Appearing in NIPS 2005 workshop "Inductive Transfer: 10 Years Later".

Learning to Learn and Collaborative Filtering

Kai Yu, Volker Tresp
Siemens AG, 81739 Munich, Germany
{kai.yu, volker.tresp}@siemens.com

Abstract

This paper reviews several recent multi-task learning algorithms in a general framework. Interestingly, the framework establishes a connection to recent collaborative filtering algorithms using low-rank matrix approximation. This connection suggests to build a more general *nonparametric* approach to collaborative preference learning that additionally explores the content features of items.

1 Introduction

Learning from multiple related tasks has been known as "multi-task learning" or "learning to learn"[3]. It explores the dependency between learning tasks at hand, and aims (1) to achieve a better performance than the result of learning single tasks independently, and (2) to further generalize previously learned knowledge for benefiting new learning tasks. In this paper we focus on a specific family of multi-task problems where each task is to learn a predictive function and all of them share the same input space and output space.

We first review single-task learning. Given a hypothesis space $\mathcal{H}$ endowed with $\Omega_{\mathcal{H}} : \mathcal{H} \to \mathbb{R}_+$ measuring the complexity of hypotheses, a single-task learner learns a predictive function $f \in \mathcal{H}$ by generalizing measured responses $\mathbf{y} = \{y_i\}$ of f acting on inputs $\mathbf{X} = \{\mathbf{x}_i\}$. Let $\ell(y_i, f(\mathbf{x}_i))$ be the empirical loss, the learning problem amounts to

$$\min_f \sum_i \ell\left(y_i, f(\mathbf{x}_i)\right) + \lambda \Omega_{\mathcal{H}}(f) \tag{1}$$

In contrast, multi-task learning considers m scenarios, where each scenario j has an empirical data set $D_j = (\mathbf{X}_j, \mathbf{y}_j)$. Intuitively, we may be able to find an optimal hypothesis space that is suitable for all the scenarios. Therefore, let us consider *a space of hypothesis spaces* $\mathcal{S}$, where $\mathcal{H}_\theta \in \mathcal{S}$ is a hypothesis space, with θ being the index or parameters of hypothesis spaces. If we can find a $L(D_j, \mathcal{H})$ to measure the cost incurred by applying $\mathcal{H}$ to model the data of scenario j, then a general multi-task learning can be formulated as

$$\min_\theta \sum_j L(D_j, \theta) + \gamma C_{\mathcal{S}}(\theta) \tag{2}$$

where $C_{\mathcal{S}}(\theta)$ measures the complexity of $\mathcal{H}_\theta \in \mathcal{S}$, playing a role similar to $\Omega_{\mathcal{H}}$ in the single-task setting. Then given the optimum $\mathcal{H}^*$, each predictive function f_j can be estimated via a usual single-task learning.

Optimizing the hypothesis space $\mathcal{H}$ amounts to capture the *common structure* shared over multiple predictive tasks. This common structure describes the dependency between them. If comparing the formalisms of signal-task learning (1) and multi-task learning (2), we can find that they appear to be very similar, both end up with minimizing an empirical loss and a complexity penalty. In contrast to generalizing empirical examples to new data in the single-task setting, multi-task learning aims to generalize learned knowledge to new tasks. Within the described framework, now we are able to review several recently proposed multi-task learning methods

Parametric Bayesian Multi-task Learning: The method in [2] considers linear functions $f_j(\mathbf{x}) = \mathbf{w}_j^\top \mathbf{x}$ where $\mathbf{x}$ has finite dimensionality and can be results of an explicit (non-linear) mapping from original features. Following a hierarchical Bayesian model, each $\mathbf{w}_j$ is sampled from a common prior Gaussian distribution $\mathcal{N}(\mathbf{w}_j|\mathbf{m}, \mathbf{\Sigma})$. Then the learning is to maximize the marginalized log-likelihood of D_j with respect to the parameters $\theta = (\mathbf{m}, \mathbf{\Sigma})$. This is equivalent to define the loss $L(D_j, \theta) = -\log \int p(\mathbf{y}_j|\mathbf{X}_j, \mathbf{w}_j)p(\mathbf{w}_j|\mathbf{m}, \mathbf{\Sigma})d\mathbf{w}_j$ and then solve

$$\min_\theta \sum_j L(D_j, \theta) \tag{3}$$

Compared to the formulation (2), this approach has no prior on θ and is thus not fully Bayesian. Due to the lack of complexity control, the learned knowledge θ^* from existing tasks might not be generalizable to new tasks.

Regularized Multi-task Learning: Recently a multi-task learning algorithm based on support vector machines was proposed [4]. It decomposes the linear function's weights as $\mathbf{w}_j = \mathbf{v}_j + \mathbf{w}_0$, where $\mathbf{w}_0$ is the same for all the tasks and $\mathbf{v}_j$ are independent from each other. Let $\boldsymbol{\theta} = \mathbf{w}_0$, then the problem can be formulated as (2) with

$$L(D_j, \theta) = \min_{\mathbf{v}_j} \sum_i (y_i^{(j)} - \mathbf{w}_j^\top \mathbf{x}_i)_+ + \lambda \|\mathbf{v}_j\|^2 \tag{4}$$

$$C_{\mathcal{S}}(\theta) = \|\mathbf{w}_0\|^2 \tag{5}$$

where $(\cdot)_+$ is the hinge loss. From a Bayesian point of view, the model assumes $\mathbf{w}_j$ following $\mathcal{N}(\mathbf{w}_0, \sigma\mathbf{I})$. As a consequence, it only estimates those functions' mean $f_0(\mathbf{x}) = \mathbf{w}_0^\top \mathbf{x}$ while ignoring their second-order dependencies, due to the fact that the covariance of $\mathbf{w}_j$ is fixed, not adapted to observed tasks.

Common Feature Mapping: A family of methods [8, 11, 1] learn to *explicitly* map the inputs $\mathbf{x}$ into a latent space via $\mathbf{t} = \phi_\theta(\mathbf{x})$. Then in the new feature space each task can be independently treated as a single-task learning problem $f_j(\mathbf{x}) = \mathbf{w}_j^\top \phi_\theta(\mathbf{x})$. Let $\mathbf{w}_j$ follow a Gaussian with zero mean and unitary covariance, one can easily write down the marginalized likelihood of D_j given θ. If θ follows a prior distribution $p(\theta)$, the multi-task learning can be formulated as (2) with

$$L(D_j, \theta) = -\log p(\mathbf{y}_j|\mathbf{X}_j, \theta) \tag{6}$$

$$C_{\mathcal{S}}(\theta) = -\log p(\theta) \tag{7}$$

which gives the penalized maximum likelihood estimate θ^*. Very importantly $p(\theta)$ encodes the prior knowledge about the hypothesis space. For example, in a very recent paper [11] θ is enforced to produce latent variables that are maximally independent, and thus a higher order dependency of functions is considered. Usually feature mapping methods have to limit the dimensionality of latent space, and hence restrict the degrees of freedom of predicting functions. Another approach [1] alleviates the problem by assuming $f_j(\mathbf{x}) = \mathbf{v}_j^\top \psi(\mathbf{x}) + \mathbf{w}_j^\top \phi_\theta(\mathbf{x})$. Since the first part $\mathbf{v}_j^\top \psi(\mathbf{x})$ is independent over tasks and can possibly work with infinite dimensional features, f has no direct restrictions. But the common structure itself is still restricted by a parametric feature mapping $\phi_\theta(\mathbf{x})$ with a predefined dimensionality.

Nonparametric hierarchical Gaussian processes: The approach in [10] improves the parametric Bayesian method [2] from two perspectives: (1) In order to prevent overfitting, a conjugate normal-inverse-Wishart prior $p(\mathbf{m}, \mathbf{\Sigma})$ is used to control the complexity of $\theta = (\mathbf{m}, \mathbf{\Sigma})$; (2) The parametric functions are generalized to be infinite-dimensional and nonparametric. Thus the common structure is directly defined on the function space, characterized by a mean function f_0 and kernel $\mathbf{K}$. Interestingly, the prior $p(\mathbf{m}, \mathbf{\Sigma})$ in the parametric case corresponds to another normal-inverse-Wishart distribution $p(f_0, \mathbf{K})$ in the nonparametric case [6]. Then the multi-task learning is equivalent to kernel learning by $\min_{f_0, \mathbf{K}} \sum_j L(D_j, f_0, \mathbf{K}) + C(f_0, \mathbf{K})$ with

$$L(D_j, f_0, \mathbf{K}) = -\log \int p(\mathbf{y}_j | \mathbf{X}_j, f) p(f | f_0, \mathbf{K}) df \tag{8}$$

$$C(f_0, \mathbf{K}) = -\log p(f_0, \mathbf{K}) \tag{9}$$

An efficient EM algorithm has been developed. Compared to other multi-task learning methods, hierarchical GP does not restrict the dimensionality of either predictive functions or the shared common structure, and models both the first (i.e. common mean) and second (i.e. kernel) order dependencies of tasks. Note that [9] also presented an EM algorithm for kernel matrix completion, which is completely different in terms of purposes and underlying principles.

2 Collaborative Filtering as Multi-task Learning

Collaborative filtering (CF) predicts a user's preferences (i.e. ratings) on new products (i.e. items) based on other users' ratings, following the assumption that users sharing same ratings on past items tend to agree on new items.

2.1 Collaborative Filtering via Low-Rank Matrix Approximation

Let $\mathbf{Y} \in \mathbb{R}^{n \times m}$ be the matrix representing m users's ratings on n items. Since typically each user has rated only a small number of items, the matrix $\mathbf{Y}$'s most entries have missing values. CF can be thought as a matrix completion problem. One way to do so is to perform a low-rank approximation $\mathbf{Y} \approx \mathbf{U}\mathbf{V}^\top$ [7, 5], where $\mathbf{U} \in \mathbb{R}^{n \times k}, \mathbf{V} \in \mathbb{R}^{m \times k}$. Given the matrix approximation, user j's ratings on item i can be predicted as $\arg\min_y \ell(y, \mathbf{U}_i \mathbf{V}_j^\top)$ given a predefined loss $\ell(\cdot, \cdot)$. A maximum-margin factorization of the matrix $\mathbf{Y}$ with missing values is formulated as

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \beta \sum_{(i,j) \in S} \ell(Y_{i,j}, \mathbf{U}_i \mathbf{V}_j^\top) \tag{10}$$

where S is the index set for non-missing entries. The Frobenius norms of $\mathbf{U}$ and $\mathbf{V}$ serve as regularization terms. When k goes infinity, [7] shows the problem can be solved as semidefinite programming problem (SDP), which however scales to only hundreds of users and items in their experiments. Very recently, [5] suggests an alternating optimization that demonstrates a good scalability when k is finite, e.g. $k = 100$.

2.2 Collaborative Filtering via Multi-Task Learning

We first show an equivalence between maximum-margin matrix approximation based CF and multi-task learning. The connection will derive a new nonparametric CF method which works with an infinite dimensionality while still remains scalability on large data sets.

Theorem 2.1 *If $\mathbf{K} = \mathbf{U}\mathbf{U}^\top$ is full rank, then the problem (10) can be formulated as*

$$\min_{\mathbf{K}} \sum_j L(\mathbf{Y}_j, \mathbf{K}) + \gamma C(\mathbf{K}),$$

with

$$L(\mathbf{Y}_j, \mathbf{K}) = \min_{\mathbf{f}_j \in \mathbb{R}^n} \sum_{i \in S_j} \ell(Y_{i,j}, \mathbf{f}_j(i)) + \mathbf{f}_j^\top \mathbf{K}^{-1} \mathbf{f}_j, \quad C(\mathbf{K}) = \text{trace}(\mathbf{K})$$

where $\mathbf{Y}_j$ *are the ratings from user* j *and* S_j *the index set for items rated by user* j

The proof is done by an application of the *representor theorem* and the identity $\|\mathbf{U}\|_F^2 = \text{trace}(\mathbf{U}\mathbf{U}^\top)$. Theorem 2.1 shows that low-rank matrix approximation based CF can be formulated as a kernel learning problem similar to the hierarchical GP multi-task learning [10]: each user j is modeled by a predictive function $\mathbf{f}_j$ and the common structure is modeled by a kernel matrix $\mathbf{K}$. The new model avoids to directly specify the (possibly infinite) dimensionality of $\mathbf{U}$, but just bounds the trace of $\mathbf{K}$.

Based on the connection we suggest to apply hierarchical GP to collaborative filtering based on our previous work [6, 10]. The approach has certain advantages over the derived learning problem in theorem 2.1: (1) The mean function $\mathbf{f}_0$ is now explicitly modeled, which reflects people's average ratings on all the items; (2) The empirical loss $D_j(\mathbf{Y}_j, \mathbf{K})$ in theorem 2.1 is computed via the *mode estimate* of $\mathbf{f}_j$, while GP computes the loss in a more robust way by the integral over the *entire distribution* $p(\mathbf{f}_j|\mathbf{f}_0, \mathbf{K})$; (3) Each user j's ratings on item i is predicted by not only the mean but also the variance. An assessment to the confidence of predictions is necessary in building recommender systems; (4) In hierarchical GP the kernel matrix is computed from a basic kernel function $\kappa(\mathbf{x}, \mathbf{z})$, thus the problem amounts to learn a kernel function based on the content features $\mathbf{x}_i$ of items i, which leads to a novel collaborative filtering algorithm that explores the content features of items; (5) The penalty in hierarchical GP comes from a general prior on $\mathbf{f}_0$ and $\mathbf{K}$. Replacing trace($\mathbf{K}$) by a *conjugate prior*, namely a normal-inverse-Wishart distribution $p(\mathbf{f}_0, \mathbf{K})$, will leads to a tractable algorithm to estimate $\mathbf{f}_0$ and $\mathbf{K}$. A simple EM algorithm to estimate $\mathbf{f}_0$, $\mathbf{K}$ and $\mathbf{f}_j$, $j = 1, \ldots, m$ is scalable to tens of thousands of users and thousands of items, with the complexity $O(ml^3)$, where m is user size and l is average number of ratings per user. Since typically a user rated a small set of items (e.g. $l = 38$ in EachMovie), the algorithm has a linear scalability to the user size.

3 Preliminary Experiments

A preliminary experiment was run on EachMovie data set, with 10,000 users (having more than 20 ratings), 1,648 movies. and 380,000 ratings taking values $\{1, \ldots, 6\}$. 30% ratings were hold out for evaluation. The algorithm took about 5 hours on a laptop with a 1.4 GHz CPU. The normalized mean absolute error (NMAE) was 0.442, comparable to the best results reported so far (see [5]). Very interestingly, GP model produced very accurate estimates of prediction errors, computed as summation of predictive variance and estimated noise variance. This feature will enable us to know how reliable an individual recommendation is. Currently we are testing the algorithm on several data sets and making comparisons with other algorithms. Besides NMAE, some ranking score will also be evaluated.

References

[1] Ando, R. K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data, 2004. Technical Report RC23462, IBM T.J. Watson Research Center.

[2] Bakker, B. and Heskes, T. Task clustering and gating for Bayesian multitask learning. *The Journal of Machine Learning Research archive*, 4, 2003.

[3] Caruana, R. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[4] Evegniou, T. and Pontil, M. Regularized multi–task learning. In *Proc. of 17–th SIGKDD Conf. on Knowledge Discovery and Data Mining*. 2004.

[5] Rennie, J. D. M. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*. 2005.

[6] Schwaighofer, A., Tresp, V., and Yu, K. Hierarchical Bayesian modelling with Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS) 18*. MIT Press, 2005.

[7] Srebro, N., Rennie, J. D. M., and Jaakola, T. S. Maximum-margin matrix factorization. In *Neural Information Processing Systems (NIPS) 18*. 2005.

[8] Teh, Y., Seeger, M., and Jordan, M. Semiparametric latent factor models. In *AISTATS*. 2005.

[9] Tsuda, K., Akaho, S., and Asai, K. The EM algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.

[10] Yu, K., Tresp, V., and Schwaighofer, A. Learning Gaussian processes from multiple tasks. In *Proceedings of 22nd International Conference on Machine Learning (ICML 2005)*. Bonn, Germany,, 2005.

[11] Zhang, J., Ghahramani, Z., and Yang, Y. Learning multiple related tasks using latent independent component analysis. In *to appear in NIPS*. 2005.