Sparse Learning Under Regularization Framework

Haiqin Yang

Supervisors:
Prof. Irwin King
Prof. Michael R. Lyu

Department of Computer Science & Engineering
The Chinese University of Hong Kong

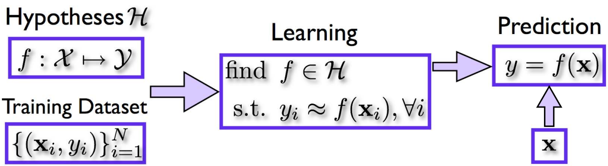
Outline

Supervised Learning

Data: N i.i.d. paired data sampled from $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$ as

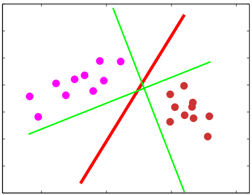
$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N, \quad \mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d, \quad y_i \in \mathcal{Y} \subseteq \mathbb{R}$$

Procedure:

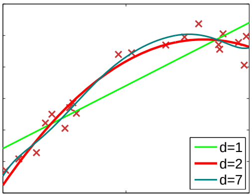


Tasks:

Classification



Regression



Regularization

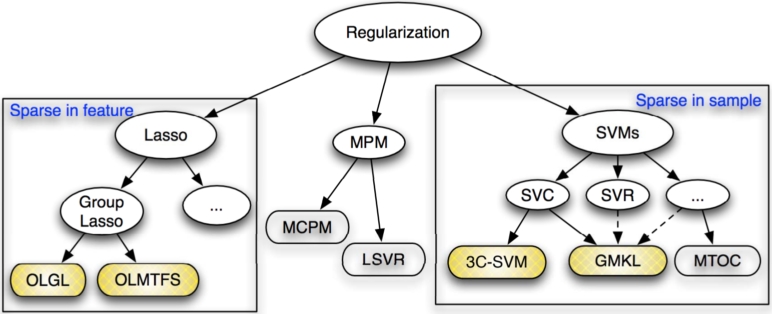
- Formulation

$$f^* = \arg\min_{f \in \mathcal{H}} \left(R[f] + C\mathcal{R}_{\mathcal{D}}^{\ell}[f] \right),$$

$R[f]$:	Regularization, measures complexity of f
$\mathcal{R}_{\mathcal{D}}^{\ell}[f]$:	Empirical risk, measured by square, hinge, etc.
$C \geq 0$:	Trade-off parameter

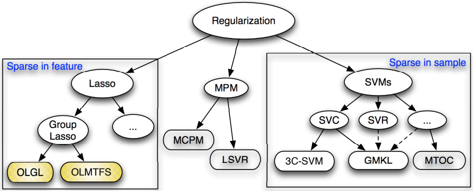
- Advantages
  - Controlling the functional complexity to avoid overfitting
  - Providing an intuitive and principled tool for learning from high-dimensional data
    - Lasso: Perform regression while selecting features
    - SVM: Regularization corresponds to maximum margin

Overview



- Sparse learning models under regularization
  - Sparse in feature level
  - Sparse in sample level
- Online learning
- Semi-supervised learning
- Multiple kernel learning (MKL)

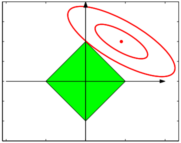
Sparse in feature level

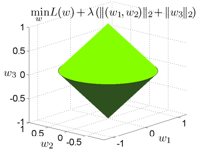


- Models

$$\text{Lasso:} \quad \min_{\mathbf{w}} \quad \|\mathbf{X}\mathbf{w} - \mathbf{Y}\|^2 + \lambda \|\mathbf{w}\|_1$$

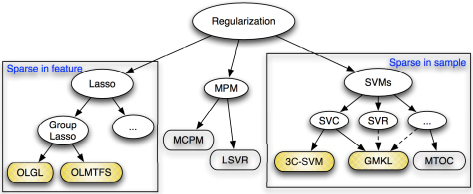
$$\text{Group Lasso:} \quad \min_{\mathbf{w}} \quad \|\mathbf{X}\mathbf{w} - \mathbf{Y}\|^2 + \lambda \sum_{g=1}^{G} \sqrt{d_g} \|\mathbf{w}^g\|_2$$





- Our contributions
  - **Online Learning for Group Lasso** (ICML'10)
  - **Online Learning for Multi-Task Feature Selection** (CIKM'10)

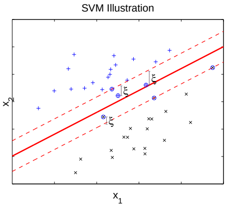
Sparse in sample level

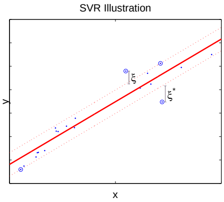


- SVM

$$\min_{\mathbf{w}} \quad C\sum_i [1 - y_i f_{\vartheta}(\mathbf{x}_i)]_+ + \frac{1}{2}\|\mathbf{w}\|^2 \Leftrightarrow \max_{\boldsymbol{\alpha}\in\mathcal{A}} \quad \mathbf{1}_N^\top \boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top \mathbf{K}(\boldsymbol{\alpha}\circ\mathbf{y})$$

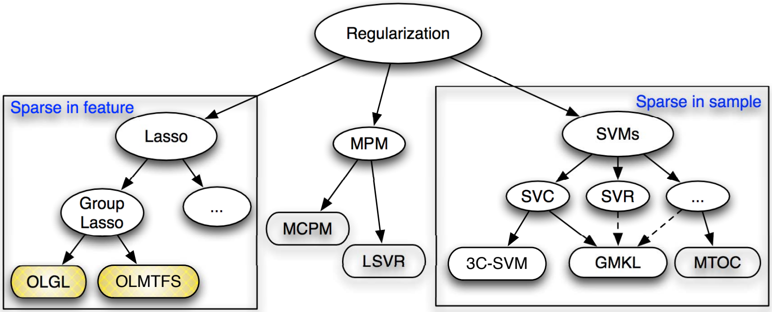
$$f(\mathbf{x}) = \sum_{i=1}^N y_i\alpha_i^\star K(\mathbf{x}_i,\mathbf{x}) + b^\star, \quad \text{most } \boldsymbol{\alpha}^\star \text{ are zeros}$$





- Our contributions
  - **Maximum Margin Semi-supervised Learning With Irrelevant Data** (TR'09)
  - **Efficient Sparse Generalized Multiple Kernel Learning** (TNN Revision)

First Part (Ch. 3, 4): Online Learning for Group Lasso and Multi-Task Feature Selection

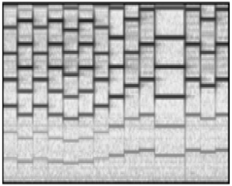


Motivation

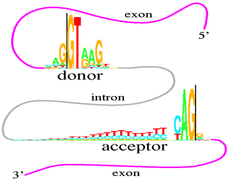
- Problems
  - Data come sequentially
  - Massive data
  - With redundant/irrelevant features
- Two senarios
  - Data contain group features
  - Multiple related tasks share common features

Ch. 3: Online Learning for Group Lasso

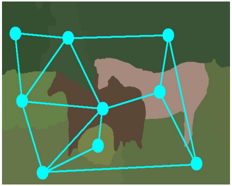
- Applications with group structure



McAuley et al., 2005



Meier et al., 2008



Harchaoui & Bach, 2007

- Group features
  - Continuous features represented by k-th order expansions
    $x_1 \Rightarrow \mathbf{x}_1 = [x_1, x_1^2, \ldots, x_1^k]$
  - Categorical features represented a group of dummy variables
    $x_2 \Rightarrow \mathbf{x}_2 = [x_{21}, x_{22}, \ldots, x_{2m}]$
- Problems
  - Some features are redundant or irrelevant
  - Data come in sequence
  - Data are large in volume

Ch. 3: Online Learning for Group Lasso

- Related work
  - Group lasso and its extensions (Yuan & Lin, 2006; Meier et al., 2008; Roth & Fischer, 2008; Jacob et al., 2009; etc.)
  - Online learning algorithms (Shalev-Shwartz & Singer, 2006; Zinkevich, 2003; Bottou & LeCun, 2003; Langford et al., 2009; Duchi & Singer, 2009; Xiao, 2009)

    Batch learned algorithms cannot solve the above problems mentioned!
- Our contributions
  - The first online learning framework for the group lasso
  - Easy implementation: three lines of main codes
  - Efficiency in both time complexity and memory cost, $\mathcal{O}(d)$
  - Sparsity in both the group level and the individual feature level
  - Easy extension to group lasso with overlap and graph lasso

Ch. 4: Online Learning for Multi-Task Feature Selection

- Observations: Related tasks contain helpful information;
  Redundant/irrelevant features exist
  - **Gene selection** from microarray data in related diseases
    - Variables: Gene expression coefficients corresponding to the amount of mRNA in a patient's sample (e.g., tissue biopsy)
    - Tasks: Distinguish healthy from unhealthy for different diseases
    - Problems: few samples ($<$ 100's), large variables ($>$1000's)
  - **Text categorization** from documents in multiple related categories
    - Features represented by a vector of vocabulary on word frequency counts
    - Vocabulary: $>$ 10000's words
    - Tasks: 1) Detecting spam-emails from persons with same interests;
      2) Automatic classifying related web page categories
- Related work
  - A generalized L_1-norm single-task regularization (Argyriou et. al. 2008)
  - Mixed norms of L_1, L_2, and L_∞ norms (Obozinski et. al. 2009)
  - Nesterov's method on MTFS (Liu et. al. 2009)
  - $L_{0,0}$-regularization based on MIC (Dhillon et. al. 2009)

Ch. 4: Online Learning for Multi-Task Feature Selection

- Problems
  - Features among tasks are redundant or irrelevant
  - Data come in sequence
  - Data are large in volume
- Our contributions
  - The first online learning framework for multi-task feature selection
  - Easy implementation: three lines of main codes
  - Efficiency in both time complexity and memory cost, $\mathcal{O}(d \times Q)$
  - Find important features and important tasks that dominating the features
  - Easily extend to nonlinear models

Models

Lasso: A shrinkage and selection method for linear regression

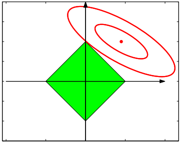
$$\min_{\mathbf{w}} \quad \|\mathbf{X}\mathbf{w}-\mathbf{Y}\|^2+\lambda\|\mathbf{w}\|_1$$

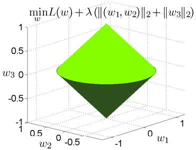
Group Lasso: Find important explanatory factors in a grouped manner

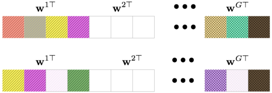
$$\min_{\mathbf{w}} \quad \|\mathbf{X}\mathbf{w}-\mathbf{Y}\|^2+\lambda\sum_{g=1}^{G}\sqrt{d_g}\|\mathbf{w}^g\|_2$$

Sparse Group Lasso: Yield sparse solutions in the selected group

$$\min_{\mathbf{w}} \quad \|\mathbf{X}\mathbf{w}-\mathbf{Y}\|^2+\lambda\sum_{g=1}^{G}(\sqrt{d_g}\|\mathbf{w}^g\|_2+r_g\|\mathbf{w}^g\|_1)$$







Summary

Model framework

$$\min_{\mathbf{w}} \quad \sum_{i=1}^{N} \ell(\mathbf{w},\ \mathbf{z}_i) + \Omega_\lambda(\mathbf{w})$$

$\ell(\cdot,\cdot)$: Loss function, e.g., square loss, logit loss, etc.
$\Omega_\lambda(\cdot)$: Regularization on the weight

Favorable properties

- Obtaining sparse solution
- Performing feature selection and classification/regression simultaneously
- Improving classification/regression performance

Multi-Task Feature Selection Models

- **Data:** i.i.d. observations: $\mathcal{D} = \bigcup_{q=1}^{Q} \mathcal{D}_q$

  $\mathcal{D}_q = \{\mathbf{z}_i^q = (\mathbf{x}_i^q, y_i^q)\}_{i=1}^{N_q}$ sampled from $\mathcal{P}_q$, $q = 1, \ldots, Q$

  $\mathbf{x} \in \mathbb{R}^d$–input variable, $y \in \mathbb{R}$–response

- **Model:** $f_q(\mathbf{x}) = \mathbf{w}^{q\top}\mathbf{x}, \quad q = 1, \ldots, Q$

- **Objective:** $\min_{\mathbf{W}} \quad \sum_{q=1}^{Q} \frac{1}{N_q} \sum_{i=1}^{N_q} \ell^q(\mathbf{W}_{\bullet q}, \mathbf{z}_i^q) + \Omega_\lambda(\mathbf{W})$

  $\mathbf{W} = \left(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^Q\right) = (\mathbf{W}_{\bullet 1}, \ldots, \mathbf{W}_{\bullet Q}) = \left(\mathbf{W}_{1\bullet}^\top, \ldots, \mathbf{W}_{d\bullet}^\top\right)^\top$

- Regularization

  iMTFS: $\Omega_\lambda(\mathbf{W}) = \lambda \sum_{q=1}^{Q} \|\mathbf{W}_{\bullet q}\|_1 = \lambda \sum_{j=1}^{d} \left\|\mathbf{W}_{j\bullet}^\top\right\|_1$

  aMTFS: $\Omega_\lambda(\mathbf{W}) = \lambda \sum_{j=1}^{d} \left\|\mathbf{W}_{j\bullet}^\top\right\|_2$

  MTFTS: $\Omega_{\lambda,\mathbf{r}} = \lambda \sum_{j=1}^{d} \left(r_j \left\|\mathbf{W}_{j\bullet}^\top\right\|_1 + \left\|\mathbf{W}_{j\bullet}^\top\right\|_2\right)$

$$
\overset{\textbf{iMTFS}}{\begin{pmatrix} x & 0 & 0 & x & x \\ 0 & x & x & x & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x & 0 & x & x & x \end{pmatrix}},
\overset{\textbf{aMTFS}}{\begin{pmatrix} x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x & x & x & x & x \end{pmatrix}},
\overset{\textbf{MTFTS}}{\begin{pmatrix} x & 0 & x & x & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x & 0 & x & x \end{pmatrix}}
$$

Online Learning Algorithm Framework for Group Lasso

Initialization: $\mathbf{w}_1 = \mathbf{w}_0$, $\bar{\mathbf{u}}_0 = \mathbf{0}$

for $t = 1, 2, 3, \ldots$

1. Compute the subgradient on $\mathbf{w}_t$, $\mathbf{u}_t \in \partial\, l_t$
2. Update the average subgradient $\bar{\mathbf{u}}_t$:
   $$\bar{\mathbf{u}}_t = \frac{t-1}{t}\bar{\mathbf{u}}_{t-1} + \frac{1}{t}\mathbf{u}_t$$
3. Calculate the next iteration $\mathbf{w}_{t+1}$:
   $$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \Upsilon(\mathbf{w}) \triangleq \left\{\bar{\mathbf{u}}_t^\top \mathbf{w} + \Omega_\lambda(\mathbf{w}) + \frac{\gamma}{\sqrt{t}} h(\mathbf{w})\right\}$$

end for

Remarks

- Motivated by the dual averaging method for Lasso (Xiao, 2009)
- $h(\mathbf{w})$: Make the new search point in the vincinity
- FOBOS (Duchi & Singer, 2009):
  $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{\frac{1}{2}\|\mathbf{w} - (\mathbf{w}_t - \eta_t \mathbf{u}_t)\|^2 + \eta_t \Omega(\mathbf{w})\right\}$
- Overlapped groups or graph lasso

Online Learning Algorithm Framework for MTFS

Initialization: $\mathbf{W}_1 = \mathbf{W}_0$, $\bar{\mathbf{G}}_0 = \mathbf{0}$

for $t = 1, 2, 3, \ldots$

1. Compute the subgradient on $\mathbf{W}_t$, $\mathbf{G}_t \in \partial\, l_t$
2. Update the average subgradient $\bar{\mathbf{G}}_t$:
   $$\bar{\mathbf{G}}_t = \frac{t-1}{t}\bar{\mathbf{G}}_{t-1} + \frac{1}{t}\mathbf{G}_t$$
3. Calculate the next iteration $\mathbf{W}_{t+1}$:
   $$\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}} \Upsilon(\mathbf{W}) \triangleq \left\{\bar{\mathbf{G}}_t^\top \mathbf{W} + \Omega_\lambda(\mathbf{W}) + \frac{\gamma}{\sqrt{t}} h(\mathbf{W})\right\}$$

end for

Remarks

- $\mathbf{W}$: becomes a matrix for MTFS
- Original formulation is in linear case; it can be extended to non-linear case easily

Updating Rules for Online Group Lasso

Group Lasso: $\Omega_\lambda(\mathbf{w}) = \lambda \sum_{g=1}^{G} \sqrt{d_g}\|\mathbf{w}^g\|_2$, $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma}\left[1 - \frac{\lambda\sqrt{d_g}}{\|\bar{\mathbf{u}}_t^g\|_2}\right]_+ \cdot \bar{\mathbf{u}}_t^g$$

Sparse Group Lasso: $\Omega_{\lambda,\mathbf{r}}(\mathbf{w}) = \lambda \sum_{g=1}^{G}\left(\sqrt{d_g}\|\mathbf{w}^g\|_2 + r_g\|\mathbf{w}^g\|_1\right)$, $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma}\left[1 - \frac{\lambda\sqrt{d_g}}{\|\mathbf{c}_t^g\|_2}\right]_+ \cdot \mathbf{c}_t^g, \quad c_t^{g,j} = \left[|\bar{u}_t^{g,j}| - \lambda r_g\right]_+ \cdot \operatorname{sign}\left(\bar{u}_t^{g,j}\right)$$

Enhanced Sparse Group Lasso: $\Omega_{\lambda,\mathbf{r}}(\mathbf{w}) = \lambda \sum_{g=1}^{G}\left(\sqrt{d_g}\|\mathbf{w}^g\|_2 + r_g\|\mathbf{w}^g\|_1\right)$, $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \rho\|\mathbf{w}\|_1$

$$\mathbf{w}_{t+1}^g = -\frac{\sqrt{t}}{\gamma}\left[1 - \frac{\lambda\sqrt{d_g}}{\|\tilde{\mathbf{c}}_t^g\|_2}\right]_+ \cdot \tilde{\mathbf{c}}_t^g, \quad \tilde{c}_t^{g,j} = \left[|\bar{u}_t^{g,j}| - \lambda r_g - \frac{\gamma\rho}{\sqrt{t}}\right]_+ \cdot \operatorname{sign}\left(\bar{u}_t^{g,j}\right)$$

Efficiency: $\mathcal{O}(d)$ in memory cost and time complexity

Updating Rules for Online MTFS

Define: $h(\mathbf{W}) = \frac{1}{2}\|\mathbf{W}\|_F^2$

- **iMTFS**: For $i = 1, \ldots, d$ and $q = 1, \ldots, Q$,

$$(W_{i,q})_{t+1} = -\frac{\sqrt{t}}{\gamma}\left[|(\bar{G}_{i,q})_t| - \lambda\right]_+ \cdot \mathrm{sign}\left((\bar{G}_{i,q})_t\right).$$

- **aMTFS**: For $j = 1, \ldots, d$,

$$(\mathbf{W}_{j\bullet})_{t+1} = -\frac{\sqrt{t}}{\gamma}\left[1 - \frac{\lambda}{\|(\bar{\mathbf{G}}_{j\bullet})_t\|_2}\right]_+ \cdot (\bar{\mathbf{G}}_{j\bullet})_t.$$

- **MTFTS**: For $j = 1, \ldots, d$,

$$(\mathbf{W}_{j\bullet})_{t+1} = -\frac{\sqrt{t}}{\gamma}\left[1 - \frac{\lambda}{\|(\bar{\mathbf{U}}_{j\bullet})_t\|_2}\right]_+ \cdot (\bar{\mathbf{U}}_{j\bullet})_t,$$

  where the q-th element of $(\bar{\mathbf{U}}_{j\bullet})_t$ is calculated by

$$(\bar{U}_{j,q})_t = \left[|(\bar{G}_{j,q})_t| - \lambda r_j\right]_+ \cdot \mathrm{sign}\left((\bar{G}_{j,q})_t\right),\ q = 1, \ldots, Q.$$

Efficiency: $\mathcal{O}(d \times Q)$ in memory cost and time complexity

Theoretical Results

Average regret for group lasso

$$\bar{R}_T(\mathbf{w}) := \frac{1}{T}\sum_{t=1}^{T}\left(\Omega_\lambda(\mathbf{w}_t) + l_t(\mathbf{w}_t)\right) - S_T(\mathbf{w})$$

Average regret for MTFS

$$\bar{R}_T(\mathbf{w}) := \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{T}\sum_{t=1}^{T}\left(\Omega_\lambda(\mathbf{W}_t) + l_t(\mathbf{W}_t)\right) - S_T(\mathbf{W})$$

Theoretical bounds

$\bar{R}_T \sim \mathcal{O}(1/\sqrt{T})$

Experimental Setup for Online Group Lasso

Data

- ★ Synthetic data
- ★ Realworld data for gene finding

Comparison algorithms

- ★ Lasso
- ★ GL: Group Lasso
- ★ L_1-RDA: Regularized Dual Averaging for L_1-regularization
- ★ DA-GL: Dual Averaging for group lasso
- ★ DA-SGL: Dual Averaging for sparse group lasso

Platform

- ★ PC with 2.13 GHz dual-core CPU
- ★ Batch-mode algorithms: R-package, grplasso
- ★ Online-mode algorithms: Matlab

Synthetic data

Data generation scheme

Sparsity on both group and element levels

- ✓ $\mathbf{w} \in \mathbb{R}^{100}$, $w_i = \{0, \pm 1\}$, $G = 10$, $\#\,\mathrm{NNZ} = \{10, 8, 6, 4, 2, 1, 0, 0, 0, 0\}$
- ✓ $\mathbf{x}_i = L\mathbf{v}_i$, $y_i = \mathrm{sign}\,(\mathbf{w}^\top \mathbf{x}_i + \epsilon)$, $\epsilon \sim \mathcal{N}(0, 4^2)$, $i = 1, \ldots, N_{tr}$
  L: Cholesky decomposition of the correlation matrix, $\Sigma^g_{i,j} = 0.2^{|i-j|}$
- ✓ $N_{tr} = N_{ts} = \{25, 50, 100, 500, 1000, 5000, 10^4, 10^5\}$

Measurement

- ✓ Accuracy
- ✓ Average F1 score: Measure true weight
- ✓ The larger the better

Parameters

- ✓ λ: $\lambda_{\max} * \{0.5, 0.2, 0.1, 0.05\}$
- ✓ $\gamma = L/D$
- ✓ DA-SGL: $r_g = 1$

Synthetic Data Results

Accuracy

- ★ Accuracies increase with the increase of the number of training samples
- ★ DA-SGL achieves the best accuracy, especially when the number of training sample is small
- ★ DA-GL achieves slightly worse results than the DA-SGL and the GL when the number of training sample is large
- ★ Two batch-trained algorithms achieve nearly the same accuracy when the number of training samples is large

	Lasso	GL	L_1-RDA	DA-GL	DA-SGL
25	54.2 ± 14.1	54.2± 11.4	56.6± 9.9	57.0± 11.6	**57.6**± 11.0
50	58.2 ± 7.7	60.0± 6.3	59.5± 6.9	**60.9**± 6.2	**60.9**± 6.0
100	62.7 ± 5.5	64.0± 5.1	61.7± 4.8	64.5± 4.1	**64.6**± 4.5
500	75.6 ± 2.4	75.7± 2.3	66.2± 3.0	74.8± 2.3	**75.9**± 2.2
1000	77.7 ± 1.5	77.8± 1.5	65.9± 2.0	76.3± 1.4	**77.9**± 1.6
5000	**79.4** ± 0.4	**79.4**± 0.3	67.8± 1.5	78.2± 0.6	**79.4**± 0.8
10^4	**80.0** ± 0.2	**80.0**± 0.1	68.0± 1.3	79.8± 0.3	**80.0**± 0.1
10^5	**80.1** ± 0.1	**80.1**± 0.1	69.7± 1.2	79.9± 0.1	**80.1**± 0.1

Synthetic Data Results

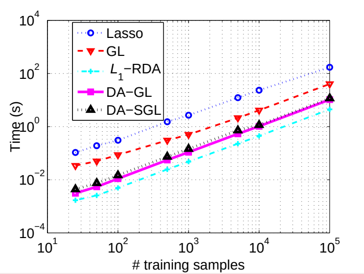
Averaged F1 score

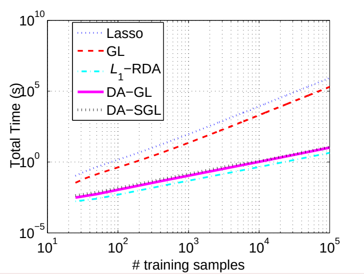
- ✓ DA-SGL outperforms all other four algorithms
- ✓ The DA-SGL combines both the advantages of the lasso and the GL
- ✓ GL and the DA-GL got similar average F1 scores

	Lasso	GL	L_1-RDA	DA-GL	DA-SGL
25	23.6± 8.5	37.3± 13.6	35.6± 6.3	37.2± 3.0	**37.9**± 4.5
50	35.0± 9.3	**49.8**± 6.0	39.7± 6.5	49.7± 3.0	**49.8**± 4.9
100	47.0± 7.2	**57.4**± 2.4	46.5± 9.7	57.1± 2.7	**57.4**± 5.9
500	65.0± 2.5	65.5± 2.1	63.6± 9.7	65.2± 6.8	**81.9**± 5.3
1000	70.1± 2.4	67.2± 2.1	64.9± 8.7	67.2± 4.7	**87.3**± 4.3
5000	88.2± 2.4	68.2± 2.0	66.8± 8.0	68.3± 2.9	**93.7**± 2.5
10^4	94.1± 2.3	69.1± 1.8	67.4± 5.5	68.4± 2.5	**94.2**± 2.1
10^5	**97.3**± 2.2	69.5± 1.7	68.1± 5.1	68.7± 2.3	**97.3**± 2.1

Efficiency

Running time





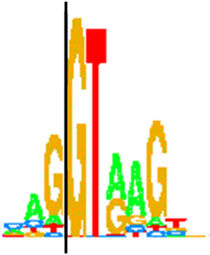
Data loading time

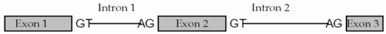
About 1 hour for loading large datasets in R

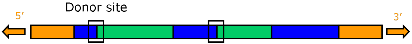
Splice Site Detection

Description

- ★ Splice sites (ss): Regions between coding (exons) and non-coding (introns) DNA segments
- ★ Donor splice site: 5' end of an intron
- ★ Structure: Last three bases of the exon+first six bases of the intron=N_3GTN_4
- ★ Real splice sites: U2 type spliceosom
- ★ Decoy splice sites: Non-splice sites







Experimental Setup and Results on Splice Site Detection

Setup

- ✓ Following (Meier et al., 2008): Training set (5,610/5,610), validation set (2,805/59,804), test set (4,208/89,717)
- ✓ Features: N_3GTN_4 (remove consensus "GT", length = 7) up to 2nd order interaction, $d = 2604 \left(\binom{7}{1} * 4 + \binom{7}{2} * 4^2 + \binom{7}{3} * 4^3 \right)$
- ✓ λ is varying from $[0.01, 10]$
- ✓ Algorithm parameter: γ is tuned
- ✓ DA-SGL: $r_g = \sqrt{d_g}$

Measurement–Maximal correlation coefficient

- $\rho_{\max} = \max\{\rho_\tau | \tau \in (0, 1)\}$
- The larger the better

Results

Accuracy

% Non-zero	L1-RDA	DA-GL	DA-SGL
10	0.5632	**0.5656**	**0.5656**
40	0.6056	0.6071	**0.6082**
60	0.6481	0.6496	**0.6501**
80	0.6494	**0.6520**	**0.6520**

Group lasso in (Meier et al., 2008): **0.6593**

Time issue

- Online algorithms: $\approx 10^3$ seconds (running time)
- Batch group lasso: $\approx 4 \times 10^3$ seconds (running time)

Experimental Setup for Online MTFS

Data

- ★ School data
- ★ Computer survey data

Comparison algorithms

- ★ iMTFS
- ★ aMTFS
- ★ DA-iMTFS
- ★ DA-aMTFS
- ★ DA-MTFTS

Platform

- ★ PC with 2.13 GHz dual-core CPU
- ★ Batch-mode algorithms: Matlab
- ★ Online-mode algorithms: Matlab

School Data

Description

- **Objective:** Predict exam scores
- **Data:** Exam scores of 15,362 students from 139 secondary schools in London during the years 1985, 1986, and 1987, $Q = 139$
- **Features:** Year of the exam (YR), 4 school-specific and 3 student-specific features, $d = 27$

Setup

- Evaluation: Explained variance (R^2) $1 - \frac{SS_{\text{err}}}{SS_{\text{tol}}}$, the larger the better
- Loss: Square loss
- Parameters setting: Cross validation (hierarchical search and grid search)

School Data Results

Accuracy

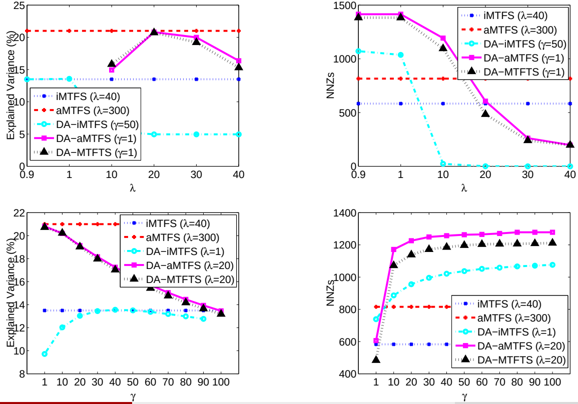
- Learning multiple tasks simultaneously can gain over 50% improvement than learning the task individually
- Online learning algorithms attain (nearly) the same accuracies as batch-trained algorithms
- DA-MTFTS attains the same accuracy as DA-aMTFS with fewer NNZs

Method	Explained variance	NNZs	Parameters
aMTFS	**21.0**±1.7	815.5±100.6	$\lambda = 300$
iMTFS	13.5±1.8	583.0±16.6	$\lambda = 40$
DA-aMTFS	**20.8**±1.8	605.8±180.3	$\lambda = 20, \gamma = 1$, ep=120
DA-MTFTS	**20.8**±1.9	483.7±130.7	$\lambda = 20, \gamma = 1$, ep=120
DA-iMTFS	13.5±1.8	1037.1±21.4	$\lambda = 1, \gamma = 50$, ep=120

Effect of λ and γ

Results

- ✓ NNZs decreases as λ increases
- ✓ NNZs increases as γ increases
- ✓ Fewer NNZs in DA-MTFTS than DA-aMTFS



Conjoint Analysis

Description

- **Objective:** Predict rating by estimating respondents' partworths vectors
- **Data:** Ratings on personal computers of 180 students for 20 different PC, $Q = 180$
- **Features:** Telephone hot line (TE), amount of memory (RAM), screen size (SC), CPU speed (CPU), hard disk (HD), CDROM/multimedia (CD), cache (CA), color (CO), availability (AV), warranty (WA), software (SW), guarantee (GU) and price (PR); $d = 14$

Setup

- Evaluation: Root mean square errors (RMSEs)
- Loss: Square loss
- Parameters setting: Cross validation (hierarchical and grid search)

Conjoint Analysis Results

Accuracy

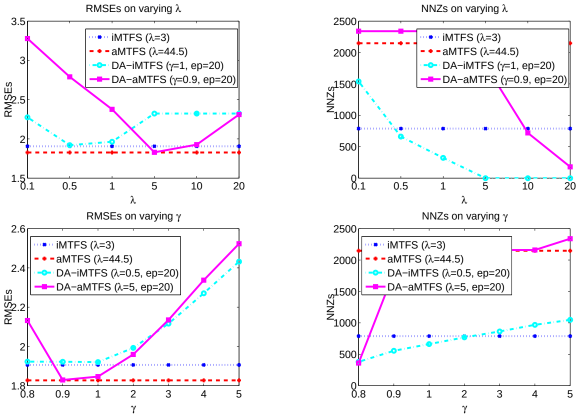
- Learning partworths vectors across respondents can help to improve the performance
- Online learning algorithms attain nearly the same accuracies as batch-trained algorithms

Method	RMSEs	NNZs	Parameters
aMTFS	**1.82**	2148	$\lambda = 44.5$
iMTFS	1.91	789	$\lambda = 3$
DA-aMTFS	2.04	540	$\lambda = 20.0, \gamma = 0.9$, ep=1
DA-aMTFS	**1.83**	1800	$\lambda = 5, \gamma = 0.9$, ep=20
DA-iMTFS	2.43	199	$\lambda = 2.0, \gamma = 2.0$, ep=1
DA-iMTFS	1.92	662	$\lambda = 0.5, \gamma = 1.0$, ep=20

Effect of λ and γ

Results

- ✓ NNZs decreases as λ increases
- ✓ NNZs increases as γ increases



Efficiency

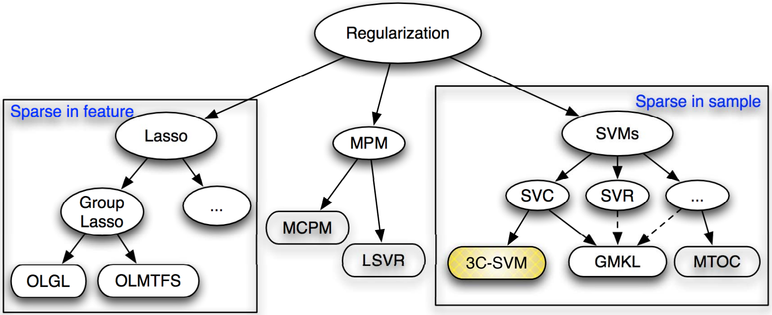
Time cost

- School Data
  - aMTFTS: 1.30s
  - DA-MTFTS: 0.99s
- Conjoint Analysis
  - iMTFTS: 0.326s
  - aMTFTS: 0.162s
  - DA-iMTFS: 0.08s
  - DA-aMTFS: 0.07s

Summary

- A novel online learning algorithm framework for group lasso & multi-task feature selection
- Apply this framework for several group lasso and multi-task feature selection models
- Provide closed-form solutions to update the models
- Provide the convergence rate of the average regret
- Experimental results demonstrate the proposed algorithms in both efficiency and effectiveness

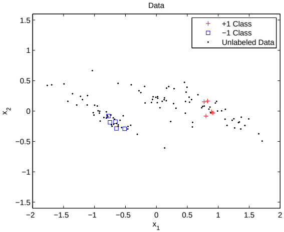
Second Part (Ch. 5): Tri-Class Support Vector Machine



Ch. 5: Maximum Margin Semi-supervised Learning With Irrelevant Data

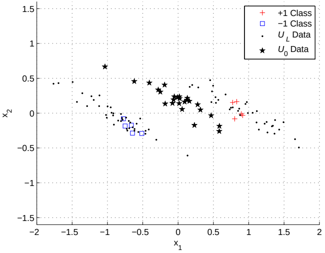
- Scenarios: Web documents categorization–Classify "sports news" vs. "financial news"; Digit recognition–Distinguish "5" vs. "8"
- Problems
  - Labeling data is costly and time consuming
  - Many unlabeled data are easy to collect and may provide useful information
- Solution: To learn from both labeled and unlabeled data simultaneously!
- Significance of SSL: Close to natural (human or animal) learning

Assumption



- Related work
  - EM (Nigam et al. 2000), Co-training (Blum & Mitchell, 1998), Transductive SVM (Joachims 1999; Collobert et al. 2006), Graph-based methods (Argyriou et al. 2006; Zhou et al. 2003; Zhu et al. 2003; Belkin et al. 2006)
- Previous assumption: unlabeled data are from the same distribution as the labeled data.
- Usual situation: unlabeled data may be a mixture of relevant and irrelevant data

Setup



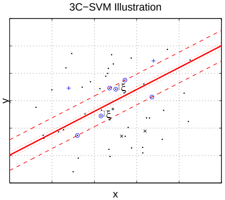
- $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$
  $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, $y_i \in \{-1, 0, 1\}$
- $\mathcal{U} = \mathcal{U}_{\mathcal{L}} \cup \mathcal{U}_0 = \{\mathbf{x}_i\}_{i=1}^{U}$
- **Objective:** seek $f_{\vartheta}(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$ with $\vartheta = (\mathbf{w}, b)$ to separate the binary class data correctly with the help of (mixed) unlabeled data

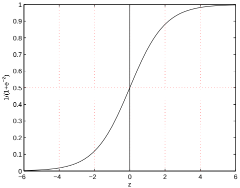
Model

- **Objective function:**

$$\min_{\vartheta} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}} r_i \ell_L(f_{\vartheta}(\mathbf{x}_i), y_i) + \sum_{\mathbf{x}_i \in \mathcal{U}} r_i \ell_U(f_{\vartheta}(\mathbf{x}_i)),$$

- **Facts:** if $f_{\vartheta}(\mathbf{x}_i) \gg 0$, more confident on $+1$-class
  if $f_{\vartheta}(\mathbf{x}_i) \ll 0$, more confident on -1-class
- **Principle:** rely more on relevant data,
  ignore irrelevant data





Model

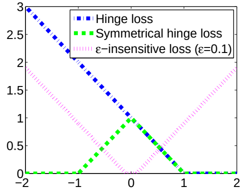
- **Objective function:**

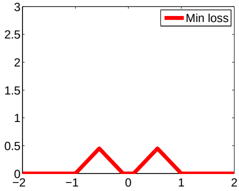
$$\min_{\vartheta} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \overbrace{\sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\vartheta}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_{\varepsilon}(f_{\vartheta}(\mathbf{x}_i))}^{\text{Loss on labeled data}}$$

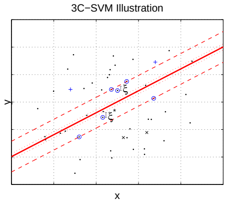
$$+ \overbrace{\sum_{\mathbf{x}_i \in \mathcal{U}} r_i \min\{H_1(|f_{\vartheta}(\mathbf{x}_i)|), I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_i)|)\}}^{\text{Loss on unlabeled data}} .$$

$$H_1(u) = \max\{0,\, 1-u\}, \quad I_{\varepsilon}(u) = \max\{0,\, |u| - \varepsilon\}$$

- **Illustration:**



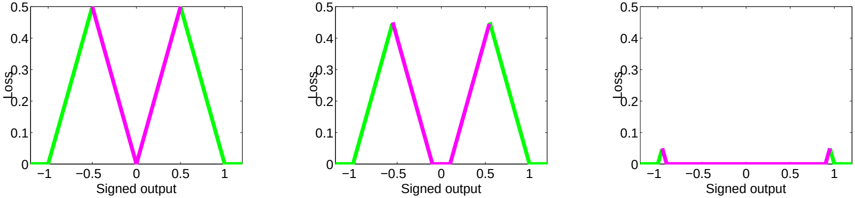




Model Generalization

- **Illustration:** $L_{\min}(u) = \min\{\max\{0, 1 - |u|\}, \max\{0, |u| - \varepsilon\}\}$

$\varepsilon = 0$ $\qquad$ $\varepsilon = 0.1$ $\qquad$ $\varepsilon = 0.9$



- **Model relationship:**

3C-SVM

$\mathcal{L}$	-1	0	1
$\mathcal{U}$	-1	0	1

SVM

$\mathcal{L}$	-1		1
$\mathcal{U}$			

S^{3}VM

$\mathcal{L}$	-1		1
$\mathcal{U}$	-1		1

$\mathcal{U}$-SVM

$\mathcal{L}$	-1	0	1
$\mathcal{U}$			

Theorem: How unlabeled irrelevant data help?

Objective function:

$$\min_{\vartheta} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_\vartheta(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_\varepsilon(f_\vartheta(\mathbf{x}_i))$$
$$+ \sum_{\mathbf{x}_i \in \mathcal{U}} r_i \min\{H_1(|f_\vartheta(\mathbf{x}_i)|), I_\varepsilon(|f_\vartheta(\mathbf{x}_i)|)\}.$$

3C-SVM with $r_i = \infty$ for unlabeled data and $\varepsilon = 0$

Unlabeled data $\mathbf{x}_j$ satisfies

(a) $|\mathbf{w}^T\phi(\mathbf{x}_j) + b| \geq 1 \Rightarrow$ data lie on or out of the margin gap,

or

(b) $\mathbf{w}^T\phi(\mathbf{x}_j) + b = 0 \Rightarrow \mathbf{w}^T(\phi(\mathbf{x}_j) - \phi(\mathbf{x}_0)) = 0$, $\mathbf{x}_j, \mathbf{x}_0 \in \mathcal{U}_0$

Removing Min-Terms and Absolute Values

$$\min_{\vartheta} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{\mathbf{x}_i \in \mathcal{L}_{\pm 1}} r_i H_1(y_i f_{\vartheta}(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in \mathcal{L}_0} r_i I_{\varepsilon}(f_{\vartheta}(\mathbf{x}_i))$$

$$+ \sum_{\mathbf{x}_{k+L} \in \mathcal{U}} r_{k+L} \overbrace{\left(\underbrace{H_1(|f_{\vartheta}(\mathbf{x}_i)| + D(1-d_k))}_{Q_1} + \underbrace{I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_i)| - Dd_k)}_{Q_2} \right)}^{\min\{H_1(|f_{\vartheta}(\mathbf{x}_i)|), I_{\varepsilon}(|f_{\vartheta}(\mathbf{x}_i)|)\}}$$

- **Integer programming:** $\begin{cases} d_k = 0 \Rightarrow Q_1 = 0 \\ d_k = 1 \Rightarrow Q_2 = 0 \end{cases}$
- $H_1(|u| + a)$: Introducing non-convexity, solved by ramploss $H_{1-a}(u) - H_{\kappa}(u) + H_{1-a}(-u) - H_{\kappa}(-u)$
- $I_{\varepsilon}(|u| - a) = H_{-\varepsilon-a}(-u) + H_{-\varepsilon-a}(u)$
- Absolute terms are removed by introducing auxiliary labels

Concave-Convex Procedure

- **Objective function:** $Q^{\kappa}(\boldsymbol{\vartheta}, \mathbf{d}) = Q^{\kappa}_{vex}(\boldsymbol{\vartheta}, \mathbf{d}) + Q^{\kappa}_{cav}(\boldsymbol{\vartheta})$
- Each step

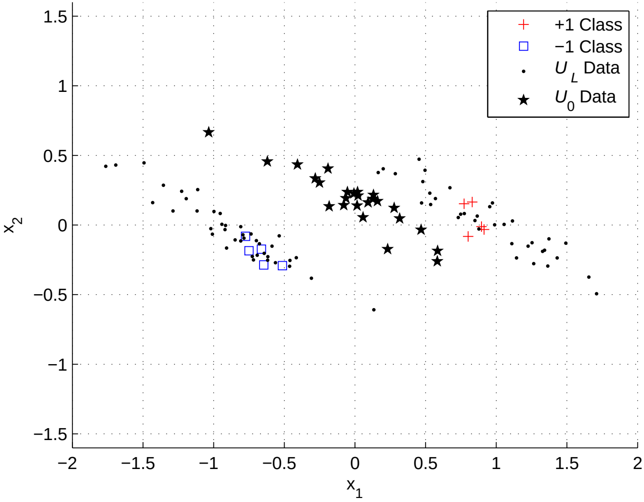
$$\boldsymbol{\vartheta}^{t+1} = \arg\min_{\boldsymbol{\vartheta}} \left(Q^{\kappa}_{vex}(\boldsymbol{\vartheta}, \mathbf{d}^t) + \frac{\partial Q^{\kappa}_{cav}(\boldsymbol{\vartheta}^t)}{\partial \boldsymbol{\vartheta}} \cdot \boldsymbol{\vartheta} \right),$$

$$\underset{\text{QP}}{\overset{\text{Dual}}{\Longleftrightarrow}} \begin{cases} \max\limits_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} & -\frac{\lambda}{2}\|\mathbf{w}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)\|^2 + \boldsymbol{\varrho}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) \\ \text{s.t.} & \mathbf{A}_e[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] = \boldsymbol{\mu}^T \mathbf{Y}_{\bullet U}, \\ & \mathbf{A}[\boldsymbol{\alpha}; \boldsymbol{\alpha}^*] \le \mathbf{0}, \\ & \mathbf{0} \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \le \mathbf{r}. \end{cases}$$

$$d_k = \begin{cases} 1 & \text{if} \quad \xi_k \le \xi_k^* \\ 0 & \text{otherwise} \end{cases}, \quad \xi_k = H_1(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+L})|), \quad \xi_k^* = I_{\varepsilon}(|f_{\boldsymbol{\vartheta}}(\mathbf{x}_{k+L})|), \; k = 1, \ldots, U.$$

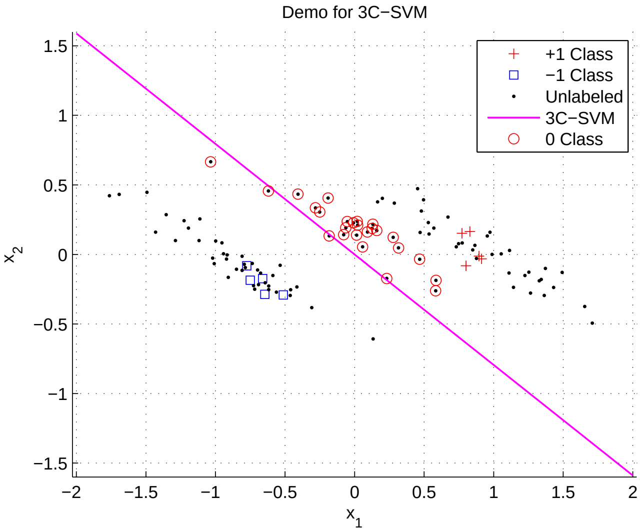
- **Solution:** w is linear combined by α and α^*
  b is attained by KKT condition

3CSVM Demo



Video

3CSVM Result



Experimental Setup

- **Datasets**
  - Two toy datasets
  - Two real-world digit recognition datasets
- **Comparing algorithms**
  - SVMs
  - S^3VMs
  - $\mathcal{U}$-SVMs
  - 3C-SVMs
- **Platform**
  - Matlab 7.3
  - MOSEK 5.0

Data Generation

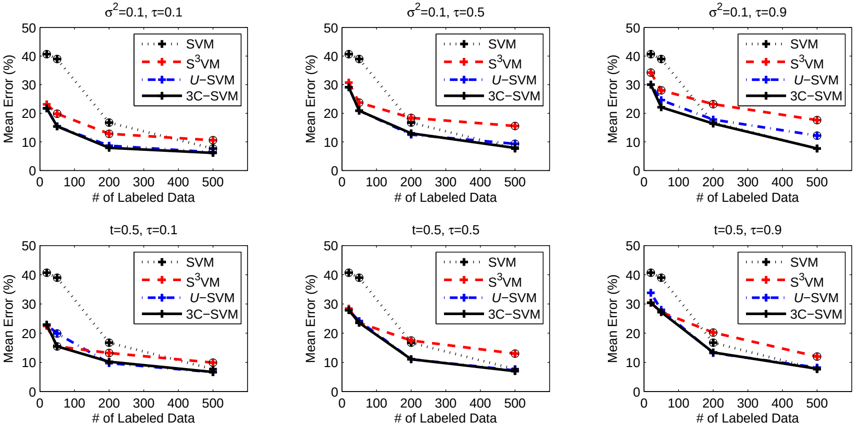
- Following scheme from Sinz et al., 2008
- ± 1-class: $c_i^{\pm} = \pm 0.3$, $i = 1, \ldots, 50$, $\sigma_{1,2}^2 = 0.08$, $\sigma_{3,\ldots,50}^2 = 10$
- Two Gaussians with the Bayes risk being approximately 5%
- First $\mathcal{U}_0$: zero mean, $\sigma_{1,2}^2 = 0.1$, $\sigma_{3,\ldots,50}^2 = 10$
- Second $\mathcal{U}_0$: variance values are the same as ± 1-class data, mean is $t \cdot \mathbf{c}^+$, $t = 0.5$

Test Procedure

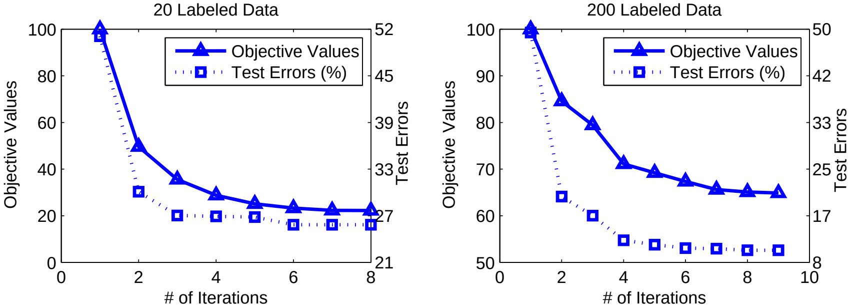
- $L = 20, 50, 200, 500$
- $U = 500 = (\tau U, (1-\tau)U)$, $\tau = 0.1, 0.5, 0.9$
- Labeled + Unlabeled/500 Test, ten-run average
- Hyperparameters
  - Linear kernel
  - Regularized parameters, forward tuning

	$C_{\mathcal{L}}$	$C_{\mathcal{U}}$	ε	κ
SVM	✓	×	×	×
$\mathcal{U}$-SVM	–	✓	✓	×
S^3VM	–	–	×	✓

Accuracy


σ²=0.1, τ=0.1
σ²=0.1, τ=0.5
σ²=0.1, τ=0.9
t=0.5, τ=0.1
t=0.5, τ=0.5
t=0.5, τ=0.9
Mean Error (%)
# of Labeled Data
0
10
20
30
40
50
100
200
300
400
500
SVM
S³VM
U−SVM
3C−SVM

Objective Function Values and Test Errors



Real-world Datasets

- Datasets:
  - Small size: USPS
  - Large size: MNIST
- Setup
  - ± 1-class: Digits "5" and "8"
  - $\mathcal{U}_0$: Other digits
  - $L = 20$
  - $U = 500 = (\tau U, (1-\tau)U)$, $\tau = 0.1, 0.5, 0.9$
  - RBF kernel: $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$, $\gamma = \frac{1}{0.3d}$
  - Other hyperparameters are set similar to those in the synthetic datasets

Accuracy Results

Dataset	Algorithm	$\tau = 0.1$	$\tau = 0.5$	$\tau = 0.9$
USPS	SVM	72.4± 15.9 (**0.7**)	72.4± 15.9 (**9.5**)	72.4± 15.9 (53.1)
	S^{3}VM	56.6 ± 5.9 (**0.0**)	54.5 ± 3.0 (**0.0**)	52.8 ± 6.9 (**0.0**)
	$\mathcal{U}$-SVM	83.1 ± 2.5 (**0.0**)	73.4 ± 4.4 (**0.0**)	64.2 ± 3.6 (**0.0**)
	3C-SVM	**87.2**±2.3	**80.6**±4.8	**75.4**±7.3
MNIST	SVM	70.9± 11.4 (**0.3**)	70.9± 11.4 (**0.8**)	70.9± 11.4 (13.6)
	S^{3}VM	58.9 ± 8.7 (**0.0**)	55.3 ± 8.1 (**0.0**)	53.2 ± 6.3 (**0.0**)
	$\mathcal{U}$-SVM	84.2 ± 2.2 (**0.2**)	80.0 ± 4.6 (**0.9**)	75.0 ± 3.9 (**1.0**)
	3C-SVM	**85.3**±1.6	**82.8**±2.9	**77.6**±3.9

Balance Constraint

- Ideally, $\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\vartheta}(\mathbf{x}_t) = \frac{1}{L}\sum_{i=1}^{L} y_i$, but no improvement from experimental results
- A possible better on, $\frac{1}{U}\sum_{t=L+1}^{L+U} f_{\vartheta}(\mathbf{x}_t) = c$
  c: a user-specified constant, but need tuning

Summary

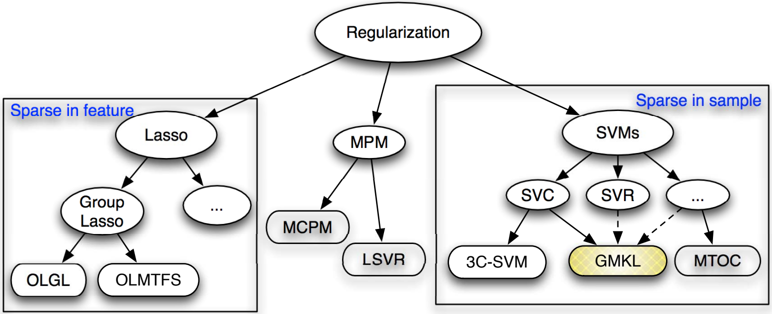
Summary

- A novel maxi-margin classifier, 3C-SVM, can distinguish data into -1, $+1$, and 0, three categories
- The model incorporates standard SVMs, S^3VMs, and $\mathcal{U}$-SVMs as specific cases
- It is solved by the CCCP, in a high efficiency algorithm
- Effectiveness and efficiency are demonstrated

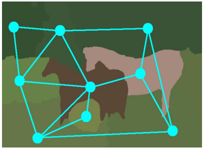
Future work

- Algorithm speedup
- Multi-class extension
- Theoretical analysis, generalization bound

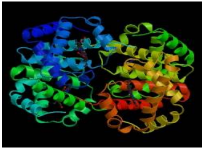
Third Part (Ch. 6): Sparse Generalized Multiple Kernel Learning



Ch. 6: Efficient Sparse Generalized Multiple Kernel Learning



Harchaoui & Bach, 2007



Zien & Ong, 2007

- Applications: Multi-source data fusion (web classification, genome fusion); Image annotation; Text mining; etc.
- Characteristics: Complex tasks; Heterogenous–various medias (text, images, etc.); Huge data
- Solution: Kernel methods⇒Multiple kernels learning
  - Learning combinations of kernels: $\sum_{q=1}^{Q} \theta_q \mathbf{K}_q$, $\theta_q \geq 0$
    - Summing kernels corresponds to concatenating feature spaces
    - E.g., $k_1(\mathbf{z}_1, \mathbf{z}_2) = \langle \phi_1(\mathbf{z}_1), \phi_1(\mathbf{z}_2) \rangle$, $k_2(\mathbf{z}_1, \mathbf{z}_2) = \langle \phi_2(\mathbf{z}_1), \phi_2(\mathbf{z}_2) \rangle$

$$k_1(\mathbf{z}_1, \mathbf{z}_2) + k_2(\mathbf{z}_1, \mathbf{z}_2) = \left\langle \begin{pmatrix} \phi_1(\mathbf{z}_1) \\ \phi_2(\mathbf{z}_1) \end{pmatrix}, \begin{pmatrix} \phi_1(\mathbf{z}_2) \\ \phi_2(\mathbf{z}_2) \end{pmatrix} \right\rangle$$

Ch. 6: Efficient Sparse Generalized Multiple Kernel Learning

- Related work
  - Formulation
    - L_1-MKL (Bach et al. 2004; Lanckriet et al. 2004, etc.): $\|\boldsymbol{\theta}\|_1 \leq 1$
    - L_2-MKL, L_p-MKL (Cortes et al. 2009; Kloft et al. 2010; Xu et al. 2010; etc.): $\|\boldsymbol{\theta}\|_p \leq 1$, $p \neq 1$
  - Speedup: SDP (Lanckriet et al. 2004); SOCP (Bach et al. 2004); SILP (Sonnenburg et al. 2006); Subgradient method (Rakotomamonjy et al. 2008); Level method (Xu et al. 2009; Liu et al. 2009)
- Properties and problems
  - L_1-MKL yields sparse solutions, but discard some useful information
  - L_p-MKL ($p > 1$) yields non-sparse solutions, but prone to noise
- Contributions
  - Generalize L_1-MKL and L_p-MKL
  - Theoretical analysis on the properties of grouping effect and sparsity
  - Solved by the level method

Our Generalized MKL

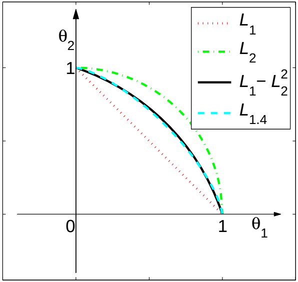
Formulation

$$\min_{\boldsymbol{\theta}\in\Theta}\ \max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad \mathbf{1}_N^\top\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top\left(\sum_{q=1}^{Q}\theta_q\mathbf{K}_q\right)(\boldsymbol{\alpha}\circ\mathbf{y})$$

$$\Theta = \{\boldsymbol{\theta}\in\mathbb{R}_+^Q : v\|\boldsymbol{\theta}\|_1 + (1-v)\|\boldsymbol{\theta}\|_p^p \le 1\}$$

$$\mathcal{A} = \{\boldsymbol{\alpha}\in\mathbb{R}_+^N,\ \boldsymbol{\alpha}^\top\mathbf{y}=0,\ \boldsymbol{\alpha}\le C\mathbf{1}_N\}$$

Here, we consider $p=2$



Properties

- $v\|\boldsymbol{\theta}^\star\|_1 + (1-v)\|\boldsymbol{\theta}^\star\|_2^2 \Leftrightarrow 1$
- For $\mathbf{K}_i = \mathbf{K}_j$,

  $v \neq 1 \quad \theta_q^\star = \max\left\{0, \frac{1}{2(1-v)}\left(\frac{1}{2\lambda}(\boldsymbol{\alpha}\circ\mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha}\circ\mathbf{y}) - v\right)\right\}$ sparsity

  $v = 1 \quad \theta_i$ and θ_j are not unique
- $\frac{(\boldsymbol{\alpha}^\star\circ\mathbf{y})^\top \mathbf{K}_i(\boldsymbol{\alpha}^\star\circ\mathbf{y})}{(\boldsymbol{\alpha}^\star\circ\mathbf{y})^\top \mathbf{K}_j(\boldsymbol{\alpha}^\star\circ\mathbf{y})} \approx 1 \Rightarrow \theta_i^\star \approx \theta_j^\star$ Grouping effect

	L_1-MKL	L_2-MKL	GMKL	Lasso	Elastic net	Group Lasso
Sparsity	✓	×	✓	✓	✓	✓
Non-linearity	✓	✓	✓	×	×	×
Grouping	×	✓	✓	×	✓	×

Algorithm–Level Method

- **Given**: predefined tolerant error $\delta > 0$
- **Initialization**: Let $t = 0$ and $\boldsymbol{\theta}^0 = c\mathbf{1}_q$,
- **Repeat**
  1. Solve the dual problem of an SVM with $\sum_{q=1}^{Q} \theta_q^t \mathbf{K}_q$ to get $\boldsymbol{\alpha}$;
  2. Construct the cutting plane model, $h^t(\boldsymbol{\theta}) = \max_{1\le i\le t} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^i)$;
  3. Calculate the lower bound and the upper bound of the cutting plane $\underline{\mathcal{D}}^t = \min_{\boldsymbol{\theta}\in\Theta} h^t(\boldsymbol{\theta})$, $\overline{\mathcal{D}}^t = \min_{1\le i\le t} \mathcal{D}(\boldsymbol{\theta}^i, \boldsymbol{\alpha}^i)$ and the gap, $\Delta^t = \overline{\mathcal{D}}^t - \underline{\mathcal{D}}^t$;
  4. Project $\boldsymbol{\theta}^t$ onto the level set by solving
  $$\begin{aligned} \min_{\boldsymbol{\theta}\in\Theta} \quad & \|\boldsymbol{\theta} - \boldsymbol{\theta}^t\|_2^2 \\ \text{s.t.} \quad & \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^i) \le \underline{\mathcal{D}}^t + \tau\Delta^t, \quad i = 1, \ldots, t. \end{aligned}$$
  5. Update $t = t + 1$;
- **until** $\Delta^t \le \delta$.

- Formulation:
$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha})$$
$$\Theta = \{\boldsymbol{\theta}\in\mathbb{R}_+^Q : v\|\boldsymbol{\theta}\|_1 + (1-v)\|\boldsymbol{\theta}\|_p \le 1\}$$
$$\mathcal{A} = \{\boldsymbol{\alpha}\in\mathbb{R}_+^N,\ \boldsymbol{\alpha}^\top\mathbf{y} = 0,\ \boldsymbol{\alpha} \le C\mathbf{1}_N\}$$
- Convergence rate: $\mathcal{O}(\delta^{-2})$

Experiments

- Datasets
  - Two toy datasets
  - Eight UCI datasets
  - Three protein subcellular localization data
- Algorithms
  - GMKL
  - L_1-norm MKL (SimpleMKL)
  - L_2-norm MKL
  - Uniformly Weighted MKL (UW-MKL)
- Platform
  - Mosek to solve the QCQP
  - Matlab
  - PC with Intel Core 2 Duo 2.13GHz CPU and 3GB memory.
- Objectives
  - Select important features in a group manner: two toy examples
  - Test efficiency: eight UCI datasets
  - Solve the proteins subcellular localization problem: three datasets

Datasets

Dataset	# Classes	# Training (N)	# Test	# Dim	# Kernel (Q)
Toy1	2	150	150	20	273
Toy2	2	150	150	20	273
Breast	2	341	342	10	143
Heart	2	135	135	13	182
Ionosphere	2	175	176	33	442
Liver	2	172	173	6	91
Pima	2	384	384	8	117
Sonar	2	104	104	60	793
Wdbc	2	284	285	30	403
Wpbc	2	99	99	33	442
Plant	4	470	470		69
Psort+	4	270	271		69
Psort-	5	722	722		69

Experimental Setup

- Preprocessing
  - Construct base kernels
  - Normalize base kernels
- Stopping criteria
  - $\#$ iterations ≤ 500, $\max |\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}| \leq 0.001$
  - L_1-MKL: duality gap ≤ 0.01
  - GMKL, L_2-MKL: $\tau = 0.90$ to 0.99 when $\Delta^t / \mathcal{V}^t \leq 0.01$

Toy Data Generation Scheme

Scheme

- **Toy 1**

$$Y_i = \text{sign}\left(\sum_{j=1}^{3} f_1(x_{ij}) + \epsilon_i\right)$$

- **Toy 2**

$$Y_i = \text{sign}\left(\sum_{j=1}^{3} f_1(x_{ij}) + \sum_{j=4}^{6} f_2(x_{ij}) + \sum_{j=7}^{9} f_3(x_{ij}) + \sum_{j=10}^{12} f_4(x_{ij}) + \epsilon_i\right)$$

$$f_1(a) = -2\sin(2a) + 1 - \cos(2), \; f_2(a) = a^2 - \frac{1}{3},$$
$$f_3(a) = a - \frac{1}{2}, \; f_4(a) = e^{-a} + e^{-1} - 1$$

Remarks

- The outputs (labels) are dominated by only some features
- Each mapping acts on three features equally, implicitly incorporating grouping effect
- Each mapping is with zero mean on the corresponding feature, which yields zero mean on the output

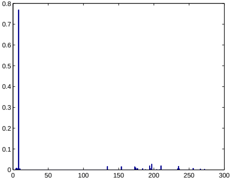
Toy Data Results

Dataset	Method	Accuracy	# Kernel	Time (s)
Toy 1	GMKL	**70.4**$\pm$3.3	36.8$\pm$5.0	2.9$\pm$0.2
	L_1-MKL	69.2$\pm$4.5	22.1$\pm$5.2	4.4$\pm$1.2
	L_2-MKL	68.2$\pm$3.0	273	2.9$\pm$0.4
	UW-MKL	66.3$\pm$5.3	273	–
Toy 2	GMKL	**72.9**$\pm$3.2	43.4$\pm$7.1	2.8$\pm$0.1
	L_1-MKL	72.3$\pm$3.1	30.2$\pm$8.1	4.9$\pm$1.3
	L_2-MKL	71.9$\pm$3.6	273	2.9$\pm$0.1
	UW-MKL	71.6$\pm$4.0	273	–

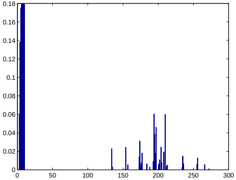
Remarks

- GMKL obtains significant improvement on the accuracy
- The non-sparse MKL models are prone to the noise
- GMKL selects more kernels, about 1.5 times of that selected by the L_1-MKL; while the L_2-MKL selects all kernels
- GMKL and L_2-MKL cost similar same, and cost less time than L_1-MKL

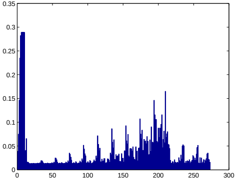
Selected Kernels on Toy Data



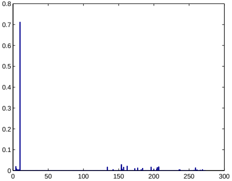
L_1-MKL on Toy 1



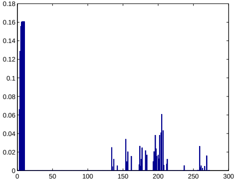
GMKL on Toy 1



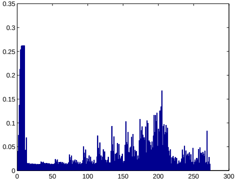
L_2-MKL on Toy 1



L_1-MKL on Toy 2

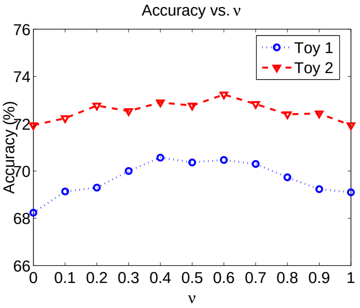


GMKL on Toy 2

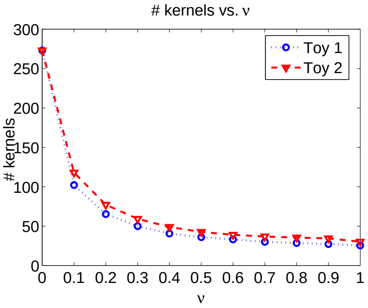


L_2-MKL on Toy 2

Effect of ν on Toy Data



Accuracy vs. ν



No. of selected kernels vs. ν

Remarks

- $\nu = 0$: L_2-MKL
- $\nu = 1$: L_1-MKL
- The best accuracy is achieved when ν is about 0.5
- The number of selected kernels decreases as ν increases

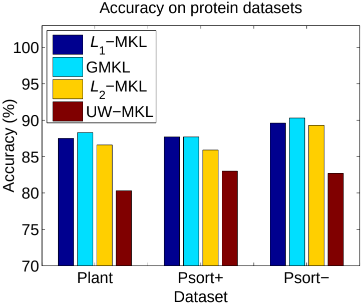
Results on UCI data

Dataset	Method	Accuracy	# Kernel	Time (s)	Dataset	Method	Accuracy	# Kernel	Time (s)
Breast	GMKL	**97.2**±0.5	61.1±6.5	2.8±0.5	Pima	GMKL	†**76.9**±1.6	27.1±2.4	3.8±0.2
	L_1-MKL	97.0±0.7	18.6±3.8	23.0±3.9		L_1-MKL	76.5±1.9	18.7±2.7	24.8±3.4
	L_2-MKL	96.9±0.4	143	5.1±0.3		L_2-MKL	76.0±1.8	117	6.2±1.0
	UW-MKL	**97.2**±0.5	143	–		UW-MKL	76.2±1.7	117	–
Heart	GMKL	**83.9**±1.9	38.5±5.4	1.4±0.1	Sonar	GMKL	80.4±4.1	81.1±6.5	12.4±0.6
	L_1-MKL	83.4±2.6	29.7±4.6	3.5±0.7		L_1-MKL	80.4±4.2	60.3±7.4	16.7±2.0
	L_2-MKL	82.8±2.5	182	1.7±0.1		L_2-MKL	†**83.8**±3.7	793	3.9±0.3
	UW-MKL	**83.9**±1.9	182	–		UW-MKL	81.5±4.3	793	–
Ionosphere	GMKL	91.8±1.7	66.5±7.2	5.1±0.3	Wdbc	GMKL	†**96.0**±1.1	79.7±7.6	6.6±0.8
	L_1-MKL	91.5±2.1	38.4±5.0	19.2±3.3		L_1-MKL	95.3±1.4	34.9±8.9	37.8±5.8
	L_2-MKL	**92.0**±1.8	442	4.0±0.4		L_2-MKL	95.9±0.7	403	7.8±1.6
	UW-MKL	89.9±1.8	442	–		UW-MKL	93.9±1.0	403	–
Liver	GMKL	67.6±1.8	19.5±1.7	1.0±0.0	Wpbc	GMKL	**76.7**±3.3	275.4±96.9	1.3±1.0
	L_1-MKL	64.3±2.8	9.2±3.0	1.7±0.4		L_1-MKL	76.6±2.8	40.4±10.2	4.8±1.0
	L_2-MKL	†**69.7**±2.2	91	1.4±0.0		L_2-MKL	76.3±3.7	442	1.6±0.2
	UW-MKL	67.2±4.6	91	–		UW-MKL	76.6±2.9	442	–

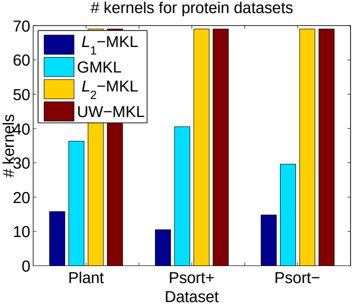
Remarks

- GMKL achieves highest accuracy on five datasets, while L_2-MKL obtains the highest accuracy for the rest three datasets
- GMKL selects more kernels, but achieves better results than L_1-MKL
- GMKL and L_2-MKL cost less time than L_1-MKL

Results on Protein Subcellular Localization Data



Accuracy

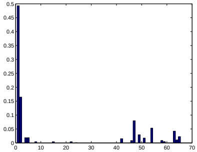


No. of selected kernels

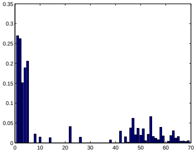
Significant test:

Dataset	GMKL vs. L_1-MKL	GMKL vs. L_2-MKL	GMKL vs. UW-MKL
Plant	0.109	0.109	0.002
Psort+	0.754	0.022	0.002
Psort-	0.022	0.002	0.002

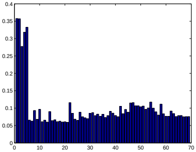
Kernel Weights on Protein Data



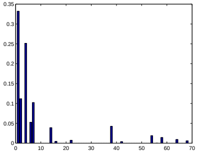
L_1-MKL on Plant



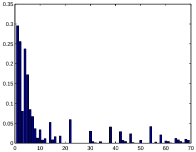
GMKL on Plant



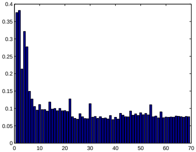
L_2-MKL on Plant



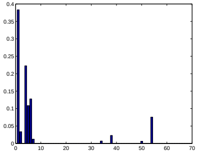
L_1-MKL on Psort+



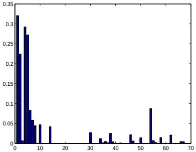
GMKL on Psort+



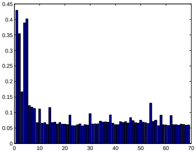
L_2-MKL on Psort+



L_1-MKL on Psort-



GMKL on Psort-



L_2-MKL on Psort-

Summary

- A generalized multiple kernel learning (GMKL) model by imposing L_1-norm and L_2-norm regularization on the kernel weights
- Properties of sparsity and grouping effect are analyzed theoretically
- The model is solved by the level method and the convergence rate is provided
- Experiments on both synthetic and real-world datasets are conducted to demonstrate the effectiveness and efficiency of the model

Future work

- Apply GMKL in other applications, e.g., regression, multiclass classifications
- Apply techniques, e.g., warm start, to speed up GMKL
- Extend GMKL to include the uniformly-weighted MKL as a special case

Conclusions

- Provide promising solutions for large-scale applications in three main learning areas
  - Online learning framework for group lasso and multi-task feature selection
  - Semi-supervised learning model to learn from mixture of relevant and irrelevant data
  - Multiple kernel learning model with sparsity and grouping effect to provide more accurate data similarity representation
- Proposed models are analyzed theoretically and verified empirically
- Toolboxes are provided

Future work

- Developing parsimonious learning models and efficient algorithms
- Real-world applications with the following characteristics
  - Heterogeneous
  - Dynamic
  - Social relation or social information

Contributions

- Main work
  1. **Online Learning for Group Lasso** (ICML'10)
  2. **Online Learning for Multi-Task Feature Selection** (CIKM'10)
  3. **Maximum Margin Semi-supervised Learning With Irrelevant Data** (TR'09)
  4. **Efficient Sparse Generalized Multiple Kernel Learning** (TNN Revision)
- Toolboxes
  1. http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=OLGL
  2. http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=OLMTFS
  3. http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=3CSVM
  4. http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=GMKL

Other Work

1. Localized Support Vector Regression for Time Series Prediction (Neurocomputing'09)
   http://www.cse.cuhk.edu.hk/~hqyang/codes/LSVR_demo.rar
2. Simple and Efficient Multiple Kernel Learning By Group Lasso (ICML'10)
3. Multi-task Learning for One-class Classification (IJCNN'10)
   http://www.cse.cuhk.edu.hk/~hqyang/codes/MT1C_demo.rar
4. Ensemble Learning for Imbalanced E-commerce Transaction Anomaly Classification (ICONIP'09)
5. Sprinkled Latent Semantic Indexing for Text Classification with Background Knowledge (ICONIP'08)
6. Efficient Minimax Clustering Probability Machine by Generalized Probability Product Kernel (IJCNN'08)
   http://www.cse.cuhk.edu.hk/~hqyang/codes/MCPM_demo.rar
7. Non-Monotonic Feature Selection (TR'09)
8. Online Learning for Conjoint Analysis (TR'10)

Questions ?

Haiqin Yang
www.cse.cuhk.edu.hk/~hqyang
hqyang@cse.cuhk.edu.hk