Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2008-020 April 11, 2008

Transferring Nonlinear Representations using Gaussian Processes with a Shared Latent Space

Raquel Urtasun, Ariadna Quattoni, Neil Lawrence, and Trevor Darrell

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Transferring Nonlinear Representations using Gaussian Processes with a Shared Latent Space

Raquel Urtasun
UC Berkeley EECS & ICSI
MIT CSAIL

Ariadna Quattoni
MIT CSAIL

Neil Lawrence
University of Manchester

Trevor Darrell
UC Berkeley EECS & ICSI
MIT CSAIL

Abstract

When a series of problems are related, representations derived from learning earlier tasks may be useful in solving later problems. In this paper we propose a novel approach to transfer learning with low-dimensional, non-linear latent spaces. We show how such representations can be jointly learned across multiple tasks in a Gaussian Process framework. When transferred to new tasks with relatively few training examples, learning can be faster and/or more accurate. Experiments on digit recognition and newsgroup classification tasks show significantly improved performance when compared to baseline performance with a representation derived from a semi-supervised learning approach or with a discriminative approach that uses only the target data.

1 Introduction

When faced with a new task, it is advantageous to exploit knowledge and structures found useful in solving related problems. A common paradigm to exploit such knowledge is to learn a feature space from previous tasks and transfer that representation to a future task [2, 5, 17, 1]. Ideally, the transferred representation is of lower dimension than the raw feature space, and the set of functions implied by the new representation still contains the optimal classifier for the new task. When this is the case, the new task can be learned more robustly and/or with fewer training examples in the transferred space than in the raw space.

In this paper we propose a novel approach to transfer learning based on discovering a non-linear low dimensional latent space that is shared accross tasks in a Gaussian process framework, and transferring that space to future tasks.

Transfer of probabilistic representations has been explored in a Gaussian Processes (GP) paradigm, by explicitly sharing a covariance function and/or kernel hyperparameters across tasks [10, 18]. More recently, Bonilla et al. extended previous approaches to model the relateness between tasks with a parametric [3] and non-parametric covariance [4]. However, it is often the case that the relateness is not task-dependent but sample dependent. In other words some samples of task A might be related to task B, while some other samples might not be related at all. Our method estimates the relateness of the different samples by estimating a low dimensional representation that is shared across tasks. Samples that are related are close in latent space.

Probably the closest approach to ours is the one developed by Teh et al. [16], that proposed a semiparametric linearly mixed factor model that models the dependencies by a set of Gaussian Processes. In contrast, our method learns directly a non-linear low dimensional latent space that is

shared accross tasks. When using a Gaussian noise model, our model does not require a variational approximation.

Experiments on digit recognition and newsgroup classification tasks indicate that the ability to transfer non-linear, low-dimensional features across problems can provide significant performance improvements, especially when the target task has relatively few training examples compared to the source problems used to learn the latent space. Baseline experiments confirm that learning the shared latent space discriminatively is important; semi-supervised learning underperformed transfer learning.

In the remainder of the paper, we first describe our approach to learn a discriminative latent space with a single classification task. We then extend this to the multi-task setting, jointly optimizing the latent space to account for each task as well as the underlying data. The learnt latent space is used when learning future tasks. We experiment with different digit recognition and newsgroups classification tasks, and conclude with a discussion of our method and avenues for future work.

2 Probabilistic Discriminative Latent Variable Model

Conventional classification methods suffer when applied to problems with high dimensional input spaces and very small training sets. If, however, the high dimensional data in fact lie on a low-dimensional manifold, accurate classification may be possible with a small amount of training data if that manifold is discovered by the classification method.

More formally, let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$ be the matrix composed of all the observations, with $\mathbf{y}_i \in \Re^D$, and let $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_N]^T$ be the matrix composed of all the labels, with $\mathbf{z}_i \in \Re^S$. Traditional discriminative approaches to classification tried to estimate the mapping from observations to labels. Probabilistic discriminative approaches estimate the probability of the labels given the observations $p(\mathbf{Z}|\mathbf{Y})$.

We are interested in learning a low dimensional representation of the data useful for classification. To have a full Bayesian treatement of the problem, one should marginalize over all the possible latent configurations

$$p(\mathbf{Z}|\mathbf{Y}) = \int_{\mathbf{X}} p(\mathbf{Z}, \mathbf{X}|\mathbf{Y}) d\mathbf{X} \,, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T$ denote the matrix whose rows represent the positions in latent space, $\mathbf{x}_i \in \Re^d$. However, such marginalization is intractable when the relationship between the observations and the labels is non-linear. Instead one could approximate $p(\mathbf{Z}|\mathbf{Y})$ by taking the MAP estimate of $\mathbf{X}$. Similar approximations have been used to learn latent variable models [7].

We place a Gaussian process prior over the observations, such that

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^{D} N(\mathbf{y}_{(j)}|\mathbf{0}, \mathbf{K}_Y) \tag{2}$$

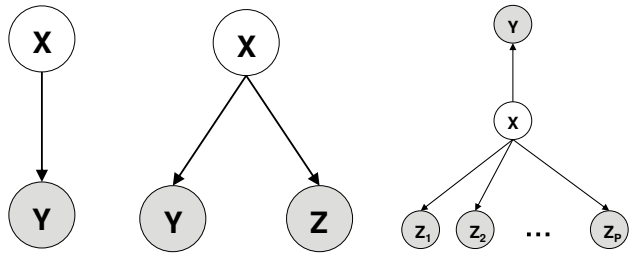
where $\mathbf{y}_{(i)}$ is the i-th column of $\mathbf{Y}$, and the elements of the kernel matrix $\mathbf{K}_Y \in \Re^{N \times N}$ are defined by the covariance function, $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j)$. In particular, we use a kernel that is the sum of an RBF, a bias or constant term, and a noise term,

$$k_Y(\mathbf{x}, \mathbf{x}') \;=\; \beta_1 \exp\left(-\frac{\beta_2}{2}||\mathbf{x} - \mathbf{x}'||^2\right) + \beta_3 \delta_{\mathbf{x},\mathbf{x}'} + \beta_4 \,,$$

where $\bar{\beta} = \{\beta_1, \cdots, \beta_4\}$ comprises the kernel hyperparameters that govern the output variance, the RBF support width, the bias and the variance of the additive noise, respectively.

We introduce an additional intermediate variable $\mathbf{u} \in \Re^S$, such that $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]^T$ and place a Gaussian process prior over $\mathbf{U}$

$$p(\mathbf{U}|\mathbf{X}) = \prod_{j=1}^{S} N(\mathbf{U}_{(j)}|\mathbf{0}, \hat{\mathbf{K}}_Z) \,, \tag{3}$$



(a) (b) (c)

Figure 1: **Graphical models** of (a) GPLVM, (b) our probabilistic discriminative model for a single task, and (c) our Discriminative Transfer model.

where $\mathbf{U}_{(j)}$ is the j-th column of $\mathbf{U}$. We relate this intermediate variable to the observed label using a *noise model*, $p(\mathbf{Z}|\mathbf{U})$. Different noise models can be chosen, typical examples are the Gaussian and probit noise models.

Assuming independence between the observations and labels given the latent variables

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}) = p(\mathbf{Y}|\mathbf{X})p(\mathbf{Z}|\mathbf{X}) \tag{4}$$

we can write

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) \propto p(\mathbf{X})p(\mathbf{Y}|\mathbf{X}) \int_{\mathbf{U}} p(\mathbf{Z}|\mathbf{U})p(\mathbf{U}|\mathbf{X})d\mathbf{U} \, . \tag{5}$$

Fig. 1 (b) depicts the graphical model of our discriminative latent variable model.

With a Gaussian noise model, $p(\mathbf{Z}|\mathbf{U}) = \mathcal{N}(\mathbf{Z}|\mathbf{U}, \sigma^2\mathbf{I})$, the integral in (5) can be done in closed form, and learning the model is equivalent to minimizing the negative log posterior

$$\begin{aligned} \mathcal{L} \;\; = \;\; & \frac{D}{2} \ln |\mathbf{K}_Y| \; + \; \frac{1}{2} \mathrm{tr} \left(\mathbf{K}_Y^{-1} \mathbf{Y}\mathbf{Y}^T \right) \; + \; \frac{S}{2} \ln |\mathbf{K}_Z| \; + \; \frac{1}{2} \mathrm{tr} \left(\mathbf{K}_Z^{-1} \mathbf{Z}\mathbf{Z}^T \right) \\ & + \sum_i \ln \beta_i \; + \; \sum_i \ln \gamma_i \; + \; \sum_i ||\mathbf{x}_i||^2 + C_1 \end{aligned} \tag{6}$$

where C_1 is a constant and $\mathbf{K}_Z = \hat{\mathbf{K}}_Z + \sigma^2\mathbf{I}$. With a Gaussian noise model, the model is similar to the one introduced by [15] to learn a common structure between two observation types. However, the context is very different; we are interested in learning a latent space that can discriminate between classes.

With a non-Gaussian noise model an aproximation to the integral needs to be performed. To approximate the integral required in (6) we used a single pass of the expectation propagation algorithm [12]; this is sometimes known as assumed density filtering (ADF). Updates of site means and parameters of the ADF approximation were interleaved with updates of the latent locations.

Fig. 2 shows an example where a two-class problem is learned using (a) PCA, (b) GPLVM, and our probabilistic discriminative model with (c) Gaussian and (d) probit noise models. Jointly learning the reconstruction and classification mappings separates the different classes in the latent space and results in better classification performance.

The Gaussian noise model leads to a greater separation of the data in latent space. This is perhaps to be expected, as the constraints imposed by the labels through the Gaussian noise model are stronger than those imposed by a probit noise model. The probit noise model simply forces the respective classes to fall either side of the decision boundary, whereas the Gaussian noise model encodes extra information about how far each class should be apart on average. Fig. 3 shows mean classification errors for the USPS database as a function of the number of examples when using the Gaussian and probit noise models for 2–D and 3–D latent spaces. Note that the Gaussian noise model in general outperforms the probit.

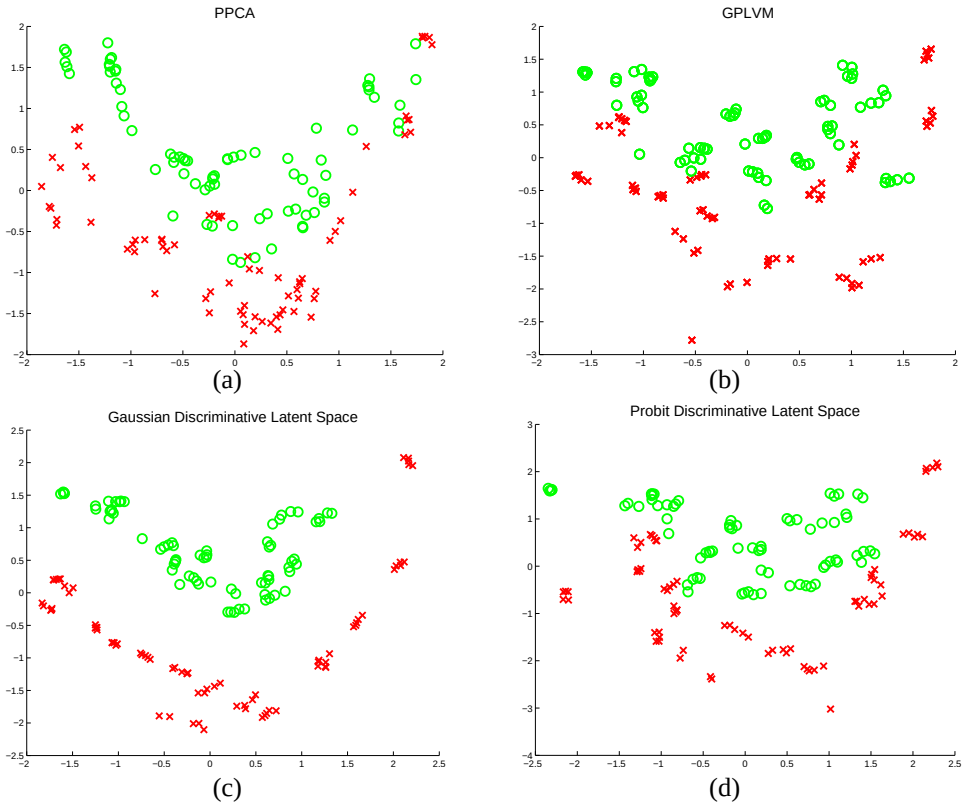


Figure 2: **Probabilistic discriminative model**. Low dimensional representations learned using (a) PCA, (b) GPLVM, (c) our probabilistic discriminative model (Section 2) with a Gaussian noise model, and (d) our probabilistic discriminative model with a probit noise model. The data is composed of two different classes. The training examples of the two classes are depicted as red crosses and green circles. Note how the discriminative model separates the classes, specially when using a Gaussian noise model.

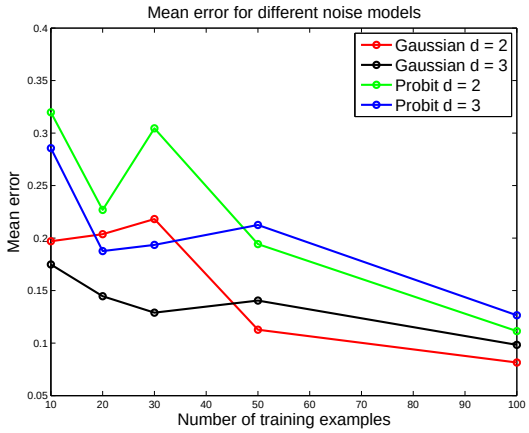


Figure 3: **Noise models**: Mean error for a discriminative single-task model using Gaussian and probit noise models and 2–D and 3–D latent spaces. The task consists in discriminating 3's vs 5's in the USPS database. For the same latent space dimensionality, the Gaussian noise model performs in general better than the probit noise model. The results were averaged over 5 trials.

3 Transfer Learning with a Shared Latent Space

In our transfer learning scenario we are interested in learning a low dimensional representation from a set of related problems. We assume that if a latent space was useful for a set of tasks, it will be useful for future related tasks.

One of the advantages of using a low dimensional representation of the data for classification is that fewer parameters need to be estimated, thus reducing the number of examples required for training. Therefore, if we can find a useful low dimensional representation using previous tasks, we can effectively reduce the number of examples needed for learning the new task. More importantly, the latent space can discover how related the different samples are: points that are close in latent space are related.

In this section we show how the discriminative latent variable model described above can be extended to the transfer learning scenario. In particular, we adopt an asymmetric approach that first learns a latent space from a set of related problems, and then uses that low dimensional space to perform classification for the target task.

3.1 Jointly learning P related problems

In this section we describe how to learn P related tasks with a discriminative shared representation. Let $\mathbf{Y}^{(p)} = [\mathbf{y}_1^{(p)}, \cdots, \mathbf{y}_{N_p}^{(p)}]^T$ be the N_p observations associated to problem p. Similarly let $\mathbf{X}^{(p)} = [\mathbf{x}_1^{(p)}, \cdots, \mathbf{x}_N^{(p)}]^T$, be the latent coordinates for problem p, and $\mathbf{Z}^{(p)} = [\mathbf{z}_1^{(p)}, \cdots, \mathbf{z}_{N_p}^{(p)}]^T$ be the labels for the p-th problem.

Since we want to discover the relateness of the samples from different tasks, we place a Gaussian process prior over all observations from all related problems such that

$$p(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) = \frac{1}{T_1} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\bar{\mathbf{Y}}\bar{\mathbf{Y}}^T\right)\right), \tag{7}$$

where T_1 is a normalization factor, $\bar{\mathbf{Y}} = [\mathbf{Y}^{(1)}, \cdots, \mathbf{Y}^{(P)}]^T$ is the set of all observations, $\bar{\mathbf{X}} = [\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(P)}]^T$ are the latent coordinates for all problems, and $\mathbf{K}_Y$ is the covariance matrix formed by elements of all the problems $k(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(k)})$.

Note that in (7) all tasks can be related (i.e., the covariance matrix is not block diagonal), and that the proximity in the latent space captures the relateness at the sample level, i.e., if two observations of different tasks are related, their corresponding latent coordinates will be close, and their covariance $k(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(k)})$ will be large.

As the mapping from latent space to labels is different for each problem, we assume independence of the labels of the different problems given the latent variables,

$$p(\bar{\mathbf{Y}}, \bar{\mathbf{Z}}|\bar{\mathbf{X}}) = p(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) \prod_{i=1}^{P} p(\mathbf{Z}^{(p)}|\mathbf{X}^{(p)}), \tag{8}$$

where $\bar{\mathbf{Z}}$ are the labels for all problems.

Note that our model (Fig. 1 (c)) is a generative model of the data with a shared latent space across problems, and a set of independent (given the latent variables) mappings modeling each classification task

$$p(\mathbf{Z}^{(p)}|\bar{\mathbf{X}}^{(p)}) = \int_{\mathbf{U}^{(p)}} p(\mathbf{Z}^{(p)}|\mathbf{U}) p(\mathbf{U}^{(p)}|\mathbf{X}^{(p)}) d\mathbf{U}^{(p)} \ . \tag{9}$$

As before, we place a Gaussian process prior on $p(\mathbf{U}^{(p)}|\mathbf{X}^{(p)})$. Different noise models can be chosen. In particular, when using a Gaussian noise model, the integration in (9) can be done in closed form. Learning the model is then done by minimizing the negative log posterior that is

$$\begin{aligned} \mathcal{L}_{TL} \ = \ & \frac{D}{2}\ln|\mathbf{K}_Y| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{Y}^T\right) + \sum_{i=1}^{P}\left(\frac{S}{2}\ln|\mathbf{K}_Z| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_Z^{(p)^{-1}}\mathbf{Z}\mathbf{Z}^{(p)^T}\right)\right) \\ & + \sum_i \ln\beta_i + \sum_{p=1}^{P}\sum_i \ln\gamma_i^{(p)} + \sum_i ||\mathbf{x}_i||^2 + C_2 \end{aligned} \tag{10}$$

where C_2 is a constant, $\mathbf{K}_Z^{(p)} = \hat{\mathbf{K}}_Z^{(p)} + \sigma_p^2\mathbf{I}$, and $\hat{\gamma}^{(p)}$ are the hyperparameters associated with problem p. Note that now, there are $P * N_p + N_Y$ hyperparameters to estimate, where N_p is the number of hyperparameters to model a single latent space to labels mapping, and N_Y is the number of hyperparameters of the mapping from the latent space to the observations.

3.2 Transfer learning with the joint model

Now that we can find a low dimensional representation that is useful for multiple tasks, we turn our attention to the problem of using that representation in a future task.

Let $\mathbf{Y}^{(target)}$ be the observations for the target problem. Given the latent representation learned from p related problems, our transfer learning algorithm proceeds as follows. First the target latent positions $X^{(target)}$ are infered, and then the mapping from latent space to labels is learned using a Gaussian Process.

Infering the latent positions for the target observations involves maximizing $p(\mathbf{Y}^{(target)}|X^{(target)}, \bar{\mathbf{X}}, \bar{\mathbf{Y}})$. This is computationally expensive. To speed up this process we incorporate back-constraints [11] to learn the inverse mapping (i.e., mapping from observations to latent space) as a parametric function $x_{ij} = g_j(\mathbf{y}_i; \mathbf{a})$. In particular we use an MLP to model the relation between $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. Given the target observations, their latent coordiantes are computed simply by evaluating the parametric mapping $x_{ij}^{(target)} = g_j(\mathbf{y}_i^{(target)}; \mathbf{a})$, where $\mathbf{a}$ is learned from the related problems.

An alternative approach to our transfer learning method consists of jointly learning the target and related problems. However, in practice when the target problems have few examples compared to the related problems, the results are very similar. The advantage of the two step optimization is that if the latent space has already been learned (from a potentially very large collection of related problems) training the target tasks becomes very fast.

4 Experimental Results

We conducted experiments on two datasets, the USPS dataset from the UCI repository, and a news-group dataset [6]. We compare our transfer learning approach (Section 3) to two baselines. The first baseline ignores the related problems and learns a discriminative single-task model (Section 2) using only data from the target problem.

The second baseline uses only the observations from the related problems but not their labels. More specifically, it learns a PCA space using data from the related problems and projects the target observations onto that space. It then learns the mapping from the projected target samples to the target labels with a Gaussian process.

For all experiments we used a two dimensional latent space and a Gaussian noise model motivated by the results shown in Section 2 (Fig. 3). All optimizations were performed with conjugate gradient descent and run for a maximum of 100 iterations.

4.1 USPS dataset

We conducted experiments on the first five digits of the USPS dataset; we regard the binary detection of each digit as a separate task. Each of the tasks consists of detecting one digit from the other digits[1]. For every task we generate a labeled training set by randomly picking 300 positive examples and 300 negative examples (sampled from the other four digits). We generate a testing set for each task in a similar manner.

In the first set of experiments we transfer a representation learned from 600 training examples (300 positive and 300 negative) from a single other problem. We evaluate the performance of our algorithm and the baselines for different training set sizes of the target problem using 10 random partitions of the data. As depicted by Fig. 4, in all cases the discriminative single problem model (Section 2) gives lower error rates than the PCA-based semi-supervised baseline, illustrating the importance of learning the latent space discriminatively. Transferring from any task except from digit 5 increases performance relative to the baseline, suggesting that digit 5 might not be as related. The *self-transfer* case gives an upperbound on the performance of transfer learning; it tells us what the performance would be if the problems were *fully* related (Fig. 4 (c)).

[1] In particular we focus on detecting 3's from the other digits (i.e., 1's, 2's, 4's, 5's) since it is known to be one of the most difficult problems in this dataset.

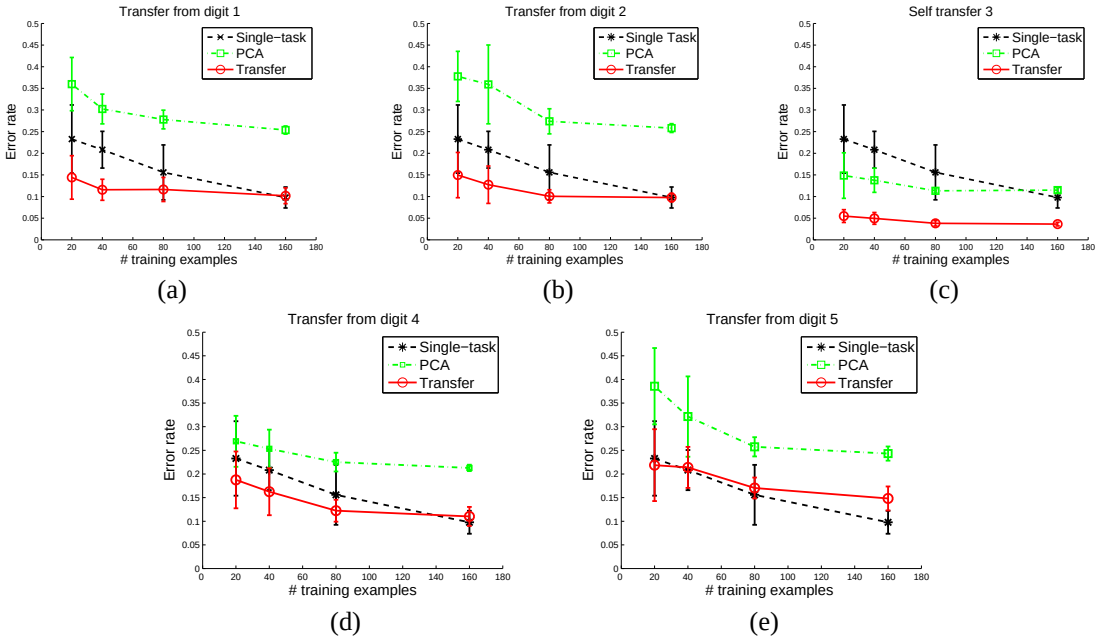


Figure 4: **Transfering the latent space** learned from a single source problem to a target problem. The target problem is to discriminate 3's from other digits, and the source problems are to similarly detect (a) 1's, (b) 2's, (c) 3's, (d) 4's, and (e) 5's. Each related problem is trained with 300 positive examples and 300 negative examples of the other four digits. Our algorithm (red) is compared against two baselines: a single-task probabilistic discriminative model that learns a latent space only from the target data (black) and the result of PCA-based semi-supervised learning with unlabeled data from the source problem (green). Note that the 5's problem might be less related to the 3's since the transfer learning does not help. In all cases except self-transfer (c), the PCA baseline underperforms the discriminative probabilistic model, showing the importance of discriminatively learning the latent space.

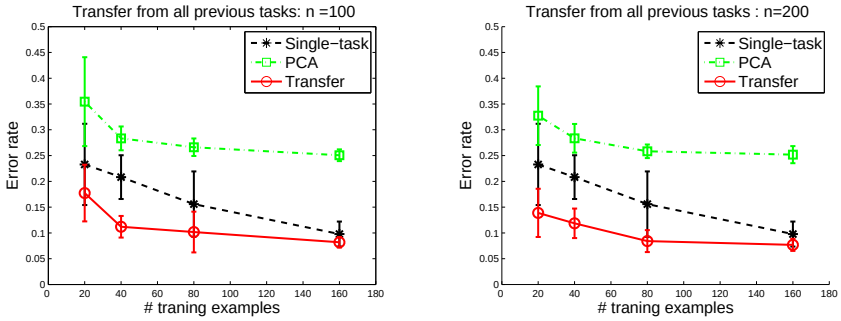
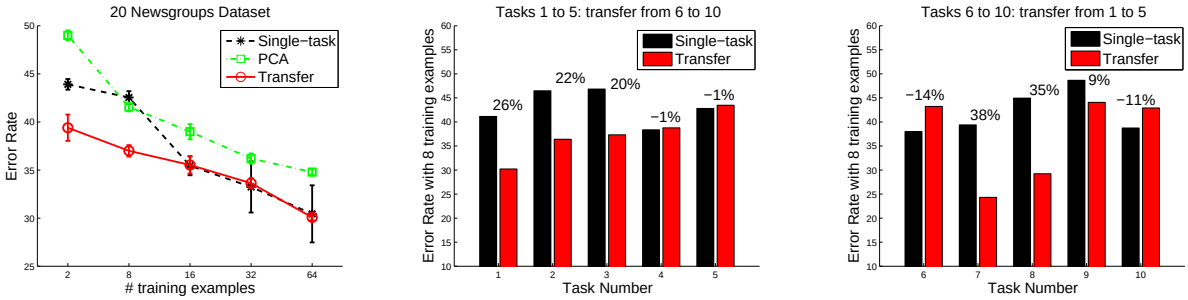


Figure 5: **Joint learning** of the latent space from multiple source problems and transfer to the target task. The source and target problems are as above. Results using Discriminative Transfer are shown in red and are compared against two baselines, PCA-based semi-supervised learning (green) and a single-task probabilistic discriminative model trained on the target problem (black). Transfer from other related (and less related) problems improves the performance with respect to learning with a single-task model, especially when the number of examples is small. PCA-based semi-supervised learning performs poorly in this case. Figure (a) shows results when using 100 positive examples and 100 negative examples for each related problem, and (b) shows results with 200 positive and 200 negative examples.

The previous experiment showed that transferring a shared latent space from a related problem can significantly reduce the number of training examples needed to learn the target problem. However, in practice we do not know what problem to choose for transfer because we do not know a priori



(a) (b) (c)

Figure 6: **NewsGroups dataset:** (a) Mean error rates over the 10 newsgroup tasks. (b-c) Mean error rates over the 10 newsgroup tasks when trained with 8 samples. Task 1= Motorcycles vs MS-Windows, Task 2= Baseball vs Politics.misc, Task 3= Religion vs Politics.guns Task 4= Atheism vs Autos, Task 5= IBM.hardware vs Forsale, Task 6= Politics.middleeast vs Sci.med, Task 7= Christian vs Hockey, Task 8= Space vs MAC.hardware, Task 9= Windows.x vs Electronics, Task 10= Sci.crypt vs Comp.graphics

which problems are related. What we need is a transfer learning algorithm that takes a set containing mostly related problems and learns good shared latent spaces without being adversely affected by the presence of a few less-related problems. In our second set of experiments we test the robustness of our transfer learning algorithm in this more realistic scenario and transfer a shared representation from all previous problems (i.e., detecting 1's, 2's, 4's, 5's). The results in Fig. 5 show that our transfer learning approach improves performance significantly compared to the baselines. Our algorithm performs similarly using 200 (Fig. 5(a)) or 400 (Fig. 5(b)) examples for each related problem.

4.2 20 Newsgroups Dataset

We used a standard newsgroup classification dataset [6] that contains postings from 20 different newsgroups[2]. Following [14], 10 binary classification tasks were defined by randomly pairing classes (e.g., a task consists of predicting if a given document belongs to the Motorcycles or MS-Windows newsgroup). Each document was represented using a bag of words. A vocabulary of approximately 1700 words was created by taking the union of the 250 most frequent words for each of the 20 newsgroups (after stemming and removing stop words). Note that this setting is different from the one in [14], where every binary task used a vocabulary formed by picking the 250 most frequent words for each of the two topics in the discrimination task.

To construct a transfer learning setting we divide the tasks into two groups, the first contained tasks 1 to 5 and the second contained tasks 6 to 10. We jointly learned each group, and use the problems from the other group as target tasks. 100 documents were used for each related problem. We report results averaged over 10 partitions of the data.

Fig. 6 (a) shows average test error results over the 10 tasks for the discriminative single-task model, PCA and our transfer learning approach. For small training set sizes (i.e. less than 16 examples) using transfer learning produces significantly lower average error than both baselines.

Fig. 6 (b-c) shows test error results for each task when trained with 8 examples. For 6 out of the 10 tasks we observe positive transfer; transfer learning reduced the error by at least 9% and at most 38 %. The tasks that exhibit most positive transfer are: Christian vs Hockey, and Space vs MAC.hardware. For two tasks there is no significant difference between transfer learning and the baseline. Finally, for the tasks Politics.middleeast vs Sci.med and Sci.crypt vs Comp.graphics we observe negative transfer.

[2]Atheism, Autos, Baseball, Motorcycles, MS-Windows,Christian, Comp.graphics, Electronics,Forsale, Hockey, IBM.hardware, MAC.hardware, Politics.guns, Politics.middleeast, Politics.misc, Religion, Sci.crypt, Sci.med, Space, Windows.x

5 Conclusion and Discussion

We have presented a new method for transfer learning based on shared non-linear latent spaces that estimates task relateness in a per-sample basis. Our method performs joint optimization within a Gaussian Process framework, and discovers latent spaces which are simultaneously effective at describing the observed data and solving several classification tasks. When transferred to new tasks with relatively few training examples, learning can be faster and/or more accurate with this approach. Experiments on digit recognition and newsgroup classification tasks demonstrated significantly improved performance when compared to baseline performance with a representation derived from a semi-supervised learning approach or with a discriminative approach that uses only the target data.

We now turn to discuss possible extensions of the algorithms proposed in this paper.

Semi-supervised learning: The discriminative latent space model of section 2, and the joint model of section 3 can be extended to the semi-supervised case by simply constructing the covariance matrix $\mathbf{K}_Y$ over the label and unlabeled data, while defining $\mathbf{K}_Z$ only over the labeled examples.

Making the algorithm efficient: The main computational burden of our approach is the computation of the inverse of $\mathbf{K}_Y$, its computational cost being of the order of $\mathcal{O}(N^3)$, where N is in the number of training examples in all the problems. The latest Gaussian proceses sparsification techniques [13] can be applied to both algorithms presented in this paper in the same way as they were applied to the Gaussian Process Latent Variable Model [8]. Of particular interest is their application to the mapping from latent space to observations since it is the most expensive computationally. The complexity of our algorithms can then be reduced to $\mathcal{O}(NM^2)$, where M is the number of inducing variables.

Noise models: Different noise models can be exploited in our framework, resulting in different behaviours. In this paper we have shown results using a Gaussian and a probit noise model. In the experiments we performed, the Gaussian noise model outperformed the probit, learning a more discriminative latent space. Other noise models could also be exploited. Of particular interest to us is the *Null Category Noise Model* (NCNM) [9] for semi-supervised learning that explicity attempts to separate the different classes. A rigourous comparison of different noise models is our subject of future research.

References

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:817–8535, 2005.

[2] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28, 1997.

[3] E. V. Bonilla, F. V. Agakov, and C. K. I. Williams. Kernel Multi-task Learning using Task-specific Features. 2007.

[4] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task gaussian process prediction. In *Neural Information Processing Systems*. MIT Press, 2007.

[5] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[6] K. Lang. Newsweeder: learning to filter netnews. In *International Conference in Machine Learning*, 1995.

[7] N. D. Lawrence. Gaussian Process Models for Visualisation of High Dimensional Data. In *Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2004.

[8] N. D. Lawrence. Learning for larger datasets with the gaussian process latent variable model, 2007. AISTATS.

[9] N. D. Lawrence and M. I. Jordan. Gaussian processes and the null-category noise model. In *Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2005.

[10] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *International Conference in Machine Learning*, 2004.

[11] N. D. Lawrence and J. Quiñonero-Candela. Local distance preservation in the GP-LVM through back constraints. In *International Conference in Machine Learning*, volume 69, pages 96–103, Banff, Alberta, Canada, July 2006.

[12] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT, 2001.

[13] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, December 2005.

[14] R. Raina, A. Y. Ng, and D. Koller. Constructing Informative Priors using Transfer Learning. In *International Conference in Machine Learning*, 2006.

[15] A. P. Shon, K. Grochow, A. Hertzmann, and R. Rao. Learning Shared Latent Structure for Image Synthesis and Robotic Imitation. In *Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2006.

[16] Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. 2005.

[17] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Neural Information Processing Systems*, 1996.

[18] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *International Conference in Machine Learning*, pages 1012–1019, 2005.