Multi-class Classification with Dependent Gaussian Processes

Mykhaylo Andriluka[1], Lorenz Weizsäcker[1], and Thomas Hofmann[2]

[1] Darmstadt University of Technology
Department of Computer Science
Hochschulstrasse 10
64289 Darmstadt, Germany
(e-mail: andriluk@mis.informatik.tu-darmstadt.de)
[2] Google Inc.
Freigutstrasse 12
8002 Zurich, Switzerland
(e-mail: thofmann@google.com)

Abstract. We present a novel multi-output Gaussian process model for multi-class classification. We build on the formulation of Gaussian processes via convolution of white Gaussian noise processes with a parameterized kernel and present a new class of multi-output covariance functions. The latter allow for greater flexibility in modelling relationships between outputs while being parsimonious with regard to the number of model parameters. We apply the model to multi-class Gaussian process classification using a sparse approximation based on the informative vector framework and investigate, both analytically as well as empirically, a scenario where our multi-class classifier performs better than combining independently trained binary classifiers.

Keywords: Gaussian processes, informative vector machine, multi-class classification.

1 Introduction

Many real-world problems that use machine learning and statistical data analysis techniques involve simultaneously predicting several dependent variables. Possible examples range from interpolation problems for values of spatially dispersed temperature sensors to modelling of user ranking functions in recommender system [Yu and Tresp, 2005] or the analysis of multi-spectral satellite imagery [Pardo-Igúzquiza *et al.*, 2006].

Recently several authors proposed methods to utilize and combine the training data available for different tasks using Gaussian process models in which dependencies between outputs are modelled using a hierarchical Bayesian framework. [Lawrence and Platt, 2004] propose a multi-output Gaussian process model called multi-task informative vector machine (MTIVM) in which the sole dependence between different tasks comes from sharing the parameters of the underlying covariance function. They obtain a point estimates for these parameters by optimizing the joint log-likelihood of the training data from multiple tasks.

Multiple dependent Gaussian processes can also be obtained by assuming that each process is a different transformation of the same set of underlying independent Gaussian processes; several authors have recently pursued this avenue to model dependent outputs. In [Boyle and Frean, 2005a] each dependent Gaussian process is assumed to be a convolution of the same white noise process with different kernel and in [Teh *et al.*, 2005] a model with an intermediate layer of latent Gaussian processes is introduced. Here, dependent Gaussian processes are obtained as linear combinations of the independent processes forming the intermediate layer.

In this paper we contribute to the development of multi-task Gaussian process models in several ways. First, we present a way to systematically derive covariance functions for multi-output Gaussian processes covering the method proposed in [Boyle and Frean, 2005a] as a special case. Second, we derive a classifier that captures the dependencies between different class labels using such covariance functions. Our model is somewhat similar to the MTIVM approach proposed in [Lawrence and Platt, 2004] where the authors also have used IVM to select the most informative points across training data for different tasks. However in our case the structure of the underlying Gaussian process is richer and explicitly incorporates dependencies between outputs.

The rest of the paper is organized as follows: We review the formulation of Gaussian processes for multi-output data and derive covariance functions for multiple dependent outputs in Section 2. Applications of this model to artificial data as well as a real world dataset are presented in Section 3. Finally, Section 4 presents a conclusion and an outlook on future work.

2 Multi-output Gaussian Processes

Gaussian processes are collections of random variables indexed by the elements of an index set $\mathcal{X}$ such that the joint probability distribution of any finite number of variables is multivariate Gaussian. It is conceptually straightforward to model multi-task data with a single joint Gaussian process by including the output number in an additional dimension of the index set:

Definition 1. Given a set $\mathcal{X}$, a Gaussian process $(\mathrm{f}(\mathbf{s}))_{\mathbf{s}\in S}$ with index set $S = \{1, 2, \ldots, M\} \times \mathcal{X}$ is called *multi-output Gaussian process with input space $\mathcal{X}$ and M outputs.*

However, it is not obvious how to define a covariance function $\mathrm{Cov} : S^2 \to \mathbb{R}$. This function should be positive definite on the set S, which means that covariance matrix defined by Cov should be positive semi-definite for any finite set of elements from the index set S. For independent outputs

$i, j \in \{1, 2, \ldots, M\}$ we have that $\mathrm{Cov}((i, \mathbf{x}_a), (j, \mathbf{x}_b)) = 0$ for all $\mathbf{x}_a, \mathbf{x}_b \in \mathcal{X}$ so that in this case Cov is positive definite if and only if it is positive definite on all $S_i = \{(i, \mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$, $i = 1, \ldots, M$. We devote the rest of this section to the key question of how to consistently define $\mathrm{Cov}((i, \mathbf{x}_a), (j, \mathbf{x}_b))$ in the more general case where outputs are not independent. From now on we assume that $\mathcal{X} = \mathbb{R}^d$ for some $d \in \mathbb{N}$.

For a given isotropic covariance function of single-output Gaussian processes we can devise a covariance function for multi-output process in the following way:

Proposition 1. *Assume that* $\mathrm{C}(\tau)$ *is an isotropic covariance function on* $\mathbb{R}^d$, *for any* $d \in \mathbb{N}$. *Further, let* S *be an index set as in Definition 1 and the function* $\mathrm{Cov} : S^2 \to \mathbb{R}$ *be given by*

$$\mathrm{Cov}((i, \mathbf{x}_a), (j, \mathbf{x}_b)) = \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} \mathrm{C}(\sqrt{Q_{ij}(\mathbf{x}_a, \mathbf{x}_b)}), \tag{1}$$

with

$$\mathrm{Q}_{ij}(\mathbf{x}_a, \mathbf{x}_b) = (\mathbf{x}_a - \mathbf{x}_b)^T A_i (A_i + A_j)^{-1} A_j (\mathbf{x}_a - \mathbf{x}_b), \tag{2}$$

$v_i, v_j \in \mathbb{R}$ *and arbitrary positive definite matrices* A_i, $i = 1, \ldots, M$. *Then,* Cov *is a positive definite function on* S^2.

The proof is given in the appendix and uses an argument similar to the one used in [Paciorek, 2003] although in somewhat different context.

We refer to the single-output covariance function C as the generating covariance function. The argument of C is a distance between $\mathbf{x}_a$ and $\mathbf{x}_b$ induced by the scalar product that is defined by the positive definite matrix $B^T B = A_i(A_i + A_j)^{-1} A_j$. If all A_i and hence B are diagonal, we can interpret the diagonal entries of B as automatic relevance determination hyperparameters that determine the relative importance of different feature dimensions for the cross-correlation between outputs.

Notice that the covariance function used in [Boyle and Frean, 2005a] which is given by

$$\mathrm{Cov}((i, \mathbf{x}_a), (j, \mathbf{x}_b)) = \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} e^{-\frac{1}{2} \mathrm{Q}_{ij}(\mathbf{x}_a, \mathbf{x}_b)} \tag{3}$$

can be obtained from Proposition 1 using a squared exponential as the generating covariance function. Intuitively, all covariance functions which can be obtained via Proposition 1 are scale mixtures of covariance function given by (3).

Figure 2 shows several samples from multi-output Gaussian process with 3 outputs and covariance function generated by different single-output covariance functions (see [Rasmussen and Williams, 2006] for their definition and properties).

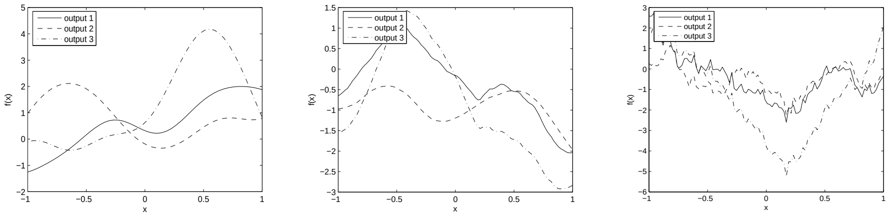

Fig. 1: Samples from 3-output MGP with covariance functions generated by (from left to right) squared-exponential, Matérn (with $\nu = 3/2$) and exponential single-output covariance functions.

For a fixed output index i, the covariance function

$$\mathrm{Cov}_i(\mathbf{x}_a, \mathbf{x}_b) = \mathrm{Cov}((i, \mathbf{x}_a), (i, \mathbf{x}_b))$$

is of the same type as the generating covariance function C. Hence samples from output i have the same continuity and differentiability properties as samples from a Gaussian process with covariance function given by C. This is a very valuable property since it allows us to choose suitable covariance function for multi-output Gaussian process in a modular way using *a priori* knowledge about the nature of the data source.

3 Experiments

In this section we would like to present two experiments on multi-class classification with dependent Gaussian processes. In each of the experiments we first select a suitable Gaussian process prior by choosing a particular covariance function and then find the point estimates for its hyperparameters by optimizing the marginal likelihood w.r.t. the training data using the scaled conjugate gradients method. We use the informative vector machine (IVM) to compute sparse approximations of non-Gaussian likelihoods that arise in the case of the probit classification noise model. Handling multiple outputs by extending the index set of the Gaussian process allows us to use IVM almost without any modifications. Since selection of informative vectors depends on the values of hyperparameters and hyperparameters are optimized using the set of informative vectors we do several interchanging iterations of informative vectors selection and hyperparameter optimization. For more details on IVM we refer to [Lawrence *et al.*, 2005].

Toy dataset: In our toy example we choose an input set as $\mathcal{X} = [-1, 1] \times [-1, 1]$ and consider 4 binary classification tasks. Each of the regions $[0, 0.7]^2$, $[-0.7, 0] \times [0, 0.7]$, $[-0.7, 0]^2$ and $[0, 0.7] \times [-0.7, 0]$ corresponds to one binary classification task in which a point is classified positively if it is

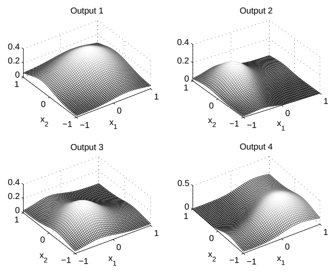

Fig. 2: Posterior probabilities for each output. Plot corresponding to output i shows $\mathrm{p}(y_i(\mathbf{x}_*) = 1|\{\mathcal{D}_j|j \neq i\})$, where $y_i(\mathbf{x}_*)$ is a classification label given to $\mathbf{x}_* \in \mathcal{X}$ by task i and $\mathcal{D}_i$ denotes training data for task i.

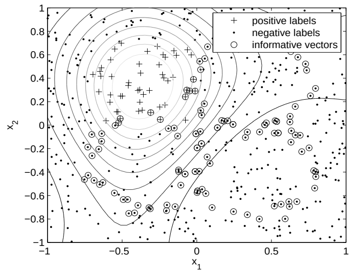

Fig. 3: Training set with positive and negative labels of second task and informative vectors chosen by IVM for sparse approximation of $\mathrm{p}(y_2(\mathbf{x}_*) = 1|\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4)$

inside the region and negatively otherwise. We generate a training set by randomly choosing 500 points from $\mathcal{X}$ and labelling them accordingly. This results in the total number of 2000 training points. The maximum number of informative vectors is set to 50 for each task. We use the sum of two squared-exponential multi-output covariance functions given by (3) with diagonal matrices $A_i, i = 1, \ldots, 4$ resulting in 24 hyperparameters.

In Fig. 2 we visualize the dependencies between tasks learned by our model. For each point and each of four tasks we plot the posterior probability of a point to have positive label obtained by conditioning the prior distribution only on the training points from other three tasks. Such posterior is equal to prior if there are no dependencies between tasks. In Fig. 3 the informative vectors are shown which were selected from the training sets of first, third, fourth tasks for the computation of posterior of second task. Notice how IVM selects points along the separation boundary of each tasks.

We conducted other experiments similar to the one described above, in particular one in which we chose different training points for each task. Further, we also used covariance function in which additional offset hyperparameters were introduced so that instead of learning the anti-correlations between outputs the model was able to learn correlations between one output and a shifted version of the other. In all these experiments we were able to learn the dependencies between tasks.

Text Classification: In our second experiment we take the well known Reuters21578 text categorization dataset. After initial preprocessing the vocabulary contains 12113 terms. The dimensionality of the documents is then further reduced to 50 using latent semantic analysis. From the 10377 available documents we choose 2500 documents for training and use the rest for testing.

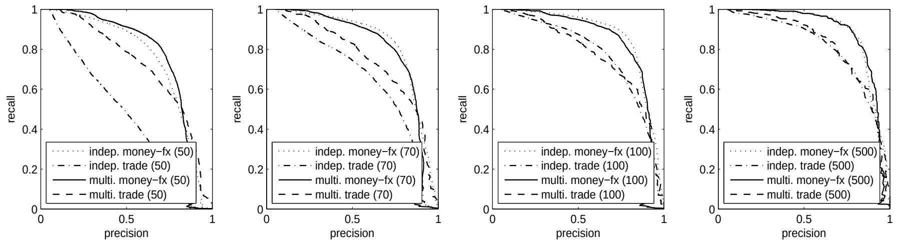

Fig. 4: Precision/recall curves for each category and independent and multi-output classifiers. Plots (from left to right) correspond to 50, 70, 100, and 500 informative vectors chosen by IVM.

We choose two categories of documents with labels “money-fx” and “trade”. The total number of positive documents for each of the two categories equals 684 and 514 respectively, from which 154 and 113 were randomly selected in the training set. The two categories are not mutually exclusive, the dataset contains 46 documents included in both categories simultaneously. For each of the categories we train an independent Gaussian process classifier with squared exponential covariance function and automatic relevance determination hyperparameters which control the influence of each input dimension on the classification outcome. This results in the total of 102 hyperparameters for both classes. Additionally we also train a 2-output Gaussian process classifier with covariance function given by equation (3) and diagonal matrices A_1, A_2 so that the multi-output model has the same number of 102 hyperparameters. In order to estimate the hyperparameters reliably we select 1000 informative vectors at each iteration of the hyperparameter optimization.

On the Fig. 4 we show the precision/recall curves for classifiers obtained by setting the number of informative vectors selected by IVM to different values. The multi-output classifier is significantly more tolerant to small number of informative vectors than independent classifiers. We have compared the number of informative vectors with positive labels to the total number of informative vectors selected by IVM from the training set of each class and found that for independent classifiers their ratio is typically close to 0.5. For multi-output classifier this ratio is higher with number of positively labelled samples dominating in the total number of selected informative vectors (ratios are typically around 0.75 - 0.8). We suppose that in the case of multi-output classifier the negative training points are shared between the tasks which allows each classifier to select more positive training points. This could be the reason for higher generalization performance of multi-output classifier when the number of informative vectors is kept small.

4 Conclusion

In this paper we have presented a multi-class classifier which uses dependent Gaussian processes in order to represent relations between different learning tasks. We have demonstrated that such classifier can learn the relationships between different classification tasks and utilize them in order to improve generalization performance in certain special cases. We have also shown that flexible modelling of prior distribution via choice of appropriate covariance functions can be preserved in multi-output case. In the future we plan to compare the multi-output models with different covariance functions and also to apply our model to a broader range of problems.

Acknowledgements: Mykhaylo Andriluka gratefully acknowledges the scholarship provided by the DFG research training group “Cooperative, Adaptive and Responsive Monitoring in Mixed Mode Environments”. The authors are also very thankful to Neil Lawrence for making his IVM code available online.

A Proof of Proposition 1

Since C is positive definite, isotropic on $\mathbb{R}^d$, for any $d \in \mathbb{N}$, we can apply Schoenberg’s theorem [Schoenberg, 1938] which states that there is a finite measure $\mu \geq 0$, such that

$$\mathrm{C}(\tau) = \int_{\mathbb{R}^+} \exp(-\tau^2 s) d\mu(s),$$

and therefore we have

$$\begin{aligned}\mathrm{Cov}((i,\mathbf{x}_a),(j,\mathbf{x}_b)) &= \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} \mathrm{C}(\sqrt{Q_{ij}(\mathbf{x}_a,\mathbf{x}_b)}) \\ &= \int_{\mathbb{R}^+} \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} \exp(-Q_{ij}(\mathbf{x}_a,\mathbf{x}_b) s) d\mu(s).\end{aligned}$$

Let exponential kernel k_i with parameter $v_i \in \mathbb{R}$ and positive-definite matrix A_i be given by

$$k_i(\mathbf{x}) = v_i \exp(-\frac{1}{2}\mathbf{x}^T A_i \mathbf{x}).$$

In [Boyle and Frean, 2005b] it is shown, that

$$\int_{\mathcal{X}} k_i(\mathbf{x}_a - \mathbf{u}) k_j(\mathbf{x}_b - \mathbf{u}) d\mathbf{u} = \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} \exp(-\frac{1}{2} Q_{ij}(\mathbf{x}_a - \mathbf{x}_b)).$$

For exponential kernel given by

$$k_i^s(\mathbf{x}) = (2s)^{d/4} v_i \exp(-\frac{1}{2}\mathbf{x}^T (2sA_i)\mathbf{x})$$

we obtain

$$\int_{\mathcal{X}} k_i^s(\mathbf{x}_a - \mathbf{u}) k_j^s(\mathbf{x}_b - \mathbf{u}) d\mathbf{u} = \frac{v_i v_j (2\pi)^{d/2}}{|A_i + A_j|^{1/2}} \exp(-Q_{ij}(\mathbf{x}_a - \mathbf{x}_b)s).$$

It follows that

$$\mathrm{Cov}((i, \mathbf{x}_a), (j, \mathbf{x}_b)) = \int_{\mathbb{R}^+} \left(\int_{\mathcal{X}} k_i^s(\mathbf{u} - \mathbf{x}_a) k_j^s(\mathbf{u} - \mathbf{x}_b) d\mathbf{u} \right) d\mu(s) \tag{4}$$

Now, consider a finite subset $\mathcal{D} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ of the index set S. We write $\mathrm{T}(n)$ and $\mathbf{x}_n$ for the task and the input of $\mathbf{s}_n$ respectively such that $\mathbf{s}_n = (\mathrm{T}(n), \mathbf{x}_n)$. Let Σ be the $N \times N$ matrix with $\Sigma_{n,m} = \mathrm{Cov}(\mathbf{s}_n, \mathbf{s}_m)$, for $n, m = 1, \ldots, N$. Then, for any $\mathbf{a} \in \mathbb{R}^N$ we have

$$\begin{aligned} \mathbf{a}^T \Sigma \mathbf{a} &= \int_{\mathbb{R}^+} \left(\int_{\mathcal{X}} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m k_{\mathrm{T}(n)}^s(\mathbf{u} - \mathbf{x}_n) k_{\mathrm{T}(m)}^s(\mathbf{u} - \mathbf{x}_m) d\mathbf{u} \right) d\mu(s) \\ &= \int_{\mathbb{R}^+} \left(\int_{\mathcal{X}} \left(\sum_{n=1}^{N} a_n k_{\mathrm{T}(n)}^s(\mathbf{u} - \mathbf{x}_n) \right)^2 d\mathbf{u} \right) d\mu(s) \geq 0 \end{aligned}$$

which completes the proof.

References

[Boyle and Frean, 2005a]P. Boyle and M. Frean. Dependent gaussian processes. In *Advances in Neural Information Processing Systems 17*, pages 217–224, 2005.

[Boyle and Frean, 2005b]Phillip Boyle and Marcus Frean. Multiple output gaussian process regression. Technical report, University of Wellington, 2005.

[Lawrence and Platt, 2004]N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the 21-st International Conference on Machine Learning, Banff, Canada*, 2004.

[Lawrence *et al.*, 2005]N. D. Lawrence, M. Seeger, and R. Herbrich. *The Informative Vector Machine: A Practical Probabilistic Alternative to the Support Vector Machine, Technical Report.* University of Sheffield, 2005.

[Paciorek, 2003]C. Paciorek. *PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A.* 2003.

[Pardo-Igúzquiza *et al.*, 2006]E. Pardo-Igúzquiza, –, and et al. Downscaling cokriging for image sharpening. In *Remote Sensing of Environment*, volume 102, pages 86–98, 2006.

[Rasmussen and Williams, 2006]C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[Schoenberg, 1938]I. Schoenberg. Metric spaces and completely monotone functions. In *Ann. of Math*, volume 39, pages 811–841, 1938.

[Teh *et al.*, 2005]Y. Teh, –, and et al. Semiparametric latent factor models. In *AISTATS*, 2005.

[Yu and Tresp, 2005]K. Yu and V. Tresp. Learning to learn and collaborative filtering. In *NIPS 2005 workshop "Inductive Transfer: 10 Years Later"*, 2005.