Manifold Regularized Multi-Task Learning

Peipei Yang, Xu-Yao Zhang, Kaizhu Huang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences,
Beijing, China 100190
{ppyang,xyz,kzhuang,liucl}@nlpr.ia.ac.cn

Abstract. Multi-task learning (MTL) has drawn a lot of attentions in machine learning. By training multiple tasks simultaneously, information can be better shared across tasks. This leads to significant performance improvement in many problems. However, most existing methods assume that all tasks are related or their relationship follows a simple and specified structure. In this paper, we propose a novel manifold regularized framework for multi-task learning. Instead of assuming simple relationship among tasks, we propose to learn task decision functions as well as a manifold structure from data simultaneously. As manifold could be arbitrarily complex, we show that our proposed framework can contain many recent MTL models, e.g. RegMTL and cCMTL, as special cases. The framework can be solved by alternatively learning all tasks and the manifold structure. In particular, learning all tasks with the manifold regularization can be solved as a single-task learning problem, while the manifold structure can be obtained by successive Bregman projection on a convex feasible set. On both synthetic and real datasets, we show that our method can outperform the other competitive methods.

Keywords: Multi-task Learning, Manifold Learning, Laplacian.

1 Introduction

In many machine learning problems, we usually have multiple corrected learning problems or tasks. Traditionally we can train each task from its training samples individually. However, if the number of training samples in each task is small, they tend to be overfitting, meaning that the performance is very likely to be bad for future samples. To handle this problem, multi-task learning (MTL) manages to learn all tasks simultaneously. By sharing information across related tasks, MTL can usually lead to better performance than the traditional single task learning.

However, in order to share information appropriately, MTL often needs to assume how the tasks are correlated. Given that a linear decision function is to be learned for each task, the relationship among tasks can be specified directly via the weight vectors associated with the decision function. For example, [7] proposed the *Regularized Multi-task Learning* (RegMTL) which assumes that the weight vector of each task is composed with a common part and an individual

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 528–536, 2012.

part. The common part contains the shared information of all the tasks and the propagation of information is enforced by minimizing the individual part for each task. It equivalently implies that the weight vectors of different tasks belong to a ball of an unknown center determined by the common part.

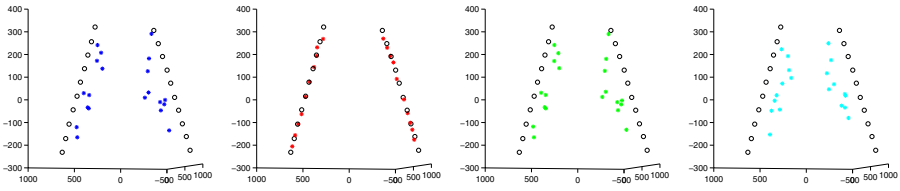
Unfortunately, such an assumption may be too strict in practice, since it is unnecessary for each task to be related with all other tasks. To solve this problem, [9] generalized this assumption to the case that these tasks can gather into several clusters and proposed the *convex Clustered Multi-task Learning* (cCMTL) [10] method. Within each cluster, it is a traditional MTL problem, i.e., the weight vectors of different tasks in a certain cluster are in a ball of an unknown center determined by the common part. cCMTL can learn the weight vectors of all tasks and the cluster structure simultaneously.

Although cCMTL provides a tool to capture the topological structure of the relationship among tasks, its assumption is still too strong and may be too simple to explore the actual task relationship. On one hand, tasks may be unable to be partitioned into several groups. On the other hand, even if several tasks belongs to a cluster, it never means each task within this cluster is correlated with each other at the same level.

Hence, the structure of the relationship between tasks could be more complex, and a general manifold structure should be considered. Take an example for illustration. Consider the problem that there are 20 related regression tasks. To show the relationship between tasks, we plot the weight vectors in Fig. 1 using hollow points in a 3-dimensional space. From this figure, the weight vectors gather into 2 clusters and each of them forms a 1-dimensional manifold. To the extent of our knowledge, there has not been a method designed to deal with this case.

Since manifold has the ability to describe not only the topological structure of data, but also the local metric structure, we propose Manifold Regularized Multi-task Learning (MRMTL) which engages manifold to capture the relationship among the tasks. All tasks and the manifold structure of their relationship are learned simultaneously, and both of them are improved with the help of each other. As manifold could be arbitrarily complex, we show that our proposed framework can contain many recent MTL models, e.g. RegMTL and cCMTL, as special cases. Moreover, the proposed framework can be solved by learning all tasks and the manifold structure alternatively. In particular, learning all tasks with the manifold regularization can be solved as a single-task learning problem, while the manifold structure can be obtained by successive Bregman projection on a convex feasible set. It is noticeable that [8] has studied the multi-task learning problem with manifold regularization. However, it supposed that the manifold structure is given preliminarily. As a key difference, our proposed approach can learn the manifold from the training samples automatically.

The rest part of this paper is organized as follows. In Section 2, we first present the problem definition and then introduce the basic framework of our method. In Section 3, the optimization algorithm is given in detail. In Section 4, we evaluate our method on a synthetic dataset and a real dataset, both of which show the effectiveness of our method. At last, we set out the final remarks in Section 5.



(a) STL (b) MRMTL (c) cCMTL (d) RegMTL

Fig. 1. Learned weight vectors W using different methods. The hollow points are ground truth while the star points are learned results.

2 Problem Definition and Main Framework

In this section, we first present the notation and problem definition. We then introduce the framework of Manifold Regularized Multi-task Learning in detail.

2.1 Notation and Problem Definition

In this paper, we consider the problem where a linear decision function is learned and thus the aim of each task is to learn a weight vector. Suppose there are n tasks. For the t-th task, we have a training data set $\mathcal{X}_t$ containing m_t data points $\mathbf{x}_{tk} \in \mathbb{R}^d$ whose dimension is d and a corresponding output set $\mathcal{Y}_t$ containing the target output y_{tk}. For binary classification problem, $\mathcal{Y}_t = \{-1, +1\}$, while for regression problem, $\mathcal{Y}_t = \mathbb{R}$.

We use $l(y, f(\mathbf{x}))$ to quantify the loss of predicting $f(\mathbf{x})$ for the input $\mathbf{x}$ when the expected output is y, which depends on the problem. For example, in binary classification, the hinge loss $l(y, f(\mathbf{x})) = \max(0, 1 - y \cdot f(\mathbf{x}))$ is often used, while in regression, the squared error $l(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ is often chosen. If the linear prediction function $f(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$ is used and we denote $W = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n]$, the empirical loss of all tasks can be then formulated as $\ell(W) = \sum_{t=1}^{n} \sum_{j=1}^{m_t} l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj})$.

2.2 Coupling Multiple Tasks with Regularization

In order to learn all the tasks simultaneously, we follow the well-established method that embeds the relationship among tasks into a regularization item and use a graph to describe the relationship among tasks. Specifically, each vertex of the graph represents a task, and each edge linking two vertices indicates the relationship between the two tasks. A greater weight of edge represents a closer relationship. Define S as the weight matrix of this graph where S_{ij} is the weight of the edge connecting the i-th and j-th vertices and D is a diagonal weight matrix whose entries are column sums of S, then $L = D - S$ is the Laplacian matrix [5] of this graph.

In Laplacian regularization [2], we have $\mathrm{tr}(WLW^\top) = \sum_{i,j} \frac{1}{2} \|\mathbf{w}_i - \mathbf{w}_j\|^2 S_{ij}$, which can be then used as the regularization to enforce the linked pairs to be

more similar. If the i-th and j-th tasks are closely correlated, the corresponding edge weight S_{ij} is large, which encourages $\|\mathbf{w}_i - \mathbf{w}_j\|^2$ to be less and thus the learned weight vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ are more liable to be similar.

However, in MTL, such task similarity S_{ij} is unavailable beforehand and should be learned from data. It is obvious that if we directly optimize on L and W simultaneously, we will simply obtain the Laplacian matrix L with all elements zero regardless of W. Therefore, in order to discover the relationship among tasks, we should add some additional constraints on L. Without more prior knowledge, a Laplacian matrix of a graph whose vertices are all connected may be a reasonable prior of L. Therefore, we get the following optimization formula of MRMTL

$$\min_{W,L} \mathcal{R}(W,L) = \sum_{t=1}^{n} \left(C \sum_{j=1}^{m_t} l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj}) + \mathbf{w}_t^\top \mathbf{w}_t \right) + \gamma \left(\mathrm{tr}(WLW^\top) + \frac{\gamma_0}{2} \|L - L_0\|_{\mathrm{F}}^2 \right)$$
$$\text{s.t. } L\mathbf{1}_n = \mathbf{0}; \quad L = L^\top; \quad L_{ij} \leq 0, \forall i \neq j$$

where L_0 is the Laplacian matrix for a graph with all nodes connected ($S_{ij} = 1, \forall i, j$), i.e., $L_0 = n(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)$, where $W = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n]$, and $\mathbf{1}_n$ is an n-dimensional vector with all elements 1. In this formulation, l is the loss from training samples and $\mathbf{w}_t^\top \mathbf{w}_t$ is the regularization. Both of them are determined by the original learning problem. $\mathrm{tr}(WLW^\top)$ is the manifold regularization which enforces the weight vectors of similar tasks to be similar. The last term provides a prior for L and prevents the trivial solution for L. The constraints guarantee that L is a Laplacian matrix, which is therefore also symmetric positive semi-definite.

2.3 Relationship with Other Methods

It is noticeable that our method includes RegMTL as a special case. Indeed, if we choose γ_0 to be large enough, we will get $L = L_0$ and the regularization item becomes $\mathrm{tr}(WL_0W^\top) = n \cdot \left\|W - \bar{\mathbf{w}}\mathbf{1}_n^\top\right\|_{\mathrm{F}}^2 = n \sum_{t=1}^{n} \|\mathbf{w}_t - \bar{\mathbf{w}}\|^2$.

By Lemma 2.2 of [7], this problem is an alternative formulation of RegMTL and thus it is just a special case of MRMTL. We can also regard MRMTL as a generalized of RegMTL in which the relationship among tasks is learned using L_0 as prior, rather than to use L_0 directly.

When the tasks gather into several clusters, [9] uses the $m \times r$ binary matrix E to denote the cluster assignment where $E_{ij} = 1$ if task-i belongs to cluster-j and $E_{ij} = 0$ otherwise. Define $M = E(E^\top E)^{-1}E^\top$, $U = \mathbf{1}_m\mathbf{1}_m^\top/m$, then M is the edge weight matrix of the graph of tasks where $M_{ij} = 1/m_c$ if task-i and task-j belong to the same cluster-c and $M_{ij} = 0$ otherwise, where m_c is the number of tasks in cluster-c. The regularization with respect to the task clustering is

$$\mathrm{tr}(WKW^\top) = \mathrm{tr}\left(W\left(\varepsilon_B(M-U) + \varepsilon_W(\mathbf{I}-M)\right)W^\top\right).$$

It is easy to verify that K is a Laplacian matrix if $\varepsilon_W \geq \varepsilon_B$, which is satisfied in cCMTL [9]. Therefore, our method indeed also includes clustered multi-task

learning as a special case in the sense that any cluster structure of the tasks can be formulated using our model. However, the solution may be different since our model is more flexible to fit the data.

3 Optimization

In this section, we first present how to solve the problem using alternative optimization, and then show how each step of the optimization is solved.

3.1 Alternative Optimization

This problem can be solved by alternative optimization. Specifically, we solve for an optimal $W^{(1)}$ with $L = L^{(0)}$ fixed as an initial value first, and then solve for an optimal $L^{(1)}$ with $W = W^{(1)}$ fixed. Such procedure is then repeated so that both L and W are optimized alternatively until convergence. Since l is usually chosen to have a lower bound and L is constrained to be positive semi-definite, there exists a lower bound for $\mathcal{R}(W, L)$. In another respect, the value of objective function decreases in each iteration, and thus it is guaranteed to converge to a local minimal value after certain iterations.

Note that the optimization is not convex and the global optimal solution is not guaranteed. Nevertheless, we found that given a proper initial solution, the local optimal solution is often good enough. Since W is solved firstly, we should specify an initial point for L. An appropriate choice is $L^{(0)} = L_0$. With this choice, $W^{(1)}$ is indeed the solution of RegMTL. After several iterations, the incorrect connections in the graph are removed and the manifold can be eventually learned.

In the following of this section, we will give the algorithm to solve W and L respectively in detail.

3.2 Fix L and Optimize on W

The part of $\mathcal{R}$ with respect to W is

$$\mathcal{R}(W) = \sum_{t,j} C \cdot l(y_{tj}, \mathbf{w}_t^\top \mathbf{x}_{tj}) + J(W), \text{ where } J(W) = \operatorname{tr}\left(W(\mathbf{I}_n + \gamma L)W^\top\right) \quad (1)$$

Denote $\mathbf{w} = \operatorname{vec}(W) = \left[\mathbf{w}_1^\top\ \mathbf{w}_2^\top\ \dots\ \mathbf{w}_n^\top\right]^\top$ as the vector concatenated by $\{\mathbf{w}_t\}$, then by Proposition 31 of [3], we have $\operatorname{vec}(Y)^\top (A \otimes B)\operatorname{vec}(X) = \operatorname{tr}(A^\top Y^\top B X)$ and thus

$$J(W) = \operatorname{tr}((\mathbf{I}_n + \gamma L)W^\top \mathbf{I}_d W) = \mathbf{w}^\top E \mathbf{w} = J(\mathbf{w}), \text{ where } E = (\mathbf{I}_n + \gamma L)^\top \otimes \mathbf{I}_d.$$

Suppose $B^\top B = E^{-1} = ((\mathbf{I}_n + \gamma L)^\top \otimes \mathbf{I}_d)^{-1} = (\mathbf{I}_n + \gamma L)^{-1} \otimes \mathbf{I}_d$ and consider the problem

$$\min_{\mathbf{u}} \mathcal{S}(\mathbf{u}) = \sum_t \sum_j C \cdot l(y_{tj}, \mathbf{u}^\top B_t \mathbf{x}_{tj}) + \mathbf{u}^\top \mathbf{u}, \text{ where } B = [B_1\ B_2\ \dots\ B_n]. \quad (2)$$

By Proposition 1 of [8], we have $\mathcal{S}(\mathbf{u}) = \mathcal{R}(B^\top \mathbf{u})$. Thus the optimal solution of (1) can be obtained by solving the single-task problem (2) and $\mathbf{w}_t = B_t^\top \mathbf{u}$.

3.3 Fix W and Optimize on L

When W is fixed, the optimization problem on L becomes

$$\begin{aligned} &\min_L \mathcal{R}(L) = \gamma \left(\frac{\gamma_0}{2} \|L - L_*\|_{\mathrm{F}}^2 + \mathcal{R}_{\mathrm{const}} \right) \\ &\text{s.t. } L\mathbf{1}_n = \mathbf{0}; \quad L = L^\top; \quad L_{ij} \leq 0, \forall i \neq j \end{aligned} \tag{3}$$

where $L_* = L_0 - \frac{1}{\gamma_0} W^\top W$ and $\mathcal{R}_{\mathrm{const}}$ is a constant independent of L. This is a *Bregman projection* problem [6] whose optimal solution is the projection of L_* on the convex set $\mathbf{C}_1 \cap \mathbf{C}_2$ where $\mathbf{C}_1 = \{L \in \mathbb{R}^{n \times n} \mid L\mathbf{1}_n = \mathbf{0}; L = L^\top\}$ and $\mathbf{C}_2 = \{L \in \mathbb{R}^{n \times n} \mid L_{ij} \leq 0, \forall i \neq j\}$. The optimal solution of L can be obtained by Successive Projection-Correction Algorithm (Algorithm B of [6]) on these two convex sets.

Projection onto $\mathbf{C}_1$. The Lagrangian formulation[1] of the projection on $\mathbf{C}_1$ is

$$\min_{L, \mu_1, \mu_2} \|L - L_*\|_{\mathrm{F}}^2 - \mu_1^\top L\mathbf{1}_n - \mu_2^\top L^\top \mathbf{1}_n$$

where from the condition $L = L^\top$ we have $\mu_1 = \mu_2 = \mu$. Setting the derivative with respect to L to zero yields $L = L_* + \frac{1}{\gamma_0}\mu\mathbf{1}_n^\top + \frac{1}{\gamma_0}\mathbf{1}_n\mu^\top$. Multiplying with $\mathbf{1}_n$ on the right of both sides of the equation, then using Sherman-Morrison inverse formula [1] and $L_* = L_*^\top$, we have

$$\mu = -\gamma_0 \left(n\mathbf{I}_n + \mathbf{1}_n\mathbf{1}_n^\top\right)^{-1} L_*\mathbf{1}_n = \frac{\gamma_0}{n}\left(\frac{1}{2n}\mathbf{1}_n\mathbf{1}_n^\top - \mathbf{I}_n\right) L_*\mathbf{1}_n$$

Then substituting it into the formula of L, we get

$$L = L_* + \frac{1}{n^2}\left(\mathbf{1}_n^\top L_*\mathbf{1}_n\right)\mathbf{1}_n\mathbf{1}_n^\top - \frac{1}{n}\left(L_*\mathbf{1}_n\mathbf{1}_n^\top + \mathbf{1}_n\mathbf{1}_n^\top L_*\right)$$

Projection onto $\mathbf{C}_2$. The projection onto $\mathbf{C}_2$ can be obtained by simply setting the positive non-diagonal elements to zero following a correction step [6].

4 Experiments

In this section, we empirically evaluate our method on both artificial data and real data. We apply our method on regression problems, and the normalized mean square error (nMSE) [4] is used as the performance measure. Specifically, it is defined as the mean squared error (MSE) divided by the variance of the target vector.

[1] The coefficient $\frac{\gamma\gamma_0}{2}$ is simply omitted.

We compare our method MRMTL with cCMTL [9], RegMTL [7], and single-task (STL) method as baseline. For each method, we use 5-fold cross validation to determine the regularization parameters.

4.1 Synthetic Data

We first evaluate on synthetic data set to give a visualized comparison of the results learned by these methods. We generate 20 related regression tasks using 20 weight vectors and then generate a certain number of training samples and 500 testing samples. The weight vectors are learned with the training samples using different methods and tested with the testing samples. We show the learned task relationship in Fig. 2 which is a 20×20 grid. The color of the grid on row-i and column-j represents the squared Euclidean distance of $\mathbf{w}_i$ and $\mathbf{w}_j$. From the results, we see that MRMTL can learn the task relationship surprisingly well, which coincides with the ground truth perfectly when the training samples is equal to 30, 40, and 50. It always gives the best performance compared with the other methods. Particularly, it demonstrates a significantly better performance than the other methods when the training samples are fewer. For the case where the number of training samples per task is 30, we also show the learned weight vectors in the 3-dimensional principal component subspace in Fig. 1. The hollow points represents

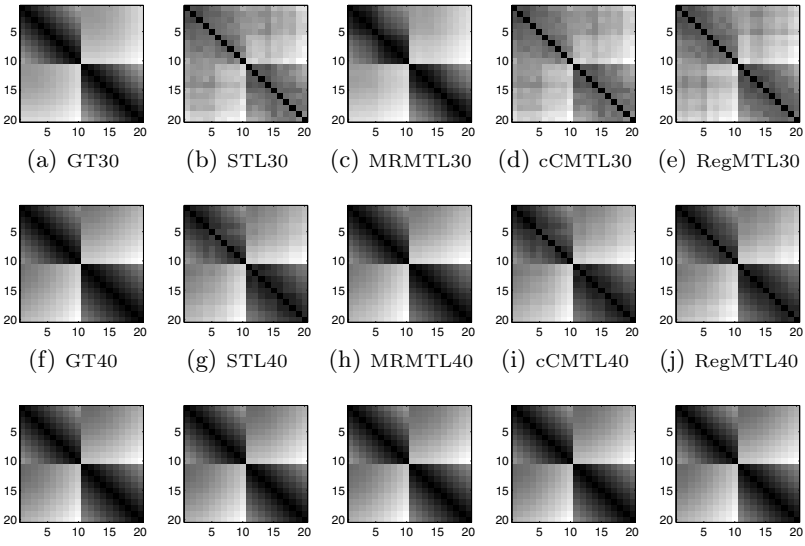


Fig. 2. Comparison of the weight vectors learned by different methods. The five columns of this figure correspond to the (1)Ground Truth (GT); (2)Single-task Learning (STL); (3)Manifold Regularization Multi-task Learning (MRMTL); (4)convex Clustered Multi-task Learning (cCMTL); (5)Regularized Multi-task Learning (RegMTL). The number in the title indicates how many percent of training samples are used.

the ground truth while star points represent the learned results. We see again that MRMTL gives the best result and the manifold is learned exactly.

4.2 Real Data

We also evaluate these methods on Sarcos data[2] [10], which relates to an inverse dynamics prediction problem for a seven degrees-of-freedom anthropomorphic robot arm. It consists of 48933 observations corresponding to 7 joint torques; each of the observations is described by 21 features including 7 joint positions, 7 joint velocities, and 7 joint accelerations. The prediction of each joint torque corresponds to one task. We randomly select 10, 20, 50, 100 samples from each task for training and the remaining for test. The experiment is repeated 5 times and the averaged nMSE (the less the better) are shown in Table. 1. From the results, we can observe that MRMTL performs the best, regardless of the number of samples used for training.

Table 1. Performance comparison on Sarcos Dataset using nMSE

Sample	STL	MRMTL	cMTL	RegMTL
10	2.8788	1.7843	2.7532	2.8867
20	0.8383	0.5487	0.7953	0.5766
50	0.2615	0.1709	0.4377	0.2066
100	0.1664	0.1188	0.3378	0.1202

5 Conclusion

In this paper, we propose a novel manifold regularized framework for multi-task learning. Different from recent work that usually assumes simple relationship among tasks, we propose to learn task decision functions as well as a manifold structure from data simultaneously. We show that our proposed framework can subsume many recent MTL models, e.g. RegMTL and cCMTL, as special cases. Moreover, the framework can be solved by alternatively learning all tasks and the manifold structure. A series of experiments on both synthetic and real data show that our method can significantly outperform the other competitive methods.

Acknowledgements. This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316301, the National Natural Science Foundation of China (NSFC) Grants 61075052 and 60825301, and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

[2] `http://gaussianprocess.org/gpml/data/`

References

1. Bartlett, M.S.: An Inverse Matrix Adjustment Arising in Discriminant Analysis. The Annals of Mathematical Statistics 22(1), 107–111 (1951)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2002)
3. Broxson, B.J.: The kronecker product. UNF Theses and Dissertations (2006)
4. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 42–50 (2011)
5. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics), vol. 92 American Mathematical Society (February 1997)
6. Dhillon, I.S., Tropp, J.A.: Matrix nearness problems with bregman divergences. SIAM Journal on Matrix Analysis and Applications 29, 1120–1146 (2008)
7. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)
8. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6, 615–637 (2005)
9. Jacob, L., Bach, F., Vert, J.P.: Clustered multi-task learning: A convex formulation. In: NIPS, pp. 745–752 (2008)
10. Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. Advances in Neural Information Processing Systems 24, 702–710 (2011)