neural Processing Letters*, vol. 15, no. 1, pp. 77–87, 2002.
- [109] WATANABE, T., *Example-Based Statistical Machine Translation*. PhD thesis, Kyoto University, 2004.
- [110] WAY, A. and GOUGH, N., “Comparing example-based and statistical machine translation,” *Natural Language Engineering*, vol. 11, pp. 295–309, September 2005.
- [111] WESTON, J. A. E., *Extensions to the Support Vector Method*. PhD thesis, University of London, 1999.
- [112] WU, D., “MT model space: statistical versus compositional versus example-based machine translation,” *Machine Translation*, vol. 19, no. 3, pp. 213–227, 2005.

- [113] YAMADA, K. and KNIGHT, K., “A syntax-based statistical translation model,” in *Proceedings of the Conference of the Association for Computational Linguistics*, 2001.
- [114] YANG, Y. and LIU, X., “A re-examination of text categorization methods,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42–49, August 1999.
- [115] YANG, Y. and CHUTE, C. G., “An example-based mapping method for text categorization and retrieval,” *ACM Trans. Inf. Syst.*, vol. 12, pp. 252–277, July 1994.
- [116] YANG, Y. and PEDERSEN, J. O., “A comparative study on feature selection in text categorization,” in *14th International Conference on Machine Learning (ICML)*, (FISHER, D. H., ed.), (Nashville, US), pp. 412–420, Morgan Kaufmann Publishers, San Francisco, US, 1997.
- [117] ZHAI, C. and LAFFERTY, J., “A study of smoothing methods for language models applied to information retrieval,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, p. 214, 2004.
- [118] ZHAI, C., VELIVELLI, A., and YU, B., “A cross-collection mixture model for comparative text mining,” in *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 743–748, ACM, 2004.

VITA

Dwi Sianto Mansjur was born in Sumatera Utara, Indonesia, in September 1979. He is currently a Ph.D. candidate in School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, where he earned her B.S. and M.S. degree in 2002 and 2006, respectively. He worked as a Graduate Research Assistant from 2004 to 2010. During summer of 2009, he was employed by HP Exstream document automation software. He will receive his PhD Degree in Electrical Engineering from Georgia Institute of Technology in 2011. His research interests include text categorization, information retrieval, machine translation, and discriminative or margin-based learning procedure for pattern recognition tasks.