Helsinki University of Technology
Publications in Computer and Information Science

Report E11
May 2007

LEARNING FROM RELEVANT TASKS ONLY

Samuel Kaski Jaakko Peltonen

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:
Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science
P.O. Box 5400
FI-02015 TKK, Finland
Tel. +358-9-451 3267
Fax +358-9-451 3277

This report is downloadable at
http://www.cis.hut.fi/Publications/

ISBN 978-951-22-8807-6
ISSN 1796-2803

Learning from Relevant Tasks Only

Samuel Kaski and Jaakko Peltonen**

Laboratory of Computer and Information Science, Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK, Finland
{samuel.kaski,jaakko.peltonen}@tkk.fi

Abstract. We introduce a problem called relevant subtask learning, a variant of multi-task learning. The goal is to build a classifier for a task-of-interest having too little data. We additionally have data for other tasks but only some are relevant, meaning that they contain samples classified in the same way as in the task-of-interest. The problem is how to utilize this "background data" to improve the classifier in the task-of-interest. We show how to solve the problem for logistic regression classifiers, and demonstrate that the solution works better than a comparable multi-task learning model. The key is to assume that data of all tasks are mixtures of relevant and irrelevant samples, and model the irrelevant part with a sufficiently flexible model such that it does not distort the model of relevant data.

Key words: multi-task learning, partially relevant data, relevant subtask learning, transfer learning

1 Introduction

All too often in classification tasks there is too little training data to estimate sufficiently powerful models. This problem is ubiquitous in bioinformatics, where the dimensionality of data measured with "high-throughput" techniques such as gene expression microarrays is large; this is known as the large p, small n problem, where p is the dimensionality and n is the number of data. The same problem appears also in image classification from few available examples, finding of relevant texts, etc. The possible solutions are to restrict the complexity of the classifier by incorporating prior knowledge, or to measure or find more data. Unfortunately prior knowledge may not exist or it may be insufficient, measuring new data may be too expensive, and there may not exist more samples of *representative* data. Most classifiers assume that the learning data is representative in the sense that it all comes from the same distribution as the test data.

There often is *partially representative* data available; for instance in bioinformatics there are databases full of data measured for different tasks or in different conditions or contexts, and for text models there is the web full of texts. Assuming that some of the background data sets are relevant, they can each be considered training data from a (partially) different distribution as the test data.

** The authors contributed equally to the work.

The learning problem can be formalized as follows: Assume several data sets of which one is special in the sense that it is known to come from the same distribution as future test data. The other data sets come partly from that same distribution, in the sense that in each set, some portion of the samples (0-100%) has been classified according to the same criteria as the "proper" training data. The rest of the samples come from some other distribution, which is typically different for each set. The question now is, *can we use the partially relevant data sets to build a better classifier for the test data?*

This learning problem is special type of multi-task learning. If learning a classifier for one data set is called a task, then the research problem of multi-task learning [1] is to find out whether learning all tasks together helps in learning each. The multi-task models have mainly been symmetrical, and transfer to new tasks is currently done by using the posterior from the other tasks as a prior (e.g. [2, 3]). We take a more direct approach to focusing on the task-of-interest, which results in different kinds of models which are better suited in the new task while being of the same order of complexity as the earlier multi-task learning models. Our learning problem is fundamentally asymmetric and more structured; the test data fits one of the tasks, the "*task of interest*," and some of the other tasks contain *subtasks* that are relevant for the task of interest, but no other task needs to be completely relevant.

More generally, this problem can be seen as a combination of information retrieval and supervised learning: the data from the task of interest can be seen as a query, which is used to retrieve relevant other tasks (or relevant parts of other tasks), and the found relevant data are then used to supplement the scarce original data while learning a classifier. Everything is done with a generative model, which allows rigorous treatments of small data sets.

2 Previous Work

The problem is partly related to several other learning problems: transfer learning, multi-task learning, and semisupervised learning.

Many variants of multi-task learning have been proposed. A common approach is to build a hierarchical (Bayesian) model of all tasks together. Information sharing between tasks can then be done by using constrained prior distributions that favor similarity of parameters between tasks. The tasks may be learned either together (e.g. [4–6]) or previous tasks can be used to learn a prior for a new task (e.g. [2, 3]). Both approaches are based on modeling all tasks symmetrically with a hierarchical (Bayesian) model. Symmetric hierarchical modeling has also been used in support vector machine learning (e.g. [7]). By contrast, we study an asymmetric situation where there is a specific task-of-interest and only some tasks, or parts thereof, are relevant for it.

Some multi-task learning-based solutions exist where all tasks are not considered relevant for all others. In [8] the tasks are assumed to come from clusters, such that tasks in the same cluster are generated with the same parameters. The model learns to cluster tasks in order to share data between the tasks in each

cluster. A similar approach based on clustering or gating of tasks is used in [9], and based on support vector machines in [7]. However, in all these approaches all tasks are still equally important with respect to the clustering so there is no specific task-of-interest.

There are some interesting but partly heuristic transfer learning approaches where a single task-of-interest does exist. In [10] two strategies are used: a single global weighting parameter is used to control the weight of auxiliary samples in nonparametric classification, or background data are used as support vectors or constraints in support vector machines. In [11] a flexible model is used where extra variables are used in logistic regression to artificially improve the log-likelihood of undesirable samples of auxiliary data, and a constraint on the use of the extra variables forces the model to seek useful auxiliary samples.

Semisupervised learning [12] is related to the learning problem in this paper as well: Whether a sample is relevant for the task-of-interest or not can be considered a binary-valued auxiliary label which is known for the "proper" training data but unknown for the background data. In this sense, the learning problem is semisupervised with regard to the auxiliary label. We emphasize, however, that predicting relevance of background samples is only part of our learning problem; the true goal is to predict classes within the task-of-interest.

3 Relevant Subtask Learning

3.1 The Setting

Consider a set of classification tasks indexed by $S = 1, ..., M$. For each task S we have a training data set $D_S = \{\mathbf{x}_i, c_i\}_{i=1}^{N_S}$ where the $\mathbf{x}_i \in \mathbb{R}^d$ are d-dimensional input features, the c_i are the class labels, and N_S is the number of training samples for that task. In this paper we assume for simplicity that d is equal in all tasks and all tasks are two-class classification tasks where c_i is either +1 or -1. The process that generates the classes is, however, different in each task. One task, with index U, is the *task-of-interest*, and the rest are *supplementary* tasks.

Some portion (0-100%) of the samples of each supplementary task are assumed to come from the same disribution as the task-of-interest. The rest come from some other distribution which may be different for the different supplementary tasks.

We wish to learn to predict classes well for data that come from the task-of-interest. We are not interested in the other tasks except as a source of information for the task-of-interest. Note that there are no paired samples between the tasks; thus, the only connections between the tasks are the possible similarities in their underlying distributions, which we do not know beforehand.

3.2 The Principle

The relevant subtask learning problem is to build a classifier, or more specifically a model for the class density $p(c|\mathbf{x}, U)$ in task U, because test data is known to

come from this distribution. In addition to data $D_U = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_U}$ of task U, there are data D_S from other tasks S available. The assumption is that some samples of each D_S may come from the distribution $p(c|\mathbf{x}, U)$ but the rest do not.

To model the data, a model family (family of classifiers) must of course be chosen for the task-of-interest by the analyst; the choice is done as usual based on prior knowledge, or resorting to a nonparametric or semiparametric model. Particular models in the family are denoted by $p(c|\mathbf{x}, U; \mathbf{w}_U)$, where $\mathbf{w}_U$ are the parameters whose values identify the model. This is standard practice; the interesting question is how to model the relationships between the task-of-interest and the other tasks, which we will discuss next.

For each of the supplementary tasks S, it is assumed that part of the data samples come from the same distribution $p(c|\mathbf{x}, U; \mathbf{w}_U)$, but part from another distribution. The former are relevant for modeling the task-of-interest whereas the latter are not, and hence the supplementary data set is *partially relevant.* The analyst needs to specify a model for the non-relevant samples as well. This can again be done based on prior information but typically a nonparametric or semiparametric model would be used to avoid the excessive manual work of collecting prior infomation about all tasks. Denote the model for the non-relevant samples of subtask S by $p_{\text{nonrelevant}}(c|\mathbf{x}, S; \mathbf{w}_S)$.

Since we assume the task S to be a mixture of relevant and nonrelevant samples, the model for it should be

$$p(c|\mathbf{x}, S; \boldsymbol{\theta}) = (1 - \pi_S)p(c|\mathbf{x}, U; \mathbf{w}_U) + \pi_S p_{\text{nonrelevant}}(c|\mathbf{x}, S; \mathbf{w}_S) \ , \qquad (1)$$

where $\pi_S \in [0, 1]$ is a parameter modeling the proportion of irrelevant samples in task S and $\boldsymbol{\theta}$ denotes all parameters of all tasks. Note that this model reduces to $p(c|\mathbf{x}, U; \mathbf{w}_U)$ for the task-of-interest (where $\pi_S = 0$).

The solution is to use (1) to model the data. The idea behind the functional form is that a flexible enough model for $p_{\text{nonrelevant}}$ "explains away" irrelevant data in the auxiliary subtasks, and hence $p(c|\mathbf{x}, U; \mathbf{w}_U)$ learns only on the relevant data. In other words, by forcing one of the (here two) subtasks to use the same parameters in all tasks, we force the model to find from the other tasks the common part that is useful for the task of interest. We call this method *Relevant Subtask Model* (RSM).

The tradeoff here is that to improve performance on the task-of-interest, we need to spend considerable computational time to model data of all the supplementary data sets as well. This is sensible assuming the bottleneck is the amount of learning data in the task-of-interest.

3.3 Case Study: Logistic Regression

In this paper we introduce our solution with a simple parametric model, but we stress that the method can easily be generalized to more general parametric or semiparametric models.

We will model the task-of-interest U with a logistic regression model,

$$p(c|\mathbf{x}, U; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-c\mathbf{w}_U^T \mathbf{x})} \,. \tag{2}$$

For notational simplicity the bias term has been included in the weight vector (here $\mathbf{w}_U$), which is well known to result in standard logistic regression when an extra dimension with constant value 1 is added to all input vectors $\mathbf{x}$.

We model the non-relevant data in the other tasks with logistic regression models as well. Each supplementary task S has a different regression model, having its own parameters:

$$p_{\text{nonrelevant}}(c|\mathbf{x}, S; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-c\mathbf{w}_S^T \mathbf{x})} \tag{3}$$

where $\mathbf{w}_S$ is the weight vector. Hence the supplementary tasks are each generated from a mixture of two logistic regression models:

$$p(c|\mathbf{x}, S; \boldsymbol{\theta}) = \frac{1 - \pi_S}{1 + \exp(-c\mathbf{w}_U^T \mathbf{x})} + \frac{\pi_S}{1 + \exp(-c\mathbf{w}_S^T \mathbf{x})} \tag{4}$$

where π_S is the mixture weight.

3.4 Optimization

This paper is intended to present a proof of concept that the model (4) works as expected. We will use simple methods, maximum conditional likelihood estimation and standard conjugate gradients, and spend effort to designing controlled experiments. More advanced methods will be added in later papers.

Since the task is classification, or more specifically modeling of the distribution of classes given data, the objective function is the conditional log-likelihood

$$L_{\text{RSM}} = \sum_S \sum_{i \in D_S} \log p(c_i|\mathbf{x}_i, S; \boldsymbol{\theta}) \tag{5}$$

where S goes over all tasks including the task-of-interest, and $p(c_i|\mathbf{x}_i, S; \boldsymbol{\theta})$ is given in (4).

To optimize RSM, we use standard conjugate gradient optimization to maximize the objective function (5) with respect to the parameters ($\mathbf{w}_U$, the $\mathbf{w}_S$, and the π_S).

4 Comparison Methods

We compare the new method RSM to three alternative standard approaches, which assume progressively stronger relationships between the tasks. The set of comparisons is intended as a proof-of-concept. The models and optimization strategies are relatively simple but comparable across the approaches. More advanced versions will be compared in later work.

All methods are optimized by maximizing the (conditional) likelihood with a standard conjugate gradient algorithm.

4.1 Single-Task Learning

The obvious way to learn a given task is to only use the learning data of that task, and not to try to benefit from the other tasks at all. This "single-task learning" strategy is expected to work well if there is a lot of data, otherwise it will overfit. Alternatively, single-task learning is wise if the other available tasks are known to be very different.

The RSM model naturally reduces to single-task learning if no other tasks are available. We simply used a single logistic regression model (2) for the single-task learning.

4.2 Multi-task Learning by Task Clustering

Currently one of the most promising multi-task learning strategies is to assume the tasks to come from task clusters, such that parameters of the tasks are shared within each cluster [8]. For this proof-of-concept study we simplify the state-of-the-art full-Bayesian approaches to a maximum likelihood -estimated clustering that is comparable to the other methods. This does not reduce generality since all approaches can in principle be given a full-Bayesian treatment.

Assume that there is a fixed number K of underlying task clusters. To keep model complexity comparable to RSM, each task cluster k consists of a mixture of two logistic regression models:

$$p(c|\mathbf{x}, k; \boldsymbol{\theta}) = \frac{\pi_k}{1 + \exp(-c\mathbf{w}_{k,1}^T \mathbf{x})} + \frac{1 - \pi_k}{1 + \exp(-c\mathbf{w}_{k,2}^T \mathbf{x})} \tag{6}$$

where the weight vectors $\mathbf{w}_{k,1}$, and $\mathbf{w}_{k,2}$ and the mixing weight π_k are the parameters of cluster k. Each task is fully generated by one of the K clusters, but it is unknown which cluster. The conditional class probability of task S is therefore

$$p(\{c_i\}_{i \in D_S} | \{\mathbf{x}_i\}_{i \in D_S}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \gamma_{k|S} \prod_{i \in D_S} p(c_i|\mathbf{x}_i, k; \boldsymbol{\theta}) \tag{7}$$

where the $\{c_i\}_{i \in D_S}$ and $\{\mathbf{x}_i\}_{i \in D_S}$ are the set of class labels and the corresponding set of input vectors from task S, and the parameter $\gamma_{k|S}$ is the probability that task S comes from cluster k.

The parameters (weights, mixing weights, and cluster probabilities for each task) are optimized by maximizing the conditional class likelihood

$$L_{\mathrm{TCM}} = \sum_S \log p(\{c_i\}_{i \in D_S} | \{\mathbf{x}_i\}_{i \in D_S}, \boldsymbol{\theta}) \; . \tag{8}$$

We call this model "Task Clustering Model" (TCM). It is meant to be a maximum likelihood version of [8], but having a more complex model per clusters (mixture of two instead of one logistic regression model).

4.3 Learning from All Data

The "extreme" multi-task alternative is to assume all data from all tasks to be useful, and to learn from them all as if they all came from the task-of-interest. We call this strategy "all together;" it may work well if all tasks are very similar, otherwise it will lose the distinctive features of the task-of-interest in the mixture of all tasks.

This is essentially the TCM model having a single cluster.

5 Experiments

We studied the performance of the models in three different experimental settings. The models RSM and TCM make different assumptions; we start from a task domain where the assumptions of both methods hold, and study with artificial data sets how the methods tolerate progressive deviations from their assumptions.

When the assumption that tasks come from clusters is violated, the performance of TCM degrades. The new RSM however, seems to be even surprisingly tolerant to deviations from the assumption of a shared subtask, if there is enough data in the task-of-interest to determine relevance of other tasks.

In the last experiment we use a more realistic scenario of classifying test documents from the Reuters-21578 collection according to the interest of one user, when classifications from other users with different interest profiles are available.

We will use the following terminology: A multi-task learning problem consists of multiple tasks, each having its own learning data coming from a certain distribution. The multi-task problem comes from a problem *domain* which specifies the distribution of the data in each task, and the relationships of the tasks.

5.1 Experiment 1: When Task Clustering Fails

In this experiment we created a continuum of multi-task problem domains where the relationship between the task-of-interest and the other tasks changes. The continuum was set up so that the tasks always follow the assumptions of RSM but the assumption of underlying task clusters in TCM starts to fail. The setting is explained in a schematic diagram in Figure 1 (left).

Technically, we constructed 10 problem domains each having different characteristics, and generated 40 learning problems from each domain. Each learning problem consisted of 10 tasks; the number of samples in each supplementary task was uniformly sampled from the interval 250–500, whereas for the task-of-interest the number of samples was three times smaller (i.e., uniform in the interval 83–167). In each task, the input features $\mathbf{x}_i$ were samples from a 5-dimensional Gaussian, and the labels c_i were sampled from a mixture of two logistic regression models. The weight vectors of those models were chosen differently in each problem domain, so that the domains form a continuum progressively worse for

TCM; Figure 1 (left) shows a conceptual illustration of the setup. After picking the classes, we lastly added a small amount of extra Gaussian noise to the features.

Figure 1 (right) shows the results as a function of the progressively changing domain, averaged over the 40 learning problems. RSM attains very high performance in all domains, close to the upper limit computed by using the parameters with which the data was generated. (The bound is only approximate because noise has been added to the input vectors afterwards.) The performance of TCM decreases as the tasks become less clustered, as expected. The number of clusters in TCM was set to the correct value used when generating the data, to give some advantage to the comparison method.

Both of the naive methods perform poorly. For "all together" the reason is that only some background data are relevant for the task-of-interest; placing all data into one data set introduces noise as well as useful information. For single-task learning the very poor performance and large variance are due to overfitting: generalization performance depends on whether the small "proper" training data happened to look like the underlying distribution or not.

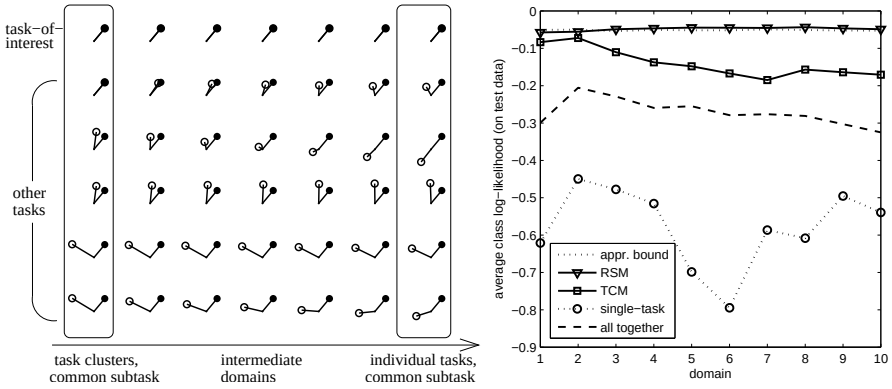

Fig. 1. Comparison of RSM to TCM and two naive methods, on a continuum of domains that are progressively less suited for TCM. **Left**: conceptual illustration; columns are domains and rows are tasks within a domain. All tasks (data sets) are generated from a mixture of two logistic regression models whose weight vectors are shown as lines. One subtask (line ending in the closed ball) is present in all tasks and corresponds to the task-of-interest. The other subtask (line ending in the open ball) is common to clusters of tasks in the leftmost domain, but in the rightmost domain it is different for each individual task. **Right**: Results; 40 learning problems (10 tasks in each problem) were generated from each domain in the continuum, and the four methods were applied to them. The curves show average performances over the replications. RSM maintains high performance for all domains while TCM becomes progressively worse.

5.2 Experiment 2: When Relevant Subtask Modeling Fails

In the previous experiment we showed that RSM outperforms TCM when the assumptions of RSM match the problem domain better. In this experiment we show what happens when the assumptions of RSM go wrong.

The setting is explained in Figure 2 (left). We created a continuum of problem domains as in the first experiment (10 domains, 40 problems per domain, 10 tasks per problem). Here the domain continuum was set up so that the generation of the tasks always follows the assumptions of TCM but the assumptions of RSM become violated progressively more: neither of the two logistic regression models needs to be common to all tasks.

The results are shown in Figure 2 (middle). TCM maintains high performance for all domains, as expected because the tasks come from task clusters in all domains. RSM starts with equal performance but becomes progressively worse as its assumptions begin to fail; however, it is always better than the naive methods (single-task learning and "all together"). The naive methods behave as in the first experiment.

In the above results the task of interest had less data than the other tasks. Intuitively, this means RSM tries to retrieve relevant tasks using very little information for the "query." We ran an additional experiment where the task-of-interest had a comparable amount of data (uniformly picked between 250–500 samples). The results are shown in Figure 2 (right). Single-task-learning overfits somewhat less of course, but the most interesting change is for RSM: it maintains high performance throughout the domain continuum. This is because RSM is able to locate the relevant tasks (here the ones from the same task cluster as the task-of-interest). RSM does not overfit to the other tasks; technically, it models the other tasks almost completely with the task-specific logistic regression model. This demonstrates successful "information retrieval" of relevant tasks.

5.3 Experiment 3: Predicting Document Relevance

In this experiment we have real data, news articles from the standard Reuters-21578 collection, but simulated users in order to again control the problem domain. The "users" browse the collection labeling articles "interesting" or "uninteresting" according to their personal interests. The goal is to learn to predict interestingness of articles for one user, the "user-of-interest." The learning problem can be seen as a combination of collaborative filtering and content-based prediction. Earlier work on such combination includes, for example, [13] where a partly heuristic kernel combination is used, and [14] where naive Bayes is used to fill in missing ratings which are then used in collaborative filtering.

The user-of-interest is interested in a particular news category and labels documents according to whether they belong to that; here we used the second largest Reuters category "acq" because the smaller categories had too few samples. The other nine users are each interested in the "acq" category (and label documents according to it) part of the time, but at other times they are interested in a different category specific to that user.

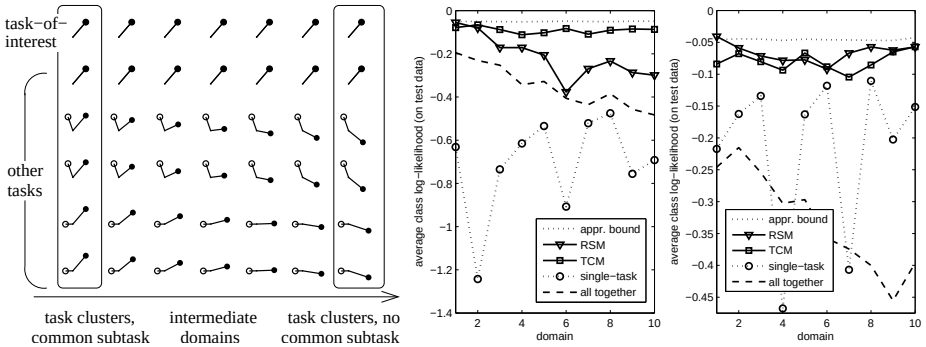

Fig. 2. Comparison of RSM to TCM and two naive methods on domains progressively less suited for RSM. **Left**: conceptual illustration. In all domains, tasks are clustered; tasks in a cluster are generated from the same logistic regression models (shown as lines). In the leftmost domain, one subtask (line ending in the closed ball) is the same in all clusters; this common subtask corresponds to the task-of-interest. In the rightmost domain, all clusters are completely different. All the domains can be learned by TCM; RSM fits the leftmost domain well but not the rightmost domain. **Middle**: Results for a continuum of 10 problem domains (10 tasks in each; all results are averages over 40 replicates); only little data in the task-of-interest (1/3 compared to others). **Right**: Results when the amount of data in the task-of-interest is comparable to the other tasks.

We performed a simplistic feature extraction for this proof-of-concept study. First stopwords were removed, numeric expressions were replaced with tags, etc. Documents were then represented as vectors of word counts. We performed a two-step dimensionality reduction: first, the 200 most "informative" words were selected[1] and other dimensions were discarded (documents with less than 10 words remaining were also discarded); then the dimensionality was further reduced to 5 by linear discriminant analysis. As a result we had 5-dimensional samples, 8194 in total.

As a design parameter we varied how much of the other users' data was labeled according to "acq" on average. For each value we repeated the experiment 10 times to reduce variation due to initializations and sampling variation in the small datasets. In each repetition, 41–83 documents were labeled by the user-of-interest and 250–500 by each of the other users; 1000 documents from the user-of-interest were left apart as test data. On average one third of each user's documents were labeled "interesting" according to that user's specific interests.

We ran RSM, TCM[2], and the naive methods for all experiments. The results are shown in Figure 3 as a function of the design parameter. RSM performs best. Since there is little data for the user-of-interest, single-task learning overfits

[1] In brief, words that had the largest contributions to the empirical mutual information between documents and words were chosen; see [15] for details.

[2] Here we used $K = 6$ clusters to make the number of parameters in RSM and TCM roughly equal.

badly. TCM and the simple "all together" method perform about equally here. At the extreme where all data begins to be relevant, RSM, TCM and "all together" naturally converge to the same performance level.

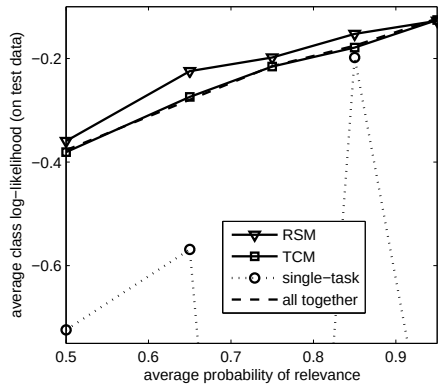

Fig. 3. Comparison of RSM to TCM and two naive methods on Reuters data. Average results over 10 generated problems are shown, as a function of one design parameter, the average probability that a sample is relevant to the task-of-interest. RSM performs the best. Performance of single-task learning varies highly due to overlearning; the worst results (at design parameter values 0.75 and 0.95) do not fit in the figure.

6 Conclusions

We have introduced a new problem called *relevant subtask learning*, where the goal is to use multiple background tasks to help learn one task-of-interest. We showed how a carefully constructed but generally applicable graphical model solves the problem; the crucial idea is to model relevant parts of other tasks with a shared mixture component, and nonrelevant parts by (at least equally) flexible models, to avoid a performance tradeoff between the task-of-interest and the other tasks. Using a simple logistic regression classifier as an example, we showed that the resulting "relevant subtask model" (RSM) outperforms a comparable traditional multi-task learning model and two naive alternatives, on continuums of toy problems and on a more realistic text classification experiment.

The method is not restricted to logistic regression, and in fact not even to supervised learning. In this paper we used simple maximum conditional likelihood estimators, which will be generalized to full-Bayesian treatments of more general models in the next stage.

Acknowledgments. S. Kaski and J. Peltonen belong to both Helsinki Institute for Information Technology and the Adaptive Informatics Research Centre.

They were supported by the Academy of Finland, decision numbers 108515 and 207467. This work was also supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. All rights are reserved because of other commitments.

References

1. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75
2. Marx, Z., Rosenstein, M.T., Kaelbling, L.P.: Transfer learning with an ensemble of background tasks. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop. (2005)
3. Raina, R., Ng, A.Y., Koller, D.: Transfer learning by constructing informative priors. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop. (2005)
4. Niculescu-Mizil, A., Caruana, R.: Inductive transfer for Bayesian network structure learning. In: Proceedings of the 11th International Conference on AI and Statistics (AISTATS 2007). (2007) Electronic proceedings.
5. Rosenstein, M.T., Marx, Z., Kaelbling, L.P.: To transfer or not to transfer. In: Inductive Transfer: 10 Years Later, NIPS 2005 workshop. (2005)
6. Yu, K., Tresp, V., Schwaighofer, A.: Learning Gaussian processes from multiple tasks. In De Raedt, L., Wrobel, S., eds.: Proceedings of the 22nd International Conference on Machine Learning (ICML 2005). ACM Press (2005) 1012–1019
7. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research **6** (2005) 615–637
8. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research **8** (2007) 35–63
9. Bakker, B., Heskes, T.: Task clustering and gating for Bayesian multitask learning. Journal of Machine Learning Research **4** (2003) 83–99
10. Wu, P., Dietterich, T.G.: Improving SVM accuracy by training on auxiliary data sources. In Greiner, R., Schuurmans, D., eds.: Proceedings of the 21st International Conference on Machine Learning (ICML 2004). Omnipress, Madison, WI (2004) 871–878
11. Liao, X., Xue, Y., Carin, L.: Logistic regression with an auxiliary data source. In De Raedt, L., Wrobel, S., eds.: Proceedings of the 22nd International Conference on Machine Learning (ICML 2005). Omnipress, Madison, WI (2005) 505–512
12. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory. (1998) 92–100
13. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In Greiner, R., Schuurmans, D., eds.: Proceedings of the 21st International Conference on Machine Learning (ICML 2004). Omnipress, Madison, WI (2004) 65–72
14. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002). AAAI Press (2002) 187–192
15. Slonim, N., Friedman, N., Tishby, N.: Unsupervised document classification using sequential information maximization. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, New York, NY, USA (2002) 129–136