Multi-Task Learning via Matrix Regularization

Andreas Argyriou

Department of Computer Science
University College London

Collaborators

- T. Evgeniou (INSEAD)
- R. Hauser (Oxford)
- A. Maurer (Stemmer Imaging)
- C.A. Micchelli (SUNY Albany)
- M. Pontil (University College London)
- Y. Ying (University of Bristol)

Outline

- Regularization with matrix variables for multi-task learning
- Learning multiple tasks on a subspace & an alternating algorithm
- Necessary and sufficient conditions for representer theorems
- Learning convex combinations of a finite or infinite number of kernels

Learning Multiple Tasks Simultaneously

- Task $=$ supervised regression/classification task
- Learning multiple related tasks vs. learning independently
- Few data per task; pooling data across related tasks
- Should generalize well on given tasks and on new tasks (*transfer learning*)
- Example: prediction of consumers' preferences to products

Example (Computer Survey)

- Consumers' ratings of products [Lenk et al. 1996]
- 180 persons – each person is a task
- A number of PC models with 13 binary input variables (RAM, CPU, price etc.)
- Integer output in $\{0, \ldots, 10\}$ (likelihood of purchase)
- Can one exploit the fact that *these tasks are related*? What representation do we *transfer* to new persons/tasks ?

Learning Paradigm

- Tasks $t = 1, \dots, n$
- m examples per task: $(x_{t1}, y_{t1}), \dots, (x_{tm}, y_{tm}) \in \mathbb{R}^d \times \mathbb{R}$
- Predict using functions $f_t(x) = \langle w_t, x \rangle$
- Matrix regularization problem w.r.t.

$$W = \begin{pmatrix} | & & | \\ w_1 & \dots & w_n \\ | & & | \end{pmatrix}$$

Learning Multiple Tasks on a Subspace

- Solve the problem [*Argyriou, Evgeniou, Pontil 2006*]

$$\min_{\substack{w_1,\dots,w_n \in \mathbb{R}^d \\ D \succ 0,\ \operatorname{tr}(D) \le 1}} \sum_{t=1}^{n} \sum_{i=1}^{m} E\left(\langle w_t, x_{ti} \rangle, y_{ti}\right) + \gamma \quad \operatorname{tr}(W^\top D^{-1} W)$$

$$\uparrow$$

$$\sum_{t=1}^{n} \langle w_t, D^{-1} w_t \rangle$$

- *Jointly convex* problem

- Learning a *common linear kernel* $(K(x, x') = x^\top D x')$ within a convex set generated by *infinite* kernels: $\{D : D \succ 0,\ \operatorname{tr}(D) \le 1\}$

Learning Multiple Tasks on a Subspace (contd.)

- The optimal values satisfy $\hat{D} \propto (\hat{W}\hat{W}^\top)^{\frac{1}{2}}$
- The representation learned is $\hat{D}$ (its range is the subspace of tasks)
- To learn a new task t', transfer $\hat{D}$

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{m} E\left(\langle w, x_{t'i} \rangle, y_{t'i}\right) + \gamma \left\langle w, \hat{D}^{-1} w \right\rangle$$

Alternating Minimization Algorithm

- Alternating minimization over W (supervised learning) and D (unsupervised "correlation" of tasks).

Initialization: set $D = \frac{I_{d\times d}}{d}$

while convergence condition is not true **do**

for $t = 1, \ldots, n,$ learn w_t *independently* by minimizing

$$\sum_{i=1}^{m} E(\langle w, x_{ti}\rangle, y_{ti}) + \gamma \langle w, D^{-1}w\rangle$$

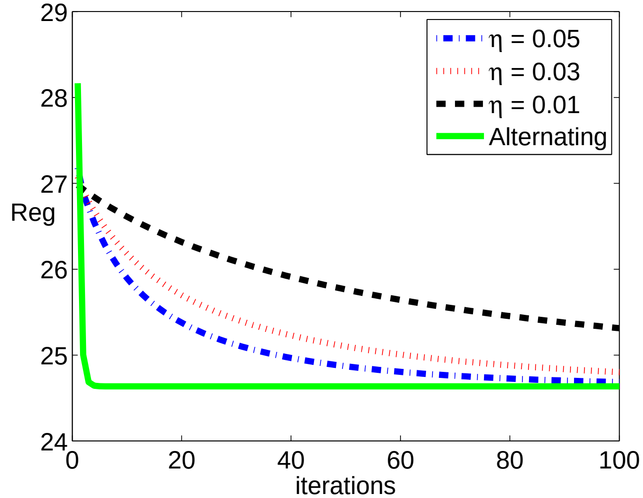
end for

set $D = \frac{(WW^\top)^{\frac{1}{2}}}{\mathrm{tr}(WW^\top)^{\frac{1}{2}}}$

end while

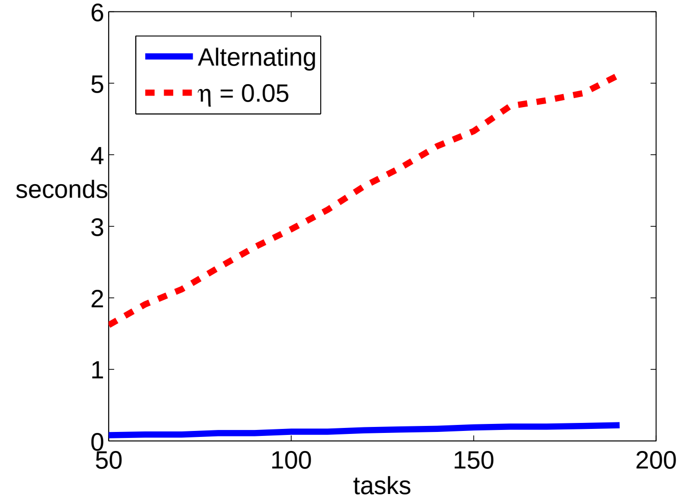
Alternating Minimization (contd.)

Reg. error



#iterations
(green = alternating)

secs.



#tasks
(blue = alternating)

- Compare computational cost vs. gradient descent ($\eta :=$ learning rate)

Connection to Rank Minimization

- Recent interest in the problem in *matrix factorization, statistics, compressed sensing* [*Cai et al. 2008, Fazel et al. 2001, Izenman 1975, Liu and Vandenberghe 2008, Srebro et al. 2005*]

- Regularization with the *rank*; relaxation with the *trace norm*

$$\min_{W \in \mathbb{R}^{d\times n}} \ \mathcal{E}(W) + \gamma \, \mathsf{rank}(W)$$

$$\min_{W \in \mathbb{R}^{d\times n}} \ \mathcal{E}(W) + \gamma \, \|W\|_{tr}^2$$

  Trace norm $\|W\|_{tr}$ is the sum of the singular values of W

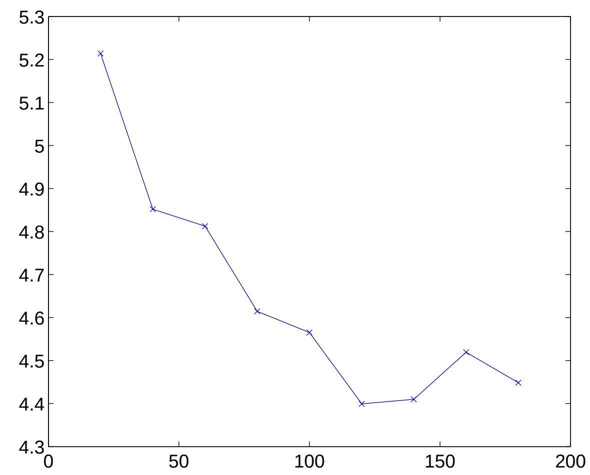
- Trace norm solution adequately recovers rank solution under conditions [*Candès and Recht 2008*] (for interpolation)

Experiment (Computer Survey)

- Consumers' ratings of products [Lenk et al. 1996]
- 180 persons (tasks)
- 8 PC models (training examples); 4 PC models (test examples)
- 13 binary input variables (RAM, CPU, price etc.) $+$ bias term
- Integer output in $\{0, \ldots, 10\}$ (likelihood of purchase)
- The square loss was used

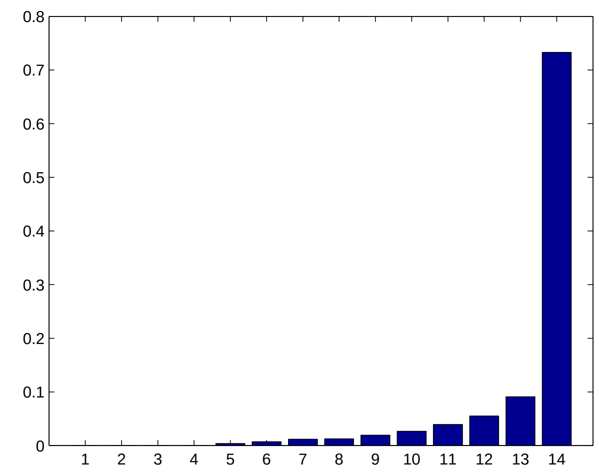
Experiment (Computer Survey)

Test error



#tasks

Eig(D)

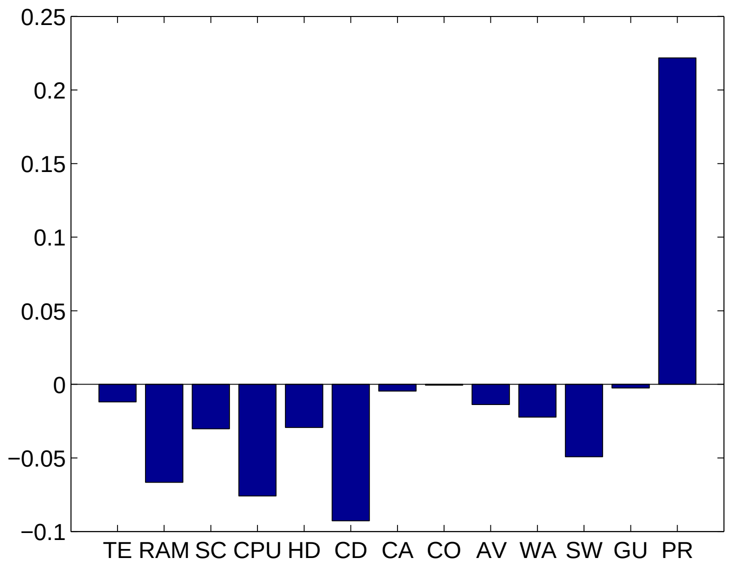


- Performance improves with more tasks
  (for learning tasks independently, error $= 16.53$)

- A single most important feature shared by all persons

Experiment (Computer Survey)

u_1



Method	RMSE
Alternating	1.93
Hierarchical Bayes [Lenk et al.]	1.90

- The most important feature weighs *technical characteristics* (RAM, CPU, CD-ROM) vs. *price*

Extensions

(1) Spectral regularization:

$$\min_{\substack{w_1,\dots,w_n \in \mathbb{R}^d \\ D \in \mathcal{D}}} \sum_{t=1}^{n} \sum_{i=1}^{m} E\left(\langle w_t, x_{ti}\rangle, y_{ti}\right) + \gamma \, \mathrm{tr}(W^\top F(D) W)$$

where F is a *spectral* matrix function:

$$F(U \Lambda U^\top) = U \; \mathrm{diag}[f(\lambda_1), ..., f(\lambda_d)] \; U^\top$$

(2) Learn a partition of tasks in K groups (subspaces):

$$\min_{D_1,\dots,D_K \succ 0} \sum_{t=1}^{n} \min_{w_t \in \mathbb{R}^d} \min_{k=1}^{K} \left\{ \sum_{i=1}^{m} E\left(\langle w_t, x_{ti}\rangle, y_{ti}\right) + \gamma \langle w_t, D_k^{-1} w_t\rangle + \mathrm{tr}(D_k) \right\}$$

Representer Theorems

- All previous formulations satisfy a *multi-task representer theorem*

$$\hat{w}_t = \sum_{s=1}^{n} \sum_{i=1}^{m} c_{si}^{(t)} x_{si} \qquad \forall\, t \in \{1, \ldots, n\} \tag{1}$$

Consequently, a nonlinear *kernel* can be used in the place of x

- *All tasks* are involved in this expression (unlike the single-task representer theorem $\Leftrightarrow$ Frobenius norm regularization)

- Generally, consider any problem of the form

$$\min_{w_1,\ldots,w_n \in \mathbb{R}^d} \sum_{t=1}^{n} \sum_{i=1}^{m} E\left(\langle w_t, x_{ti}\rangle, y_{ti}\right) + \Omega(W)$$

Representer Theorems (contd.)

- **Definitions:**

  $\mathbf{S}_+^n$ = the positive semidefinite cone
  The function $h : \mathbf{S}_+^n \to \mathbb{R}$ is matrix nondecreasing, if

$$h(A) \leq h(B) \qquad \forall\ A, B \in \mathbf{S}_+^n \quad \text{s.t. } A \preceq B$$

- **Theorem:** [Argyriou, Micchelli & Pontil 2008]
  Rep. thm. (1) holds if and only if there exists a *matrix nondecreasing* function $h : \mathbf{S}_+^n \to \mathbb{R}$ such that

$$\Omega(W) = h(W^\top W) \qquad \forall\ W \in \mathbb{R}^{d \times n}$$

Representer Theorems (contd.)

- **Theorem:** [Argyriou, Micchelli & Pontil 2008]
  The standard rep. thm. for *single-task learning*

  $$\hat{w} = \sum_{i=1}^{m} c_i x_i$$

  holds if and only if there exists a *nondecreasing* function $h : \mathbb{R}_+ \to \mathbb{R}$ such that

  $$\Omega(w) = h(\langle w, w \rangle) \qquad \forall w \in \mathbb{R}^d$$

- Completes previous results by [*Kimeldorf & Wahba, 1970, Schölkopf et al., 2001* etc.]

Connection to Learning the Kernel (LTK)

- General formulation

$$R(K) = \min_{c \in \mathbb{R}^m} \left\{ \sum_{i=1}^{m} E\big((Kc)_i, y_i\big) + \gamma \, \langle c, Kc \rangle \right\}$$

minimize R over a *convex set* $\mathcal{K}$

[*Lanckriet et al. 2004, Bach et al. 2004, Sonnenburg et al. 2006* etc.]

- If $E(\cdot, y)$ is convex then R is a convex function [Micchelli & Pontil 2005]

$$R(K) = \min_{v \in \mathbb{R}^m} \left\{ \sum_{i=1}^{m} E\big(v_i, y_i\big) + \gamma \, \langle v, K^{-1} v \rangle \right\}$$

A General Method for Learning the Kernel

- Convex set $\mathcal{K}$ is generated by *basic kernels*
- Example 1: *Finite set* of basic kernels (aka MKL)
- Example 2: *Linear* basic kernels ($\Leftrightarrow$ multi-task learning on a subspace)

$$B(x, x') = x^\top D x'$$

  where $D \succ 0, \mathrm{tr}(D) \leq 1$

- Example 3: Gaussian basic kernels

$$B(x, x') = e^{-(x-x')^\top \Sigma^{-1}(x-x')}$$

  where Σ belongs in a convex subset of the p.s.d. cone

A General Method for Learning the Kernel (contd.)

[*Argyriou, Micchelli & Pontil 2005*]

Initialization: Given an initial kernel $K^{(1)}$ in the convex set $\mathcal{K}$

while convergence condition is not true **do**

1. Compute $\hat{c} = \underset{c \in \mathbb{R}^m}{\operatorname{argmin}} \left\{ c^\top K_{\mathbf{x}}^{(t)} c + 4\gamma\, \mathcal{E}^*(c) \right\}$ (dual problem)

2. Find a basic kernel $\hat{B}$ maximizing $\hat{c}^\top B_{\mathbf{x}}\, \hat{c}$

3. Compute $K^{(t+1)}$ as the optimal convex combination of $\hat{B}$ and $K^{(t)}$

end while

- Always converges to an optimal kernel; however, step 2 is non-convex for e.g. Gaussian kernels (*but one-parameter Gaussians is solvable*)

Learning the Kernel in Semi-Supervised Learning

$$\max_{K \in \mathcal{K}} \min_{c \in \mathbb{R}^\ell} \left\{ \sum_{i=1}^{\ell} E^*(c_i, y_i) + \gamma \langle c, Kc \rangle \right\}$$

[Argyriou, Herbster & Pontil 2005]

- Here, $\mathcal{K} = \left\{ \sum_{i=1}^{N} \lambda_i (\mathbf{L}_i^+)_{labeled} : \lambda_i \geq 0, \sum_j \lambda_j = 1 \right\}$ where $\mathbf{L}_1, \ldots, \mathbf{L}_N$ are *Laplacians*.

LTK/MTL Connection to Sparsity

- **LTK:** feature space interpretation
  [*Bach et al. 2004, Micchelli & Pontil 2005*]

$$\min_{v_1,\ldots,v_N \in \mathbb{R}^m} \left\{ \sum_{i=1}^{m} E\left(\sum_{j=1}^{N} \langle v_j, \Phi_j(x_i) \rangle, y_i \right) + \gamma \left(\sum_{j=1}^{N} \|v_j\| \right)^2 \right\}$$

- Mixed L_1/L_2 norm; used in *group Lasso* and *Cosso* in statistics
  [*Antoniadis & Fan 2001, Bakin 1999, Grandvalet & Canu, 1999, Lin & Zhang 2003, Obozinski et al. 2006, Yuan & Lin 2006*]

- LTK: learns a small set of feature maps / sparse combination of kernels
  MTL: learns a small set of common features shared by all the tasks

Conclusion

- General framework for jointly learning *multiple tasks*, based on *matrix regularization*

- Use an *alternating algorithm* to learn tasks that lie on a *common subspace*; this algorithm is simple and efficient

- Necessary and sufficient conditions for *representer theorems* (in both the multi-task and single-task setting)

- Multi-task learning can be viewed as an instance of *learning combinations of infinite kernels*

- More generally, we can learn combinations of (finite or infinite) kernels with a *greedy incremental algorithm*

References

[R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. JMLR 2005]

[A. Argyriou, T. Evgeniou and M. Pontil. Multi-task feature learning. NIPS 2006.]

[A. Argyriou, C. A. Micchelli and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. COLT 2005]

[F. R. Bach, G. R. G. Lanckriet and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. ICML 2004]

[B. Bakker and T. Heskes. Task clustering and gating for Bayesian multi–task learning. JMLR 2003]

[J. Baxter. A model for inductive bias learning. JAIR 2000]

[R. Caruana. Multi–task learning. JMLR 1997]

References

[T. Evgeniou, C.A. Micchelli and M. Pontil. Learning multiple tasks with kernel methods. JMLR 2005]

[G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui and M. I. Jordan. Learning the kernel matrix with semidefinite programming. JMLR 2004]

[A. Maurer. Bounds for linear multi-task learning. JMLR 2006]

[C.A. Micchelli and M. Pontil. Learning the kernel function via regularization. JMLR 2005]

[C. S. Ong, A. J. Smola, R. C. Williamson. Learning the kernel with hyperkernels. JMLR 2005]

[R. Raina, A. Y. Ng and D. Koller. Constructing informative priors using transfer learning. ICML 2006]

[N. Srebro, J.D.M. Rennie and T.S. Jaakkola. Maximum-margin matrix factorization. NIPS 2004]