Kernel Learning for Multi-modal Recognition Tasks

J. Saketha Nath

CSE, IIT-B

IBM Workshop

Multi-modal Learning Tasks

- Multiple views or descriptions of the data is available
  - E.g. Object Categorization

Object Categorization

Figure: Daffodils and Dandelions can be distinguished using shape features.

Figure: Blue-bell and Tulip can be distinguished using color features.

Source: `http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html`

Object Categorization — Features

Typical Features:

- **HSV** Color features.
- **SIFT** Local texture and shape features.
- **HOG** Global shape features.

Multi-modal Learning Tasks

- Multiple views or descriptions of the data is available
  - E.g. Object Categorization, Multi-modal Speech/Speaker/Activity recognition

Multi-modal Learning Tasks

- Multiple views or descriptions of the data is available
  - E.g. Object Categorization, Multi-modal Speech/Speaker/Activity recognition, Text Mining ?

Multi-modal Learning Tasks

- Multiple views or descriptions of the data is available
  - E.g. Object Categorization, Multi-modal Speech/Speaker/Activity recognition, Text Mining ?
- Build classifier using all features (say SVM)

Multi-modal Learning Tasks

- Multiple views or descriptions of the data is available
  - E.g. Object Categorization, Multi-modal Speech/Speaker/Activity recognition, Text Mining ?
- Build classifier using all features (say SVM)

Can we do better?

- E.g. Nilsback and Zisserman (CVPR06) found that:
  - All the 3 kinds of features are critical
  - Not all features in each kind may be important
- Exploit natural grouping of features.

Scope and Objective

Scope:

- Binary classification task, Kernel methods like SVM
- Simultaneous feature selection and classifier construction
  - *Multiple Kernel Learning* [Lanckriet et.al., '02]

Objective:

- Customized for multi-modal tasks
  - Exploit the group structure in features (prior info)

Problem Setting

Given:

- $\mathcal{D} = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \{-1, 1\}, i = 1, \ldots, m\}$
- Feature maps $\Phi_j, j = 1, \ldots, n$ (n is modality of data)
  - E.g. $\Phi_1(x_i)$ — feature vector describing color of flower x_i
- Using each Φ_j generate various kernels (linear, polynomial, Gaussian)

Problem Setting

Given:

- Kernels $\mathbf{K}_{jk}, j = 1, \ldots, n, k = 1, \ldots, n_j$
  - n — modality of data
  - n_j — no. kernels generated from j^{th} mode

Problem Setting

Given:

- Kernels $\mathbf{K}_{jk}, j = 1, \ldots, n, k = 1, \ldots, n_j$
  - n — modality of data
  - n_j — no. kernels generated from j^{th} mode

MKL Task:

- Simultaneously determine weights given to Kernels (features) and the classifier
- Utilize prior information regarding the Kernels

Existing Methods

SVM

$$\max_{\alpha \in S} \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \mathbf{YKY}\alpha$$

- $\mathbf{K} = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{K}_{jk}$ (concatenation of all features)

Existing Methods

SVM

$$\max_{\alpha \in S} \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{YKY} \alpha$$

- $\mathbf{K} = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{K}_{jk}$ (concatenation of all features)

MKL

$$\min_{\lambda_{jk} \geq 0, \sum_{j,k} \lambda_{jk} = 1} \max_{\alpha \in S} \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{YKY} \alpha$$

- $\mathbf{K} = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \lambda_{jk} \mathbf{K}_{jk}$ (convex combination of Kernels)
- Equivalent to selecting *single best* kernel!

Proposed Methodology

- $\Phi_{jk}(\cdot)$ implicit map defined by K_{jk}
- $f(x) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^{\top} \Phi_{jk}(x) - b$

Proposed Methodology

- $\Phi_{jk}(\cdot)$ implicit map defined by K_{jk}
- $f(x) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^{\top} \Phi_{jk}(x) - b$

SVM Formulation:

$$\min_{\mathbf{w}_{jk}, b, \xi_i} \quad \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n_j} \|\mathbf{w}_{jk}\|_2 + C \sum_i \xi_i$$

$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^{\top} \Phi_{jk}(x_i) - b\right) \geq 1 - \xi_i, \ \ \xi_i \geq 0$$

Proposed Methodology

- $\Phi_{jk}(\cdot)$ implicit map defined by K_{jk}
- $f(x) = \sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^\top \Phi_{jk}(x) - b$

Convex Formulation:

$$\min_{\mathbf{w}_{jk}, b, \xi_i} \quad \frac{1}{2} \left[\max_j \left(\sum_{k=1}^{n_j} \|\mathbf{w}_{jk}\|_2 \right)^2 \right] + C \sum_i \xi_i$$

$$\text{s.t.} \quad y_i \left(\sum_{j=1}^{n} \sum_{k=1}^{n_j} \mathbf{w}_{jk}^\top \Phi_{jk}(x_i) - b \right) \geq 1 - \xi_i, \; \xi_i \geq 0$$

Dual

Formulation:

$$\min_{\lambda_j \in \Delta_{n_j} \forall j} \quad \max_{\gamma \in \Delta_n,\ \alpha \in S} \ \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \left[\sum_{j=1}^{n} \left(\frac{\sum_{k=1}^{n_j} \lambda_{jk} \mathbf{K}_{jk}}{\gamma_j} \right) \right] \mathbf{Y} \alpha \quad (1)$$

Dual

Formulation:

$$\min_{\lambda_j \in \Delta_{n_j} \forall j} \quad \max_{\gamma \in \Delta_n,\ \alpha \in S} \ \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \mathbf{Y} \left[\sum_{j=1}^{n} \left(\frac{\sum_{k=1}^{n_j} \lambda_{jk} \mathbf{K}_{jk}}{\gamma_j} \right) \right] \mathbf{Y} \alpha \qquad (1)$$

Comments:

- Equivalent to SVM formulation with $\mathbf{K} \equiv \sum_{j=1}^{n} \left(\frac{\sum_{k=1}^{n_j} \lambda^*_{jk} \mathbf{K}_{jk}}{\gamma^*_j} \right)$.
- $\frac{1}{\gamma^*_j}$ weight for j^{th} mode and λ^*_{jk} weight for k^{th} kernel in j^{th} mode.
- $\gamma^*_j \neq 0, j = 1, \ldots, n$ provided $\mathbf{K}_{jk}$ are positive definite.
- λ^*_{jk} is highly sparse for each j.
- $n = 1$ gives back MKL!

Efficient Solver

- Pose as SOCP, solve using `SeDuMi`, `Mosek`
- Extensions of iterative algorithms in MKL literature suffer from non-convexity problems

Efficient Solver

- Pose as SOCP, solve using `SeDuMi`, `Mosek`
- Extensions of iterative algorithms in MKL literature suffer from non-convexity problems

Mirror Descent based alg.:

- Iterative alg. solving an SVM at each step.
- Far more scalable than state-of-the-art MKL solvers ($n = 1$)

Results — Object Categorization

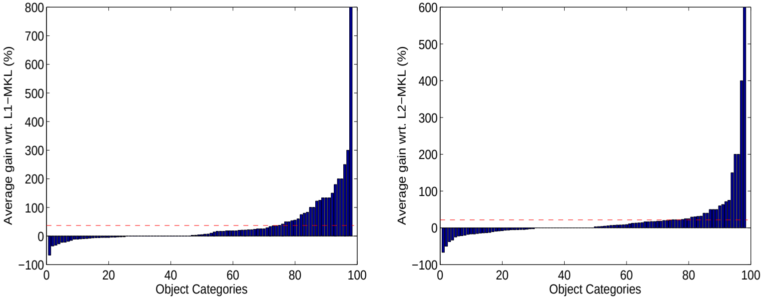

Figure: Plot of average gain (%) in accuracy with **MixNorm-MKL** on Caltech-101.

Results — Scaling

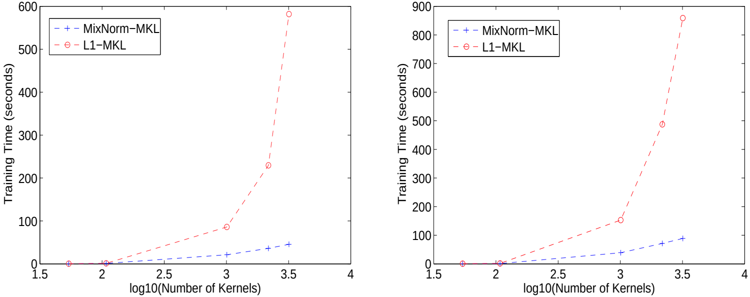

Figure: Scaling plots comparing mirror-descent based algorithm and `simpleMKL`.

THANK YOU