Neurocomputing 80 (2012) 10–18

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Predictive active set selection methods for Gaussian processes

Ricardo Henao [a],*, Ole Winther [b]

[a] DTU Informatics, Technical University of Denmark, Denmark
[b] Bioinformatics Centre, University of Copenhagen, Denmark

ARTICLE INFO

Available online 7 November 2011

Keywords:
Gaussian process classification
Active set selection
Predictive distribution
Expectation propagation

ABSTRACT

We propose an active set selection framework for Gaussian process classification for cases when the dataset is large enough to render its inference prohibitive. Our scheme consists of a two step alternating procedure of active set update rules and hyperparameter optimization based upon marginal likelihood maximization. The active set update rules rely on the ability of the predictive distributions of a Gaussian process classifier to estimate the relative contribution of a data point when being either included or removed from the model. This means that we can use it to include points with potentially high impact to the classifier decision process while removing those that are less relevant. We introduce two active set rules based on different criteria, the first one prefers a model with interpretable active set parameters whereas the second puts computational complexity first, thus a model with active set parameters that directly control its complexity. We also provide both theoretical and empirical support for our active set selection strategy being a good approximation of a full Gaussian process classifier. Our extensive experiments show that our approach can compete with state-of-the-art classification techniques with reasonable time complexity. Source code publicly available at http://cogsys.imm.dtu.dk/passgp.

1. Introduction

Classification with Gaussian process (GP) priors has many attractive features, for instance it is non-parametric, exceptionally flexible through covariance function designs, provides fully probabilistic outputs and Bayesian model comparison as principled framework for automatic hyperparameter elicitation and variable selection. However, such a set of features comes in with a great disadvantage since the computational cost of performing inference scales cubically with the size, N, of the training set. In addition, the memory requirements scale quadratically also with N. This means that applicability of Gaussian process classifiers (GPCs) is sadly limited to problems with dataset sizes in the lower ten thousands. The poor scaling of specially non-linear classification methods has inspired a considerable amount of research effort focused on sparse approximations [1–7]. See particularly [1,2] for a detailed overview of sparse approximations in GPCs. These methods attempt in general to decrease the computational cost of inference in one degree w.r.t. N, i.e. $\mathcal{O}(NM^2)$, where $M < N$ and M is the size of a working set consisting on a subset of the training data or a set of auxiliary unobserved variables. Both ways of defining the working set basically target the same objective of getting as close as possible to the classifier that uses the information of the entire training set, however they approach it from different angles. Using a subset from the entire data pool amounts to keep those data points that better contribute to the classification task and discard the remaining ones through some suitable data selection/ranking procedure [6–10]. Alternatively, building an auxiliary set tries to directly reduce the difference in distribution between the classifier using N points and the one using only M, by estimating the location of an auxiliary set in the input space, usually called pseudo-input set [1,4,11]. The latter approach is evidently more principled, however the number of parameters to be learnt grows with the number and size of the auxiliary set, making it unfeasible for datasets in the upper ten thousands and sensitive to overfitting due to the number of free parameters in the model. From a fully Bayesian perspective, in [12] the authors propose an efficient MCMC based inference that is made possible by using a sparse and approximate basis function expansion over the training dataset. The main computational burden is therefore the same as other sparse kernel methods possibly with a larger pre-factor due to sampling.

Having in mind that our main goal is to obtain the best classification performance with the least computational cost possible, we do not attempt to estimate auxiliary sets but rather

* Corresponding author.
E-mail addresses: rhenao@binf.ku.dk (R. Henao), owi@imm.dtu.ku.dk (O. Winther).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.09.017

to select a subset of the training data. The framework presented here, Predictive Active Set Selection (PASS-GP) uses the predictive distribution of a GPC in order to quantify the relative importance of each data point and then use it to iteratively update an active set. Recently, Kapoor et al. [6] proposed a similar criterion in the context of active learning with Gaussian processes. We use the term active set because it is ultimately the one used to estimate the predictive distribution that produces the classification rule and active set updating scheme. In a nutshell, our framework consists of alternating between active set updates and hyperparameter optimization based upon the marginal likelihood of the active set. We provide two active set update schemes that target different practical scenarios. The first simply called PASS-GP builds the active set by including/removing points with small/large predictive probability until no more or too few data points are included in the active set. This means that the size of the active set is not known in advance so as the expected computational complexity. The second scheme is aware that in some applications is very important to keep the computational complexity and/or memory requirements on a budget, thus being able to specify the size of the active set beforehand is essential. In fixed PASS-GP (fPASS-GP) we keep the size of the active set constant by including and removing the same amount of data points in each update to achieve the desired behavior.

The remainder of the paper presents in Section 2 a concise description of expectation propagation based inference for GPCs. Section 3 continues with our proposed framework for active set selection, followed by some theoretical insights based upon a 'representer theorem' for the predictive mean of a GP classifier in Section 4. Marginal likelihood approximations to the full GP classifier are introduced in Section 5. Finally, experimental results and discussion appear in Sections 6 and 7, respectively.

2. Gaussian processes for classification

Given a set of input random variables $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]^\top$, a Gaussian process is defined as a joint Gaussian distribution over function values at the input points $\mathbf{f}=[f_1,\ldots,f_N]^\top$ with mean vector $\mathbf{m}$ (taken to be zero in the following) and covariance matrix $\mathbf{K}$ with elements $K_{ij}=k(\mathbf{x}_i,\mathbf{x}_j)$ and hyperparameters $\boldsymbol{\theta}$. For classification, assuming independently observed binary ± 1 labels $\mathbf{y}=[y_1,\ldots,y_N]^\top$ and a probit (cumulative Gaussian) likelihood function $t(y_n|f_n)=\Phi(f_n y_n)$, we end up with an intractable posterior

$$p(\mathbf{f}|\mathbf{X},\mathbf{y})=Z^{-1}p(\mathbf{f}|\mathbf{X})\prod_{n=1}^{N}t(y_n|f_n),$$

where the normalizing constant $Z=p(\mathbf{y}|\mathbf{X})$ is the marginal likelihood. If we want to perform inference we must resort to approximations. Here we use Expectation Propagation (EP) because it is currently the most accurate deterministic approximation, see e.g. [2,13]. In EP, the likelihood function is locally approximated by an un-normalized Gaussian distribution to obtain

$$\begin{aligned} q(\mathbf{f}|\mathbf{X},\mathbf{y}) = {} & Z_{\mathrm{EP}}^{-1}p(\mathbf{f}|\mathbf{X})\prod_{n=1}^{N}z_n^{-1}\tilde{t}(y_n|f_n) \\ & =Z_{\mathrm{EP}}^{-1}p(\mathbf{f}|\mathbf{X})\mathcal{N}(\mathbf{f}|\tilde{\mathbf{m}},\tilde{\mathbf{C}}) \\ & =\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{c}), \end{aligned} \tag{1}$$

where $q(\mathbf{f}|\mathbf{X},\mathbf{y})\approx p(\mathbf{f}|\mathbf{X},\mathbf{y})$, the z_n are the normalization coefficients, $\tilde{t}(y_n|f_n)$ and $\mathcal{N}(\mathbf{f}|\tilde{\mathbf{m}},\tilde{\mathbf{C}})$ conform the site Gaussian approximations to $t(y_n|f_n)$. In order to obtain $q(\mathbf{f}|\mathbf{X},\mathbf{y})$, one starts from $q(\mathbf{f}|\mathbf{X},\mathbf{y})=p(\mathbf{f}|\mathbf{X},\mathbf{y})$ and update the individual $\tilde{t}_n$ site approximations sequentially. For this purpose, we delete the site approximation $\tilde{t}_n$ from the current posterior leading to the so-called cavity distribution

$$q_{\backslash n}(\mathbf{f}|\mathbf{X},\mathbf{y}_{\backslash n})=p(\mathbf{f}|\mathbf{X})\prod_{i\neq n}z_i^{-1}\tilde{t}(y_i|f_i),$$

from which we can obtain a cavity predictive distribution

$$q_{\backslash n}(y_n|\mathbf{X},\mathbf{y}_{\backslash n})= \int t(y_n|f_n)q_{\backslash n}(\mathbf{f}|\mathbf{X},\mathbf{y}_{\backslash n})\,d\mathbf{f}=\Phi\left(\frac{y_n m_{\backslash n}}{\sqrt{1+v_{\backslash n}}}\right), \tag{2}$$

where $m_{\backslash n}=v_{\backslash n}(C_{nn}^{-1}m_n-\tilde{C}_{nn}^{-1}\tilde{m}_n)$ and $v_{\backslash n}=(C_{nn}^{-1}-\tilde{C}_{nn}^{-1})^{-1}$. We then combine the cavity distribution with the exact likelihood $t(y_n|f_n)$, to obtain the so-called tilted distribution $q_n(\mathbf{f}|\mathbf{X},\mathbf{y})=z_n^{-1}t(y_n|f_n)$ $q_{\backslash n}(\mathbf{f}|\mathbf{X},\mathbf{y}_{\backslash n})$. Since we need to choose the parameters of the site approximations we must minimize some divergence measure. It is well known that when $q(\mathbf{f}|\mathbf{X},\mathbf{y})$ is Gaussian, minimizing $\mathrm{KL}(p(\mathbf{f})||q(\mathbf{f}))$ is equivalent to moment matching between those two distributions including zero-th moments for the normalizing constants. The EP algorithm iterates by updating each site approximation in turn and makes several passes over the training data.

With the Gaussian approximation to the posterior distribution in Eq. (1), it is possible to calculate the predictive distribution of a new data point $\mathbf{x}^\star$ as

$$q(y^*|\mathbf{X},\mathbf{y},\mathbf{x}^\star)= \int t(y^\star|f^\star)q(f^\star|\mathbf{X},\mathbf{y},\mathbf{x}^\star)\,df^\star=\Phi\left(\frac{y^\star m^\star}{\sqrt{1+v^\star}}\right), \tag{3}$$

where $q(f^\star|\mathbf{X},\mathbf{y},\mathbf{x}^\star)$ is the approximate predictive Gaussian distribution (the marginal of $q(\mathbf{f},f^\star|\mathbf{X},\mathbf{y},\mathbf{x}^\star)$ w.r.t. $\mathbf{f}$) with mean $m^\star=\mathbf{k}^{\star\top}(\mathbf{K}+\tilde{\mathbf{C}})^{-1}\tilde{\mathbf{m}}$ and variance $v^\star=k^{\star\star}-\mathbf{k}^{\star\top}(\mathbf{K}+\tilde{\mathbf{C}})^{-1}\mathbf{k}^\star$. In addition, the approximation to the marginal likelihood $p(\mathbf{y}|\mathbf{X})$ results in the normalization constant from Eq. (1), i.e. $q(\mathbf{y}|\mathbf{X})=Z_{\mathrm{EP}}$. The logarithm of $Z_{\mathrm{EP}}(\boldsymbol{\theta},\mathbf{X},\mathbf{y})$ and its derivatives can be used jointly with conjugate gradient updates to perform model selection under the evidence maximization framework. For a detailed presentation of GP including its implementation details, consult [2,13].

3. Predictive Active Set Selection

The EP algorithm is performed by iterative updates of each site approximation using the whole dataset $\{\mathbf{X},\mathbf{y}\}$. In the active set scenario on the other hand, we only want to approximate the posterior distribution in Eq. (1) using a small subset, the active set $\{\mathbf{X}_A,\mathbf{y}_A\}$. Since exploring all possible active sets is obviously intractable even for a fixed active set size M, the problem is how to select an active set that delivers a performance as good as possible within the available computing resources. The Informative Vector Machine (IVM) [8] for instance, computes in each iteration the differential entropy score for all data points not already part of the active set $\{\mathbf{X}_I,\mathbf{y}_I\}$ and perform updates by including the single point leading to a maximum score. Despite this greedy heuristic, IVM has proved to behave quite well in practice, giving the so far best reported GP performance on the USPS and MNIST tasks [8,9]. We propose an iterative approach in the same spirit with two main conceptual changes

- *Active set inclusion/deletion* based directly upon the data point weight in prediction. The 'representer theorem' for the mean prediction, discussed in Section 4, leads directly to the weight being expressed in terms of (a derivative of) the cavity predictive probability. This means that we can actually use the predictive distribution for a point in the inactive set to predict the weight it would have if it would be included in the active set. For classification we use the (cavity) predictive probability to decide upon deletion and inclusion because it is monotonically related to weight and it is a readily interpretable quantity.

- *Hyperparameter optimization* must be an integral part of algorithm, because the weights of the examples (and thus the active set) is conditioned on the hyperparameter values and vice versa. We therefore alternate between active set updates and hyperparameter optimization using several passes over the dataset.

Next we discuss the details of our (f)PASS-GP framework followed by a detailed comparison with the IVM. First we need to define rules for including and deleting points of the active set. As already mentioned, we use the predictive distribution in Eq. (3) for inclusions since data points with small predictive probability are more likely to contribute to improve the classifier performance and the quality of the active set. For deletions, we use the cavity predictive distribution in Eq. (2) because when examined carefully it can be seen as a leave-one-out estimator [14]. This means that points with cavity probability close to one do not contribute to the decision rule thus they can be discarded from the active set. With the two ranking measures set, i.e. Eqs. (2) and (3), we have essentially two possibilities. The first is to set probability thresholds on the distributions and let the model decide the size of the active set or we can rather specify directly the amount of inclusions/deletions. In PASS-GP, we include points in the active set with probability less than p_{inc} and remove them with probability greater than p_{del}. The appealing aspect of these thresholds is that they can be interpreted, for instance if we set $p_{\text{inc}} = 0.5$ we will include all misclassified observations in the current active set whereas if $p_{\text{inc}} = 0.6$ we will also include points near the decision boundary. We require two thresholds because we only want to remove points that as for the classifier are very easy to classify, so unlike p_{inc}, p_{del} must be close enough to one. In fPASS-GP, we want to keep the computational complexity of the classifier under control thereby we want the size of the active set to be fixed. For this purpose we only have to be sure that each active set update includes and removes the same amount of points. In practice we define p_{exc} as the exchange proportion w.r.t. M, meaning that each update replaces the fixed proportion of most hard to classify points in the inactive set with those more surely classified in the current active set. This update rule assumes that the active set is large enough to contain points in the active set with cavity probability close to one.

From a practical point of view, ranking every point in the inactive set at each iteration for inclusion could become prohibitive for large datasets. However, we still want to be able to cover the whole dataset rather than selecting a random subset for ranking. We then split the data into N_{sub} non-overlapping subsets and process each one of them in each iteration, such that each batch has something between 100 and 1000 data points.

Hyperparameter selection is a very important feature and needs to be done jointly with the active set update procedure. Algorithm 1 starts from a fixed randomly selected active set of size N_{init} (that is M in fPASS-GP), large enough to provide a good initial hyperparameter set values. Next we alternate between active set and hyperparameter optimization updates. Having in mind that we only expect small changes of the hyperparameters from one iteration to another, we reuse current values of $\boldsymbol{\theta}$ as initial values for the next iteration to speed-up the learning process. The addition and deletion rules in Algorithm 1 have parameters $\{p_{\text{inc}}, p_{\text{del}}\}$ and p_{exc} for PASS-GP and fPASS-GP, respectively.

Algorithm 1. Predictive Active Set Selection.

```
Input: $\{\mathbf{X},\mathbf{y}\}$, $\boldsymbol{\theta}$ and $\{N_{\text{init}}, N_{\text{sub}}, N_{\text{pass}}\}$
Input: $p_{\text{inc}}$ and $p_{\text{del}}$ (PASS-GP)
Input: $p_{\text{exc}}$ (fPASS-GP)
Output: $q(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A)$, $\boldsymbol{\theta}_{\text{new}}$ and $A$
begin
    $A \leftarrow \{1,\ldots,N_{\text{init}}\}$
    $\{\mathbf{X},\mathbf{y}\}_{\text{sub}}^{(1)},\ldots\{\mathbf{X},\mathbf{y}\}_{\text{sub}}^{(N_{\text{sub}})} \leftarrow \{\mathbf{X},\mathbf{y}\}$
    for $i=1$ to $N_{\text{pass}}$ do
        for $j=1$ to $N_{\text{sub}}$ do
            $\boldsymbol{\theta}_{\text{new}} = \text{argmax}_{\boldsymbol{\theta}} \log Z_{\text{EP}}(\boldsymbol{\theta},\mathbf{X}_A,\mathbf{y}_A)$
            Get $q(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A)$ and $q(y^*|\mathbf{X}_A,\mathbf{y}_A,\mathbf{x}^\star)$
            forall $\{\mathbf{x}_n,y_n\} \in \{\mathbf{X}_A,\mathbf{y}_A\}$ do
                if RemoveRule($q_{\backslash n}(y_n|\mathbf{X}_A,\mathbf{y}_{A\backslash n})$)
                then $A \leftarrow A \backslash \{n\}$
            end
            forall $\{\mathbf{x}_n,y_n\} \in \{\mathbf{X},\mathbf{y}\}_{\text{sub}}^{(j)}$ do
                if AdditionRule($q(y^*|\mathbf{X}_A,\mathbf{y}_A,\mathbf{x}^\star,\upsilon)$)
                then $A \leftarrow A \cup \{n\}$
            end
        end
    end
end
```

3.1. Differences between (f)PASS-GP and IVM

Since IVM is the closest relative of our active set selection method, we briefly discuss the main differences between the two: (i) the active set and thus the computational complexity is usually fixed beforehand in IVM. PASS-GP works with inclusion and deletion thresholds instead. (ii) IVM does not allow for deletions from the active set which is a clear disadvantage as points often become irrelevant at a later stage, when more points have been included. In (f)PASS-GP we can make an (almost) unbiased common ranking of all training points active as well as inactive, using a quantity that is meaningful and directly related to the weight of the training point in predictions. Using both inclusions/deletions and several passes over the training set makes (f)PASS-GP quite insensitive to the initial choice of active set. (iii) When the dataset is considerably large, IVM randomly selects a subset of points to be ranked from the inactive set, meaning that is likely that some points of the dataset are never considered for inclusion in the active set. (iv) The hyperparameter optimization is a part of the algorithm in (f)PASS-GP working on subsets of data between updates and iterating over the dataset several times. IVM makes a single inclusion per step and in principle stops when the limit for the active set is reached. (iv) In terms of complexity time per iteration IVM is faster than (f)PASS-GP, $\mathcal{O}(NM)$ against $\mathcal{O}(M^2(2+N/N_{\text{sub}}))$ where M is the size of A, however storage requirements are considerably lower, $\mathcal{O}(M^2)$ compared to $\mathcal{O}(NM)$.

4. Representer for mean prediction

The 'representer theorem' for the posterior mean of $\mathbf{f}$ [14], connects the predictive probability and the weight of a data point. Using that $p(\mathbf{f}|\mathbf{X}) = -\mathbf{K}(\partial/\partial\mathbf{f})p(\mathbf{f}|\mathbf{X})$, we get the exact relation for the posterior mean $\langle \mathbf{f} \rangle = \mathbf{K}\boldsymbol{\alpha}$ with the weight of element n being

$$\begin{aligned}\alpha_n &= \frac{1}{p(\mathbf{y}|\mathbf{X})}\int p(\mathbf{f}|\mathbf{X})\frac{\partial}{\partial f_n}p(\mathbf{y}|\mathbf{f})\,d\mathbf{f}\\ &= \frac{\langle p'(y_n|f_n)\rangle_{\backslash n}}{\langle p(y_n|f_n)\rangle_{\backslash n}} = \frac{\partial}{\partial h}\log\langle p(y_n|f_n+h)\rangle_{\backslash n}\Big|_{h=0},\end{aligned}$$

where $\langle \cdot \rangle_{\setminus n} = m_{\setminus n}$ denotes an average over a posterior without the n-th data point and $p'(y_n|f_n) = \partial p(y_n|f_n)/\partial f_n$. The final expression implies that the weight is nothing but the log derivative of the cavity predictive probability $\langle p(y_n|f_n)\rangle_{\setminus n} = p(y_n|\mathbf{X},\mathbf{y}_{\setminus n})$. For regression, $p(y_n|f_n) = \mathcal{N}(y_n|f_n,\sigma^2)$ and $\alpha_n = (y_n - \langle f_n\rangle_{\setminus n})(\sigma^2+\upsilon_{\setminus n})^{-1}$ with $\upsilon_{\setminus n} = \langle f_n^2\rangle_{\setminus n} - \langle f_n\rangle_{\setminus n}^2$. The element α_n will therefore be small when the cavity mean has a small deviation from the target relative to the variance. For a new data point pair $\{\mathbf{x}^\star, y^\star\}$, we can calculate the weight of this point *exactly*, replacing the cavity average with the full average in the expression above. We can therefore predict without any EP rerunning, how much weight this new point will have. For classification we can calculate the weight using the current EP approximation. When $z_n = y_n\langle f_n\rangle_{\setminus n}/\sqrt{1+\upsilon_{\setminus n}}$ is above ≈ 4, the cavity probability equation (2) approaches one and $\alpha_n \approx y_n \exp(-z_n^2/2)/\sqrt{2\pi(1+\upsilon_{\setminus n})}$. This fast decay indicates that GPC in many cases will be effectively sparse even though $\boldsymbol{\alpha}$ strictly does not contain zeros.

In the inclusion/deletion steps we rank data points according to their weights. For classification we can indeed use the predictive probability directly, since it is a monotonic function of the weight. Including a new data point will of course affect the value of all other weights as well leading to a rearrangement of their rank. Including multiple data points will also invalidate the predicted value of the weights (e.g. think of the extreme of two new data points being identical). We therefore have to recalculate the weights by retraining with EP for classification or simply updating the posterior for regression before going to the next step. If we have already an active set covering the decision regions well enough, this rearrangement step will amount to minor adjustments and the approximation will work well.

In this work we have only used the representer theorem for active set selection. It is also possible, but not tested here, to use all training points for prediction while only calculating the posterior on the active set. The inactive set weights are then simply set to the predicted values from the active set posterior. To get the full predictive probability one also has to calculate the contribution to the predictive variances which can be obtained by a similar theorem but for the predictive variance, see [14].

5. Marginal likelihood approximations

In this section we decompose the marginal likelihood in their active and inactive set contributions. We will argue that the contribution from the active set will dominate, justifying why we can limit ourselves to optimizing the hyperparameters over this set. In the following section we will investigate this assumption empirically. The marginal likelihood can be decomposed via the chain rule as

$$p(\mathbf{y}|\mathbf{X}) = p(\mathbf{y}_I|\mathbf{y}_A,\mathbf{X}_A,\mathbf{X}_I)p(\mathbf{y}_A|\mathbf{X}_A), \tag{4}$$

where we have used the marginalization property of GPs,

$$p(\mathbf{y}_A|\mathbf{X}) = \int p(\mathbf{y}_A|\mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A)d\mathbf{f}_A = p(\mathbf{y}_A|\mathbf{X}_A),$$

that we approximate as $q(\mathbf{y}_A|\mathbf{X}_A) = Z_{\mathrm{EP},A}$ and we identify it as the marginal likelihood for the active set A. The conditional marginal likelihood term can be written as

$$p(\mathbf{y}_I|\mathbf{y}_A,\mathbf{X}_A,\mathbf{X}_I) = \int p(\mathbf{y}_I|\mathbf{f}_I)p(\mathbf{f}_I|\mathbf{X}_I,\mathbf{X}_A,\mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A)\, d\mathbf{f}_A\, d\mathbf{f}_I, \tag{5}$$

where we used $p(\mathbf{f}|\mathbf{X}) = p(\mathbf{f}_I|\mathbf{X}_I,\mathbf{X}_A,\mathbf{f}_A)p(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A)$. We can make an EP approximation here just like in Eq. (1) by replacing the posterior $p(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A)$ by the multivariate Gaussian $q(\mathbf{f}_A|\mathbf{X}_A,\mathbf{y}_A) = \mathcal{N}(\mathbf{f}_A|\mathbf{m}_A,\mathbf{C}_{AA})$ where active set specific means and variances are found by EP. Marginalizing over $\mathbf{f}_A$ in Eq. (5) makes it now tractable

$$q(\mathbf{y}_I|\mathbf{y}_A,\mathbf{X}_A,\mathbf{X}_I) \approx \int p(\mathbf{y}_I|\mathbf{f}_I)\mathcal{N}(\mathbf{f}_I|\mathbf{m}_{I|A},\mathbf{C}_{II|A})\, d\mathbf{f}_I$$

with parameters

$$\mathbf{m}_{I|A} = \mathbf{K}_{IA}(\mathbf{K}_{AA}+\tilde{\mathbf{C}}_{AA})^{-1}\tilde{\mathbf{m}}_A,$$

$$\mathbf{C}_{II|A} = \mathbf{K}_{II} - \mathbf{K}_{IA}(\mathbf{K}_{AA}+\tilde{\mathbf{C}}_{AA})^{-1}\mathbf{K}_{AI},$$

where the tilted moments are as defined in Section 2. When the inactive set consists of a single example, we obtain the EP predictive distribution in Eq. (3), otherwise we have to solve for a new marginal likelihood. Denoting the marginal likelihood for a set $\{\mathbf{X},\mathbf{y}\}$ with a non-zero mean GP prior by

$$Z(\boldsymbol{\theta},\mathbf{X},\mathbf{y},\mathbf{m}) = \int p(\mathbf{y}|\mathbf{f})\mathcal{N}(\mathbf{f}|\mathbf{m},\mathbf{K})\, d\mathbf{f},$$

and its EP approximation by $Z_{\mathrm{EP}}(\boldsymbol{\theta},\mathbf{X},\mathbf{y},\mathbf{m})$, we can write the approximation to the marginal likelihood in Eq. (4) as

$$Z_{\mathrm{ACC}} \equiv Z_{\mathrm{EP}}(\boldsymbol{\theta},\mathbf{X},\mathbf{y}_I,\mathbf{m}_{I|A})Z_{\mathrm{EP}}(\boldsymbol{\theta},\mathbf{X},\mathbf{y}_A,\mathbf{0}).$$

Using this approximate decomposition reduces the complexity of EP from $\mathcal{O}(N^3 N_{\mathrm{pass}})$ to $\mathcal{O}((|I|^3+M^3)N_{\mathrm{pass}})$, where $|I|$ is the size of the inactive set. Unfortunately this is still too costly for large N. A final low complexity approximation to the marginal likelihood, that we denote by Z_{APP}, is to replace $p(\mathbf{y}_I|\mathbf{y}_A,\mathbf{X})$ with the product of marginals $\prod_{i\in I} p(y_i|\mathbf{y}_A,\mathbf{X}_A,\mathbf{x}_i)$. Empirically—see Fig. 3, this approximation turns out to be lower than the actual marginal likelihood, i.e. the joint distribution enforces the labels relative to the product of the marginals.

6. Experiments

The results presented in this section consist of several classification tasks performed on three well known datasets, namely USPS, MNIST and IJCNN. The first two correspond to handwritten digit databases while the third is a physical system inspired dataset assembled for the IJCNN 2001 neural network competition. We compare the two approaches introduced in Section 3 against the IVM and reduced complexity SVM (RSVM) [3]. We consider as performance measures not only classification errors, but also the error-cost trade-off and prediction uncertainty. We also present results for the approximation to the marginal likelihood of the full GP presented in Section 5. All experiments were performed on a 2.0 GHz desktop machine with 2 GB RAM.

6.1. USPS

The USPS digits database contains 9289 grayscale images of size 16 × 16 pixels, scaled and translated to fall within the range from -1 to 1. Here we adopt the traditional data splitting, i.e. 7291 observations for training and the remaining 2007 for testing. For each binary one-against-rest classifier we use the same model setup consisting of a squared exponential covariance matrix plus additive jitter

$$k(\mathbf{x}_i,\mathbf{x}_j) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\theta_2}\right) + \theta_3\delta_{ij}, \tag{6}$$

where $\delta_{ij} = 1$ if $i=j$ and zero otherwise. We have three hyperparameters in $\boldsymbol{\theta}$, namely, signal variance, characteristic length scale and jitter coefficient. Provided that the four active set methods being considered may depend upon random initialization we repeated all tasks 10 times. Individual settings for each method are

- PASS-GP: $N_{\text{init}} = 300$, $N_{\text{sub}} = 10$, $N_{\text{pass}} = 2$, $p_{inc} = 0.6$ and $p_{del} = 0.99$.
- fPASS-GP: $N_{\text{init}} = 300$, $N_{\text{sub}} = 10$, $N_{\text{pass}} = 4$, $p_{exc} = 0.02$. We allow fPASS-GP to perform more passes through the data because fPASS-GP progresses slower due to p_{exc} being small.
- RVM: $M=500$, $\boldsymbol{\theta} = [1\ 1/16\ 0]$, $C=10$ and $\kappa = 10$. More precisely, $\boldsymbol{\theta}$ and the regularization parameter, C, were obtained by grid search cross-validation, while κ was set to the value suggested by the authors of [3].
- IVM: $M=300$ and $N_{\text{pass}} = 8$. In the publicly available version of IVM, hyperparameter selection is done by alternating between full active set selection and hyperparameter optimization. Since IVM starts from an empty active set, it can be very sensitive to the initial values of $\boldsymbol{\theta}$. We experienced however that by adding a linear term, $\theta_4 \mathbf{x}_i^\top \mathbf{x}_j$, in the covariance matrix in Eq. (6) makes IVM quite insensitive to initialization. The results reported here include such a linear term because we found that using Eq. (6) alone makes the IVM to perform very poorly.

Fig. 1(a) shows mean test errors for every one-against-rest task using PASS-GP, fPASS-GP, RSVM, IVM and the full GPC with hyperparameter optimization. Besides, Fig. 1(b) shows the active set sizes for each digit using PASS-GP. From the figure, it can be seen that Gaussian process based active set methods perform similarly still slightly better than the RVM. The full GPC was only ran once due to its computational requirements, which explains the lack of error bars in Fig. 1(a). Furthermore, compared to fPASS-GP, IVM (M=300) and RSVM (M=500), PASS-GP seems to require smaller active sets to achieve similar classification performance. It is important to mention that we also tried larger values of M for the fixed active set algorithms but without any significant improvement in performance.

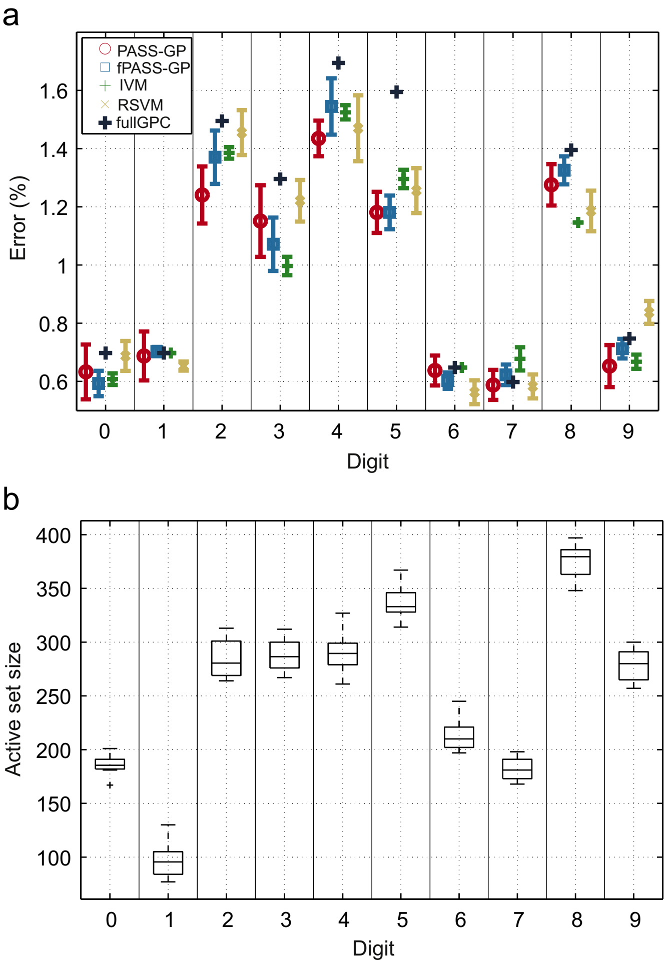

Fig. 1. Error rates and active set sizes for USPS data. (a) Mean classification errors for each digit using PASS-GP, fPASS-GP, RSVM, IVM and the full GPC with hyperparameter optimization. (b) Active set sizes for PASS-GP. Note that fPASS-GP and IVM use M=300, whereas RSVM uses M=500 for the results in (a). Error bars are standard deviations over 10 repetitions.

Fig. 2 shows classification errors for digits 2 and 4 against the others in top and bottom panels, respectively, as a function both of the active set size and running time. For fPASS-GP, RSVM and IVM we used $M = \{200, \ldots, 600\}$ and for PASS-GP we used $p_{inc} = \{0.2.0.3, \ldots, 0.9\}$. We included also the classification error obtained by the full GPC with hyperparameter optimization depicted as an horizontal dashed line. See [7] for a more detailed comparison between PASS-GP and full GPCs. Several features from Fig. 2 worth to be highlighted. (i) Gaussian process based methods approach the full GP for large values of M, as expected. (ii) Similar to Fig. 1(a), PASS-GP seems to consistently outperform fPASS-GP for similar sizes of M. (iii) For small values of M, RSVM and IVM perform better than our active set methods, however further increasing M does not considerably improves their performance. When M is small enough, it is very likely that our approaches are not able to obtain plausible estimates of the hyperparameters of the covariance function, thus its poor performance compared to RSVM that uses fixed values. Provided that the full GPC takes 8.6e5 s to run, PASS-GP and fPASS-GP are approximately three orders of magnitude faster than the full GPC with hyperparameter optimization, see [7]. From Fig. 2(b) and (d), we see that for similar active set sizes, PASS-GP and fPASS-GP have comparable computational costs as one may expect. Similarly, RSVM and IVM scale better than our active set selection methods. In terms of error-cost trade-off, RVM has a clear edge while the Gaussian process based methods can be regarded as comparable. It is important to note that for RVM, the difference in computational costs as seen in Fig. 2(b) and (d) should not be considered as significant since we are not counting the time used to obtain the parameters used by the RSVM, that unfortunately need to be selected by expensive grid search with cross-validation. The IVM turned out to be time-wise comparable to our active set methods not because its selection procedure but due to the hyperparameter optimization scheme used.

The results obtained on USPS suggest that (f)PASS-GP is performing slightly better than the full GPC. This could be due to numerical instability produced by the size of the problem, by the iterative nature of the EP algorithm and/or not enough iterations for the hyperparameter selection procedure. However, it could also mean that optimizing on the active set achieves a better "local" fit around the decision boundary region. A priori, one cannot expect that a single set of hyperparameters is able to describe all regions in input space, thus every possible active set. The same kind of local improvement observed here was also reported by [15,4] using GPC auxiliary set methods.

Combining the ten binary tasks into a one-against-rest multiclass classifier, PASS-GP obtained 4.51 ± 0.17% which is comparable or better[1] than 4.61 ± 0.11% by fPASS-GP, 4.88 ± 0.12% by RSVM and 4.38 ± 0.11% by IVM. Baselines are, 5.13% by GPC with hyperparameter optimization, 4.78% by GPC with fixed $\boldsymbol{\theta}$ and 9.75 ± 0.40% by GPC with random active set selection. Other relevant results found in the literature include 5.15% by online GP [16] and 4.98% by IVM with randomized greedy selection [9].

[1] Assuming independent errors, the standard deviation on the performance is $\sqrt{\epsilon(1-\epsilon)/N_{\text{test}}}$ giving approximately 0.4% for USPS and 0.1% for MNIST.

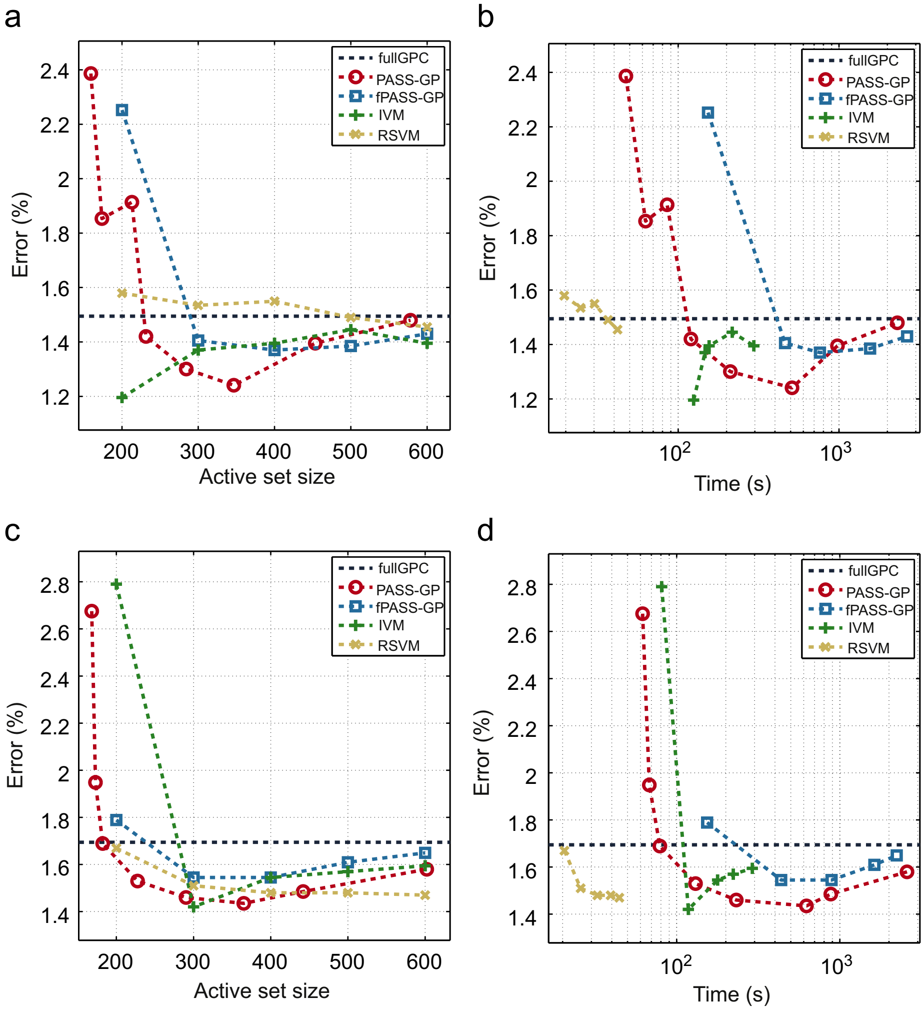

Fig. 2. Results for selected individual digits of USPS data. (a) and (c) show mean classification errors as a function of the active set size for digits 2 and 4 vs the rest, respectively. (b) and (c) show mean classification errors as a function of the running time, matching panels (a) and (c). The horizontal dashed line in all plots is the performance of the full GPC with hyperparameter selection. In both cases, the full GPC took approximately 8.6e5 s [7]. Values represent averages over ten independent repetitions with error bars omitted for clarity.

All three Gaussian process based methods are comparable with state-of-the-art techniques such as SVM, see [17]. It is worth pointing out that the best result we could obtain from IVM using the covariance matrix in Eq. (6) was $6.27 \pm 0.21\%$ for $M = 1500$ which is substantially worse than the performance of the full GPC. As reference, it has been shown that the human error rate is approximately 2.5%.

Next we want to evaluate the two approximations to the marginal likelihood proposed in Section 5. We proceed by computing the accurate but expensive approximation Z_{ACC}, the less accurate but affordable Z_{APP} and the marginal likelihood of the full GPC and the active set, simply denoted as Z_{EP} and $Z_{\mathrm{EP,A}}$, respectively. In order to show how the approximations depend on the size of the active set, we compute them for $p_{\mathrm{inc}} = \{0.5, 0.6, \ldots, 0.9, 0.99, 1\}$, with $p_{\mathrm{inc}} = 1$ being the full GPC. Fig. 3 shows that the three approximations approach the marginal likelihood of the full GPC as the inclusion threshold and so the active set size increases. As expected, Z_{ACC} is the best approximation, however the computational effort needed to compute it is roughly two orders of magnitude larger compared to the cost of computing Z_{APP} and $Z_{\mathrm{EP,A}}$. It is very interesting that even with large values of $p_{\mathrm{inc}} = 0.99$ the size of the active set remains below 10% of the training data and the contribution to the log-marginal likelihood from the inactive $Z_{\mathrm{EP}}(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}_I, \mathbf{m}_{I|A})$ set basically vanishes, since Z_{APP} and $Z_{\mathrm{EP,A}}$ are essentially the same.

Finally, we want to assess the uncertainty of the predictions made by the Gaussian process based methods by means of comparing the predictive probabilities with the true outcomes. Fig. 4 shows estimated log predictive densities for PASS-GP, fPASS-GP, IVM and the full GPC, using all USPS predictions made on the test separated into correct and incorrect predictions. Assuming no labeling errors, the true density consists of two point mass densities at {0,1} provided our one-against-rest setting. As one might expect, the full GPC achieves the best approximation, followed by fPASS-GP and PASS-GP. IVM suggests more predictive uncertainty because of the two "spurious" modes in Fig. 4(a). Another way to assess the predictive uncertainty is to compute Brier scores, that measures the average of square deviations between estimated and true predictive probabilities. For the USPS dataset we obtained: 0.53 ± 0.03, 0.27 ± 0.01, 0.71 ± 0.02 and 0.14 ± 0.00 for PASS-GP, fPASS-GP, IVM and full GPC, respectively. Note that the Brier scores are in agreement to what we observe in Fig. 4.

6.2. MNIST

The MNIST digits database has 60 000 and 10 000 as training and testing examples, respectively. Each example is a gray-scale image of 28 × 28 pixels. The estimated human test error is around

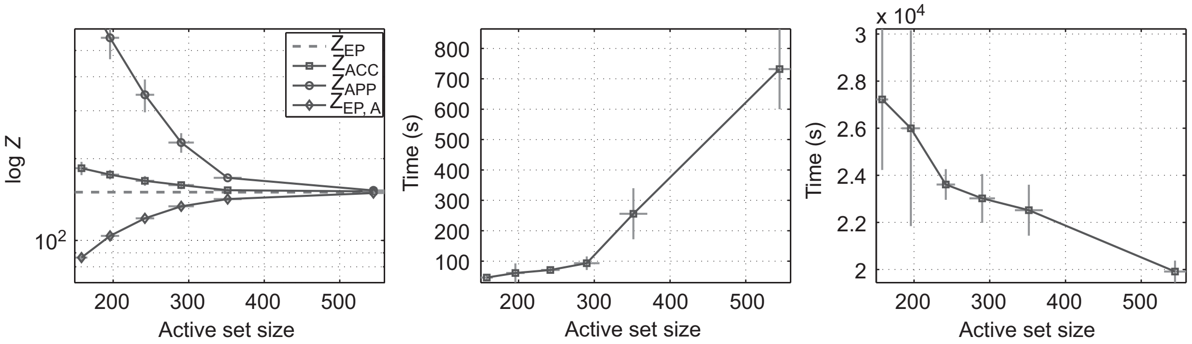

Fig. 3. Marginal log-likelihood approximations as a function of the active set size for digit 3 vs the rest. The plots show means and standard deviations (error bars) over ten repetitions. Each marker indicates a different inclusion threshold $p_{\text{inc}} = \{0.5, 0.6, \ldots, 0.9, 0.99\}$. In the left panel, Z_{EP} is for the full GPC ($p_{\text{inc}} = 1$), $Z_{\text{EP,A}}$ for the active set only and the remaining two, Z_{ACC} and Z_{APP}, are the proposed approximations. The middle and right panels show computation times required to compute $\{Z_{\text{APP}}, Z_{\text{EP,A}}\}$ and Z_{ACC}, respectively.

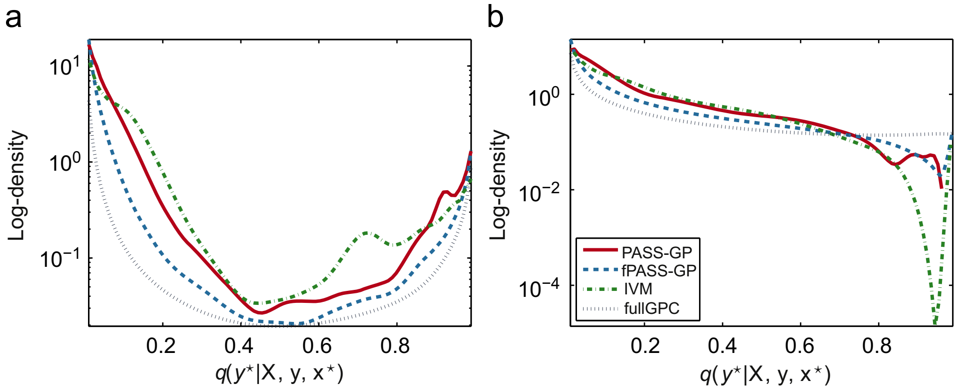

Fig. 4. Predictive density estimation for USPS data. Densities for correct and incorrect predictions are shown separately in (a) and (b), respectively. The ground truth for (a) is a two point mass mixture at {0,1} and a flat distribution for (b).

0.2%. The settings used for the algorithm are nearly the same as those for USPS with only two differences. N_{sub} is set 100 since the training set in MNIST is almost ten times larger than USPS and we are not updating the hyperparameter in each iteration but every 10-th, in order to make the training process faster. We also ran our algorithm with hyperparameter updates every single iteration without any noticeable improvement in performance (results not shown). Fig. 5 shows test error rates, active set sizes, multi-class errors and running times for each binary classifier based on PASS-GP, fPASS-GP and RSVM using a 9-th degree polynomial covariance function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^9.$$

We use this covariance matrix instead for the standard squared exponential from Eq. (6), because a polynomial covariance is well known for providing optimal results for the MNIST dataset [18]. Results for the squared exponential covariance function can be found in [7] and confirm that the polynomial covariance behave slightly better for this dataset. For IVM we could not make the polynomial covariance to work properly, thus we decided to use Eq. (6) plus a linear term like in the USPS experiment.

From Fig. 5(b) it can be seen that in every case the size of the active set is less than 4% of the training set. The results for fPASS-GP and RSVM were obtained using $M = 2000$. We did try for larger values of M but the reduction in error was not significant compared to the overhead in computational cost. Fig. 5(a) shows the classification error for each digit. The performance of the three approaches considered is comparable but letting PASS-GP with an edge over the other two, both in terms of error and variances. Fig. 5(c) shows the results of combining the ten binary classifiers. Again, PASS-GP behaves slightly better than the others, however when looking at the run times in Fig. 5(d) we can see that RSVM is computationally more affordable than our approaches, even more considering that it uses $M = 2000$. Comparing PASS-GP to fPASS-GP, the former has a smaller mean run time but with larger variance compared to the more expensive fPASS-GP. fPASS-GP is more stable time-wise, but takes more time because it uses a fixed $M = 2000$. As far as the authors know these are the first GP based results on MNIST using the whole database. IVM [8] with sub-sampled images of size 13 × 13 has been tried to produce a test error rate of $1.54 \pm 0.04\%$. Seeger [9] made additional tests on some digits (5, 8 and 9) on the full size images without any further improvement. On the other hand, PASS-GP is again comparable with state-of-the-art techniques not including preprocessing stages and/or data augmentation, for instance SVM is 1.4% and 1.22% using RBF and a 9-th degree polynomial kernel, respectively. The reported sizes of support vector sets are approximately two times larger than our active sets [18].

6.3. *Incorporating invariances*

It has been shown that a good way to improve the overall performance of a classifier is to incorporate additional prior knowledge in the training procedure particularly by means of externally handling invariances of the data. In [18], it is shown that instead of just dealing with the invariances by augmenting the original dataset—which turns out to be infeasible in many cases, it is better to augment only the support vector set of a SVM.

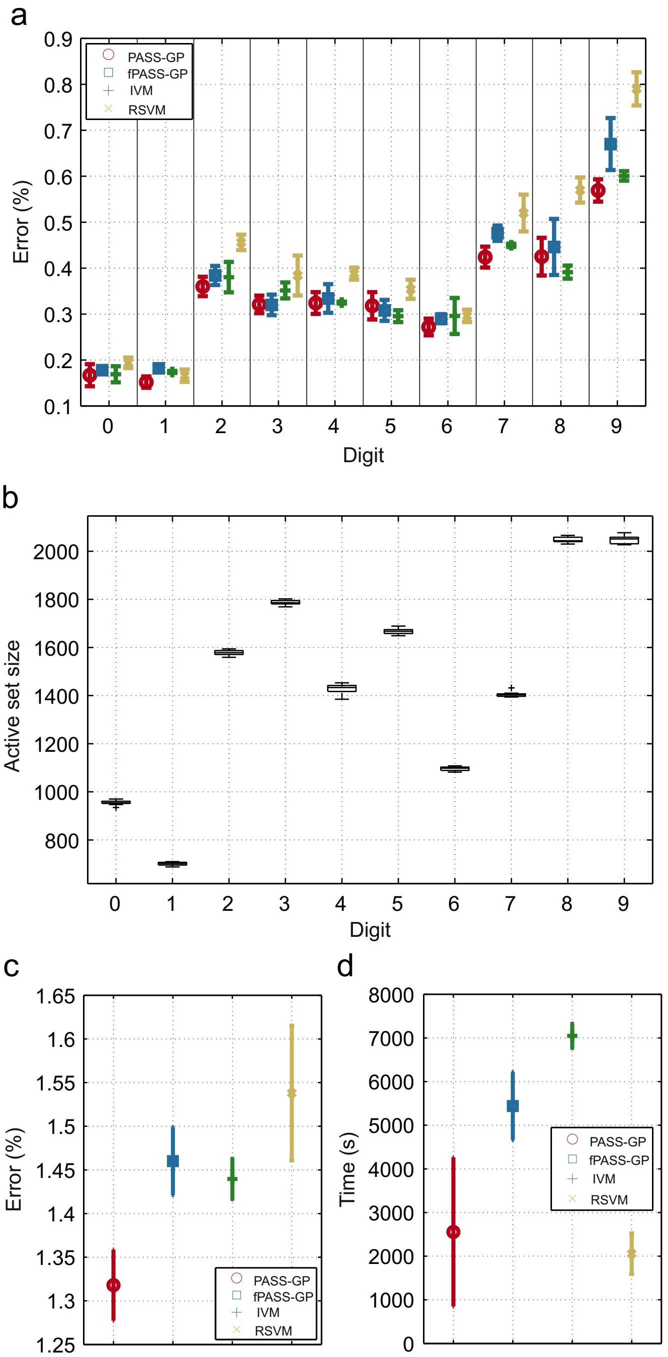

Fig. 5. Error rates, active set sizes and run times for MNIST data. (a) Mean classification errors for each digit task using PASS-GP, fPASS-GP, RSVM and IVM. (b) Active set sizes for PASS-GP. Note that fPASS-GP, RSVM and IVM use $M=2000$. (c) Mean multi-class classification errors and (d) average timings over one-against-the-rest classifiers and repetitions. Error bars in (a), (c) and (d) are standard deviations computed over 10 repetitions of the experiment.

We therefore try the same procedure as suggested in [18] consisting of four 1-pixel translations (left, up, right and down directions) on each element of the active set for USPS and eight 1-pixel translations (including diagonals as well) for MNIST, resulting in new training sets of size $5\times M$ and $9\times M$, accordingly. In this case we have used the same settings as in the previous experiments with only two differences. First, the hyperparameters have been set to those found using the original dataset. Second, we made the important observation that in order to get a performance improvement a large active set was needed. For training on the augmented dataset we increased p_{inc} from 0.6 to 0.99 for USPS and 0.9 for MNIST. We conjecture that we can get even better performance—at the expense of a substantial increase in complexity, by increasing p_{inc} in the initial run to get a larger initial active set to work with.

Results in Table 1 show that performance-wise, PASS-GP reached $3.35\pm0.03\%$ for USPS and $0.86\pm0.02\%$ for MINST on the multi-class task, what is comparable to state-of-the-art techniques. For instance SVM obtained 3.2% on USPS and 0.68% on MNIST with an equivalent procedure. The difference in performance is probably due to our active set not being large enough, since support set sizes reported for SVMs are typically twice as large [18].

6.4. IJCNN

As final experiment, we want to compare fPASS-GP, RSVM and IVM on a common ground. For this purpose we use the IJCNN dataset which is widely used by the SVM research community. It consists of 49 990 training examples, 91 701 test examples and each observation counts with 22 features. We consider $M=\{100,200,\ldots,1000\}$ with squared covariance function and fixed hyperparameters, the latter using the values suggested in [3], that is $\boldsymbol{\theta}=[1\ 1/8\ 1/16]$ for fPASS-GP and IVM, and $\boldsymbol{\theta}=[1\ 1/8\ 0]$, $C=16$ for RSVM. For IVM we include a linear term as in the previous experiments with $\theta_4=1$. Besides, each setting was repeated 10 times to collect statistics. Fig. 6 summarizes the results obtained. More specifically, Fig. 6(a) shows the mean classification error as a function of the active set. We can see that fPASS-GP is slightly better than RSVM and IVM in the entire range of M, besides the former seems to be particularly good for small values of M. When we plot mean errors as a function of running times—as a proxy for the computational cost, we see that there exist two regimes, one for small values of M where fPASS-GP outperforms RSVM and IVM, and the other where the cubic complexity of the GPCs start hurting fPASS-GP, thus letting RVM and IVM with a better error-cost trade-off.

7. Discussion

We have proposed a framework for active set selection in GPC. The core of our active set update rule is that the predictive distribution of a GPC can be used to quantify the relative weight of points in the active set that can be marked for deletion or new points from the active set with low predictive probabilities, that

Table 1
Results for USPS and MNIST using PASS-GP and active set invariances. Figures are averages over 10 and 5 repetitions, respectively.

Digit	0	1	2	3	4	5	6	7	8	9
USPS (%)	0.63	0.38	1.01	0.69	0.93	1.16	0.51	0.37	0.59	0.65
Active set	870	442	1251	1316	1654	1425	1242	987	1532	1281
MNIST (%)	0.14	0.14	0.24	0.24	0.29	0.22	0.17	0.35	0.29	0.35
Active set	6505	4372	11 401	12 988	9776	11 960	7360	9872	15 194	14 790

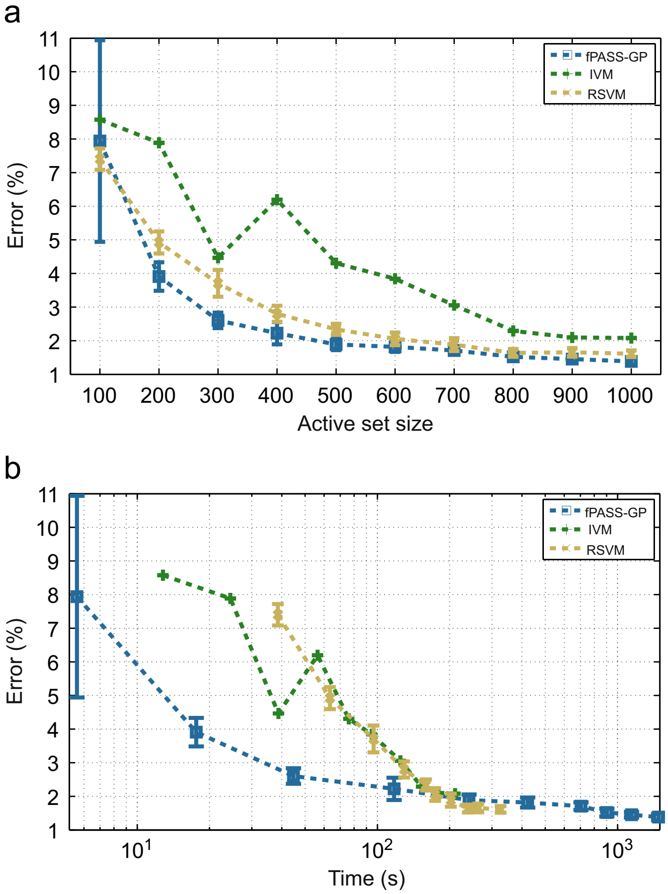

Fig. 6. Error rates and run times for IJCNN data. (a) Mean classification error as a function of the active set size using fPASS-GP, RSVM and IVM. (b) Mean classification error as a function of the run time. Error bars correspond to standard deviations computed over 10 repetitions of the experiment.

make them ideal for inclusion. The algorithmic skeleton of our framework consists on two alternating steps, namely active set updates and hyperparameter optimization. We designed two active set update criteria that target two different practical scenarios. The first we called PASS-GP focuses on interpretability of the parameters of the update rule by thresholding the predictive distributions of GPC. The second acknowledges that in some applications having a fixed computational cost is key, thus fPASS-GP keeps the size of the active set fixed so the overall cost and memory requirements can be known beforehand.

We presented theoretical and practical support that our active set selection strategy is efficient while still retaining the most appealing benefits of GPC: prediction uncertainty, model selection, prior knowledge leverage and state-of-the-art performance. Compared to other approximative methods, although slower than IVM [8] and RSVM [3], PASS-GP provides better results. We did not consider any auxiliary set method like FITC [4] because for task of the size like for example MNIST or IJCNN, it is prohibitive. Additionally, we have noticed in practice that our approximation is quite insensitive to the initial active set selection and also that more than two or three passes through the data do not yield improved performance nor large active set sizes. The code used in this work is based on the Matlab toolbox provided with [2] and is publicly available at http://cogsys.imm.dtu.dk/passgp.

The not so satisfying feature of active set approximations is that we are ignoring some of the training data. Although some of our findings on the USPS dataset actually suggest that this can be beneficial for performance, it is of interest to make a modified version where the inactive set is used approximately in a cost efficient way. The representer theorem for the mean prediction and the approximations for marginal likelihood discussed in this paper might give inspiration for such methods. In conclusion, efficient methods for GPs are still much in need when the data is abundant such as in ordinal regression for collaborative filtering.

References

[1] J. Quiñonero-Candela, C.E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, J. Mach. Learn. Res. 6 (2005) 1939–1959.

[2] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, MA, 2006.

[3] S.S. Keerthi, O. Chappelle, D. DeCoste, Building support vector machines with reduced classifier complexity, J. Mach. Learn. Res. 7 (2006) 1493–1515.

[4] A. Naish-Guzman, S. Holden, The generalized FITC approximation, in: J.C. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, vol. 20, MIT Press, Cambridge, MA, 2008, pp. 1057–1064.

[5] T. Joachims, C.-N.J. Yu, Sparse kernel SVMs via cutting-plane training, Mach. Learn. 76 (2009) 179–193.

[6] A. Kapoor, K. Grauman, R. Urtasun, T. Darrell, Gaussian processes for object categorization, Int. J. Comput. Vision 88 (2) (2010) 169–188.

[7] R. Henao, O. Winther, PASS-GP: predictive active set selection for Gaussian processes. in: 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)2010, pp. 148–153.

[8] N.D. Lawrence, M. Seeger, R. Herbrich, Fast sparse Gaussian process methods: the informative vector machine, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, The MIT Press, Cambridge, MA, 2003, pp. 600–616.

[9] M. Seeger, Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations, Ph.D. Thesis, University of Edinburg, 2003.

[10] N.D. Lawrence, J.C. Platt, M.I. Jordan, Extensions of the informative vector machine, in: J. Winkler, N.D. Lawrence, M. Niranjan (Eds.), Proceedings of the Sheffield Machine Learning Workshop, Springer-Verlag, Berlin, 2005.

[11] M. Titsias, Variational learning of inducing variables in sparse Gaussian processes. in: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, vol. 5, 2009, pp. 567–574.

[12] Z. Zhang, G. Dai, M.I. Jordan, Bayesian generalized kernel mixed models, J. Mach. Learn. Res. 12 (2011) 111–139.

[13] M. Kuss, C.E. Rasmussen, Assessing approximate inference for binary Gaussian process classification, J. Mach. Learn. Res. 6 (2005) 1679–1704.

[14] M. Opper, O. Winther, Gaussian processes for classification: mean-field algorithms, Neural Comput. 12 (11) (2000) 2655–2684.

[15] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in: Y. Weiss, B. Schölkopf, J.C. Platt (Eds.), Advances in Neural Information Processing Systems, vol. 18, The MIT Press, 2006.

[16] L. Csató, Gaussian Processes—Iterative Sparse Approximations, Ph.D. Thesis, Aston University, 2002.

[17] B. Schölkopf, A.J. Smola, Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond, The MIT Press, 2001.

[18] D. DeCoste, B. Schölkopf, Training invariant support vector machines, Mach. Learn. 46 (1–3) (2002) 161–190.

Ricardo Henao received the degree in electronic engineering and masters in industrial automation from the Universidad Nacional de Colombia, Manizales, in 2002 and 2004, respectively. He is currently a Ph.D. student at the department of DTU Informatics of the Technical University of Denmark. His research interests include graphical model learning and supervised modeling with applications to bioinformatics.

Ole Winther works in machine learning research with applications in bioinformatics, data mining, green technology and collaborative filtering. Ole Winther, M.Sc. physics and computer science '94 and Ph.D. physics '98, both at University of Copenhagen (KU), has published more than 60 scientific papers and has previously held positions at Lund University and Center for Biological Sequence Analysis, Technical University of Denmark (DTU). He currently holds joint positions as associate professor at Cognitive System, DTU Informatics and group leader at Bioinformatics, KU.