Multiple Kernel Learning

J. Saketha Nath

CSE, IIT-Bombay

Problem of Learning

Given: Set of m pairs of the form (x_i, y_i)

- $x_i \in \mathcal{X}$ is some object e.g., picture
- $y_i \in \mathcal{Y}$ is label of object e.g., chair/bike/panda

.

GIVEN: Set of m pairs of the form (x_i, y_i)

- $x_i \in \mathcal{X}$ is some object e.g., picture, financial-profile of industry etc.
- $y_i \in \mathcal{Y}$ is label of object e.g., chair/bike/panda, stock-value etc.

Problem of Learning

GIVEN: Set of m pairs of the form (x_i, y_i)

- $x_i \in \mathcal{X}$ is some object e.g., picture, financial-profile of industry etc.
- $y_i \in \mathcal{Y}$ is label of object e.g., chair/bike/panda, stock-value etc.

GOAL: Construct $f : \mathcal{X} \mapsto \mathcal{Y}$ such that $f(x) = y$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$

Mathematically:

$$\min_{f \in \mathcal{F}} \underbrace{P[f(X) \neq Y]}_{\text{generalization error}}$$

Typical sets of classifiers:

Linear $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{w}^\top \mathbf{x} - b)\}$

Quadratic $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c)\}$

Polynomial $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbb{P}(\mathbf{x}))\}$

Non-linear $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(g(\mathbf{x}))\}$

Binary Classification

Mathematically:

$$\min_{f \in \mathcal{F}} \underbrace{P[f(X) \neq Y]}_{\text{generalization error}} \qquad \text{impossible!}$$

Typical sets of classifiers:

Linear $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{w}^\top \mathbf{x} - b)\}$

Quadratic $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c)\}$

Polynomial $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbb{P}(\mathbf{x}))\}$

Non-linear $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(g(\mathbf{x}))\}$

- Obtain bounds using concentration inequalities [Boucheron et.al., 04], Rademacher complexities [Bartlett & Mendelson, 02]:

$$\underbrace{P_f}_{\text{misclass. prob.}} \leq \underbrace{P_f^m}_{\text{est. of prob.}} +2 \underbrace{\mathcal{R}(\mathcal{F})}_{\text{complexity of } \mathcal{F}} +\sqrt{\frac{2\log(2/\delta)}{m}}$$

holds $\forall\ f \in \mathcal{F}$ with probability $1-\delta$

- Obtain bounds using concentration inequalities [Boucheron et.al., 04], Rademacher complexities [Bartlett & Mendelson, 02]:

$$\underbrace{P_f}_{\text{misclass. prob.}} \leq \underbrace{P_f^m}_{\text{est. of prob.}} + 2 \underbrace{\mathcal{R}(\mathcal{F})}_{\text{complexity of } \mathcal{F}} + \sqrt{\frac{2\log(2/\delta)}{m}}$$

holds $\forall\ f \in \mathcal{F}$ with probability $1 - \delta$

- Extensions of Vapnik-Chervonenkis-type inequalities [Vapnik, 98].

- Function class is set of all linear discriminators (with strict separation)
    - $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{w}^\top \mathbf{x} - b)\}$
- $R(\mathcal{F}) \propto \|\mathbf{w}\|_2^2$

- Function class is set of all linear discriminators (with strict separation)
  - $\mathcal{F} = \{f \mid f(\mathbf{x}) = \mathsf{sign}(\mathbf{w}^\top \mathbf{x} - b)\}$
- $R(\mathcal{F}) \propto \|\mathbf{w}\|_2^2$

SVM Problem [Cortes & Vapnik, 95]:

$$\min_{\mathbf{w},b,\xi_i} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^m \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

SVM Dual Problem

$$\max_{\alpha_i} \quad \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \ \sum_{i=1}^m \alpha_i y_i$$

SVM Dual Problem

$$\max_{\alpha_i} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{m} \alpha_i y_i$$

- $f(\mathbf{x}) = \mathsf{sign}(\mathbf{w}^\top \mathbf{x} - b) = \mathsf{sign}(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} - b)$
- Training and prediction involve dot-products alone
- Dual soln. is sparse — fast algorithms

Non-linear Discriminators

- Linear discriminators too restrictive

Non-linear Discriminators

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes
  - E.g.,

$$
\begin{aligned}
f(x_1, x_2) &= \mathsf{sign}(a_1 \underbrace{x_1^2}_{z_1} + a_2 \underbrace{x_1 x_2}_{z_2} + a_3 \underbrace{x_2^2}_{z_3}) \\
&= \mathsf{sign}(a_1 z_1 + a_2 z_2 + a_3 z_3)
\end{aligned}
$$

which is infact a linear discriminator in z-space.

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes
  - E.g.,

$$f(x_1, x_2) = \mathsf{sign}(a_1 \underbrace{x_1^2}_{z_1} + a_2 \underbrace{x_1 x_2}_{z_2} + a_3 \underbrace{x_2^2}_{z_3})$$
$$= \mathsf{sign}(a_1 z_1 + a_2 z_2 + a_3 z_3)$$

    which is infact a linear discriminator in z-space.
  - computing $\mathbf{z} = \phi(\mathbf{x})$ is inefficient

Non-linear Discriminators

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes
  - E.g.,

$$f(x_1, x_2) = \mathsf{sign}(a_1 \underbrace{x_1^2}_{z_1} + a_2 \underbrace{x_1 x_2}_{z_2} + a_3 \underbrace{x_2^2}_{z_3})$$
$$= \mathsf{sign}(a_1 z_1 + a_2 z_2 + a_3 z_3)$$

    which is infact a linear discriminator in z-space.
  - computing $\mathbf{z} = \phi(\mathbf{x})$ is inefficient
  - But, $\mathbf{z}_1^\top \mathbf{z}_2 = (\mathbf{x}_1^\top \mathbf{x}_2)^d$

Non-linear Discriminators

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes
  - E.g.,

$$f(x_1, x_2) = \mathsf{sign}(a_1 \underbrace{x_1^2}_{z_1} + a_2 \underbrace{x_1 x_2}_{z_2} + a_3 \underbrace{x_2^2}_{z_3})$$
$$= \mathsf{sign}(a_1 z_1 + a_2 z_2 + a_3 z_3)$$

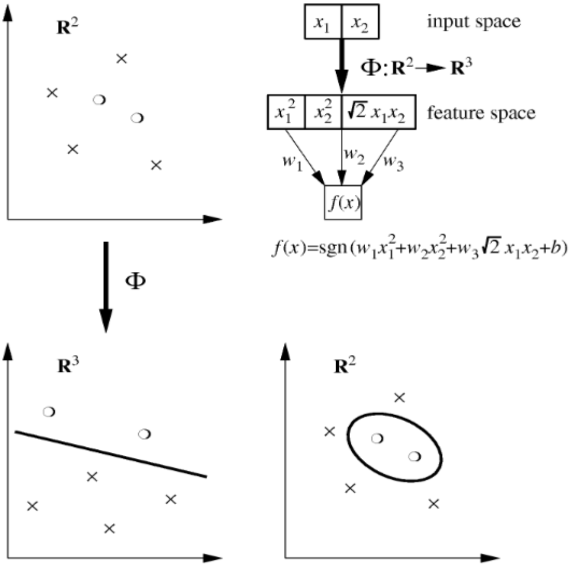
  which is infact a linear discriminator in z-space.
  - computing $\mathbf{z} = \phi(\mathbf{x})$ is inefficient
  - But, $\mathbf{z}_1^\top \mathbf{z}_2 = (\mathbf{x}_1^\top \mathbf{x}_2)^d$
  - Very useful as SVM relies on dot-products only

Non-linear Discriminators

- Linear discriminators too restrictive
- Easy extension to non-linear discriminators ? Yes
  - E.g.,

$$\begin{aligned} f(x_1, x_2) &= \text{sign}(a_1 \underbrace{x_1^2}_{z_1} + a_2 \underbrace{x_1 x_2}_{z_2} + a_3 \underbrace{x_2^2}_{z_3}) \\ &= \text{sign}(a_1 z_1 + a_2 z_2 + a_3 z_3) \end{aligned}$$

    which is infact a linear discriminator in z-space.
  - computing $\mathbf{z} = \phi(\mathbf{x})$ is inefficient
  - But, $\mathbf{z}_1^\top \mathbf{z}_2 = (\mathbf{x}_1^\top \mathbf{x}_2)^d$
  - Very useful as SVM relies on dot-products only
- Can be extended to generic input-spaces and non-linear discriminators — kernel trick



Source: [Schölkopf & Smola, 02]

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ ($\mathbf{G}_{ij} = k(x_i, x_j)$) is psd

Kernels

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ $(\mathbf{G}_{ij} = k(x_i, x_j)$ is psd
- E.g. $\underbrace{k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}}_{\text{linear}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d}_{\text{polynomial}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = \exp\{\mathbf{x}^\top \mathbf{z}\}}_{\text{Gaussian}}$
- Intuitively, k measures similarity

Kernels

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ $(\mathbf{G}_{ij} = k(x_i, x_j)$ is psd
- E.g. $\underbrace{k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}}_{\text{linear}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d}_{\text{polynomial}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = \exp\{\mathbf{x}^\top \mathbf{z}\}}_{\text{Gaussian}}$
- Intuitively, k measures similarity

[Schölkopf & Smola, 02]

- $\exists\ \phi : \mathcal{X} \mapsto \mathcal{H}\ \ni\ < \phi(x), \phi(y) >_{\mathcal{H}} = k(x, y)$

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ ($\mathbf{G}_{ij} = k(x_i, x_j)$) is psd
- E.g. $\underbrace{k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}}_{\text{linear}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d}_{\text{polynomial}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = \exp\{\mathbf{x}^\top \mathbf{z}\}}_{\text{Gaussian}}$
- Intuitively, k measures similarity

[Schölkopf & Smola, 02]

- $\exists\ \phi : \mathcal{X} \mapsto \mathcal{H}\ \ni\ < \phi(x), \phi(y) >_{\mathcal{H}} = k(x, y)$
- Kernel Trick: Replace dot-product with kernel
  - Essentially working in $\mathcal{H}$

Kernels

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ $(\mathbf{G}_{ij} = k(x_i, x_j))$ is psd
- E.g. $\underbrace{k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}}_{\text{linear}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d}_{\text{polynomial}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = \exp\{\mathbf{x}^\top \mathbf{z}\}}_{\text{Gaussian}}$
- Intuitively, k measures similarity

[Schölkopf & Smola, 02]

- $\exists\ \phi : \mathcal{X} \mapsto \mathcal{H}\ \ni\ < \phi(x), \phi(y) >_{\mathcal{H}} = k(x, y)$
- Kernel Trick: Replace dot-product with kernel
  - Essentially working in $\mathcal{H}$
  - $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} - b)$

Kernels

- Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ symmetric and positive
  - Positive: For any $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ gram-matrix $\mathbf{G}$ ($\mathbf{G}_{ij} = k(x_i, x_j)$) is psd
- E.g. $\underbrace{k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}}_{\text{linear}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^d}_{\text{polynomial}}$, $\underbrace{k(\mathbf{x}, \mathbf{z}) = \exp\{\mathbf{x}^\top \mathbf{z}\}}_{\text{Gaussian}}$
- Intuitively, k measures similarity

[Schölkopf & Smola, 02]

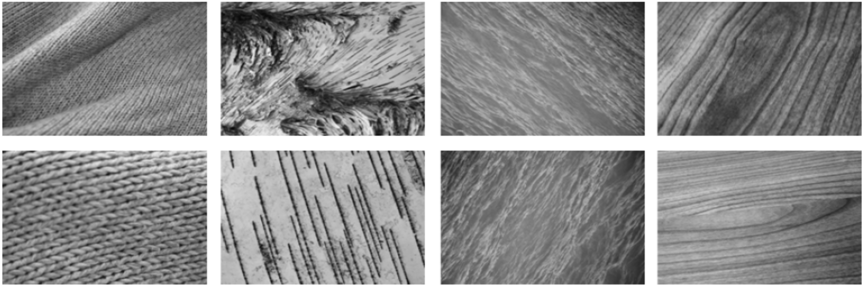
- $\exists\ \phi : \mathcal{X} \mapsto \mathcal{H}\ \ni\ < \phi(x), \phi(y) >_{\mathcal{H}} = k(x, y)$
- Kernel Trick: Replace dot-product with kernel
  - Essentially working in $\mathcal{H}$
  - $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) - b)$

Success Story

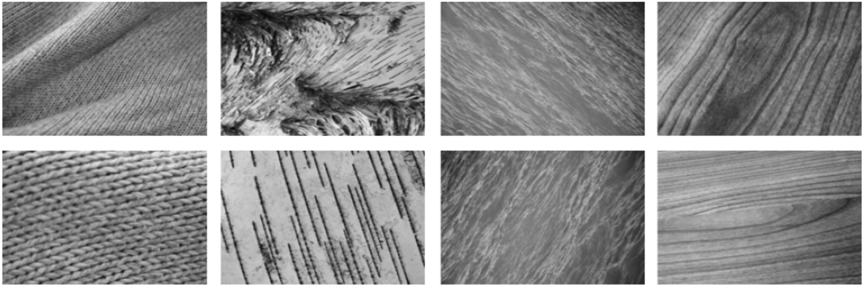
- SVMs achieve state-of-the-art performance in many applications
  - Text Classification
  - Object Categorization
  - Bio-informatics tasks

Success Story

- SVMs achieve state-of-the-art performance in many applications
    - Text Classification
    - Object Categorization
    - Bio-informatics tasks
- Choice of kernel is crucial
- Application specific highly tuned kernels
    - Own merits and demerits
    - Trade-off discriminative-power vs. invariance
    - Utilize different aspects of data



Source: [Varma & Ray, 07]



Source: [Varma & Ray, 07]

Source: [Vert, 09]

- Given base kernels: $k_1, k_2, \ldots, k_n$
- Combine them to achieve better generalization ?
  - Convex or linear or non-linear combinations

- Given base kernels: $k_1, k_2, \ldots, k_n$
- Combine them to achieve better generalization ?
  - Convex or linear or non-linear combinations

MKL Framework: [Lanckriet et.al., 04]

Simultaneously optimize for "best" combination of kernels as well as the discriminating hyperplane in context of SVMs

Combinations of Kernels

- Conic combinations of positive kernels are positive
  - $\underbrace{k}_{\phi} = \underbrace{\gamma_1 k_1}_{\phi_1} + \ldots + \underbrace{\gamma_n k_n}_{\phi_n},\ \gamma_i \geq 0$
  - ϕ can be taken as concatenation of $\phi_1, \ldots, \phi_n$
  - Convex combinations are positive

- Conic combinations of positive kernels are positive
  - $\underbrace{k}_{\phi} = \underbrace{\gamma_1 k_1}_{\phi_1} + \ldots + \underbrace{\gamma_n k_n}_{\phi_n}, \ \gamma_i \geq 0$
  - ϕ can be taken as concatenation of $\phi_1, \ldots, \phi_n$
  - Convex combinations are positive
- Products are positive
  - $k = k_1 k_2$

Combinations of Kernels

- Conic combinations of positive kernels are positive
  - $\underbrace{k}_{\phi} = \underbrace{\gamma_1 k_1}_{\phi_1} + \ldots + \underbrace{\gamma_n k_n}_{\phi_n},\ \gamma_i \geq 0$
  - ϕ can be taken as concatenation of $\phi_1, \ldots, \phi_n$
  - Convex combinations are positive
- Products are positive
  - $k = k_1 k_2$
- Polynomials are positive
  - $k = \left(\sum_i \gamma_i k_i\right)^d,\ d \geq 1, \gamma_i \geq 0$

- Conic combinations of positive kernels are positive
  - $\underbrace{k}_{\phi} = \underbrace{\gamma_1 k_1}_{\phi_1} + \ldots + \underbrace{\gamma_n k_n}_{\phi_n},\ \gamma_i \geq 0$
  - ϕ can be taken as concatenation of $\phi_1, \ldots, \phi_n$
  - Convex combinations are positive
- Products are positive
  - $k = k_1 k_2$
- Polynomials are positive
  - $k = (\sum_i \gamma_i k_i)^d,\ d \geq 1, \gamma_i \geq 0$
- Exponentials are positive
  - $k = \exp\{\sum_i \gamma_i k_i\},\ \gamma_i \geq 0$

- Recall, $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_2$

Rademacher Complexity

- Recall, $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_2$
- Modification in feature space (with kernel k):
    - $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\text{trace}(\mathbf{K})}$

- Recall, $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_2$
- Modification in feature space (with kernel k):
  - $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\text{trace}(\mathbf{K})}$
- If $\mathbf{K} = \sum_{i=1}^{n} \gamma_i \mathbf{K}_i, \gamma_i \geq 0$, then $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{n \max_i \{\text{trace}(\mathbf{K}_i)\}}$

- Recall, $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_2$
- Modification in feature space (with kernel k):
  - $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\text{trace}(\mathbf{K})}$
- If $\mathbf{K} = \sum_{i=1}^{n} \gamma_i \mathbf{K}_i, \gamma_i \geq 0$, then $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{n \max_i \{\text{trace}(\mathbf{K}_i)\}}$
  - Tighter bound: $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\log(n)}$ [Cortes et.al., 10]
  - Weak dependence on n

- Recall, $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_2$
- Modification in feature space (with kernel k):
  - $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\text{trace}(\mathbf{K})}$
- If $\mathbf{K} = \sum_{i=1}^{n} \gamma_i \mathbf{K}_i, \gamma_i \geq 0$, then $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{n \max_i \{\text{trace}(\mathbf{K}_i)\}}$
  - Tighter bound: $\mathcal{R}(\mathcal{F}) \propto \|\mathbf{w}\|_{\mathcal{H}} \sqrt{\log(n)}$ [Cortes et.al., 10]
  - Weak dependence on n

Maximization of margin (min. of $\|\mathbf{w}\|_{\mathcal{H}}$) will lead to good generalization as long as $trace(\mathbf{K})$ is bounded (finite number of kernels)

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i, \ \gamma \geq 0$

MKL FORMULATION

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i, \ \gamma \geq 0$

[LANCKRIET ET.AL., 04]:

$$\min_{w,b,\xi_i} \quad \frac{1}{2}\|w\|_{\mathcal{H}}^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.} \quad y_i(< w, \phi(\mathbf{x}_i) >_{\mathcal{H}} - b) \geq 1 - \xi_i, \ \xi_i \geq 0$$

MKL Formulation

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i,\ \gamma \geq 0$

[Lanckriet et.al., 04]:

$$\max_{\alpha} \quad \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \mathbf{YKY}\alpha$$
$$\text{s.t.} \quad \mathbf{0} \leq \alpha \leq C\mathbf{1},\ \mathbf{y}^\top \alpha = 0$$

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i,\ \gamma \geq 0$

[LANCKRIET ET.AL., 04]:

$$\min_{\gamma \geq 0} \ \max_{\alpha} \quad \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \mathbf{Y} \sum_{i=1}^{n} \gamma_i \mathbf{K}_i \mathbf{Y} \alpha$$

$$\text{s.t.} \qquad \mathbf{0} \leq \alpha \leq C\mathbf{1},\ \mathbf{y}^\top \alpha = 0$$

$$\text{trace}\left(\sum_{i=1}^{n} \gamma_i \mathbf{K}_i\right) \leq d$$

MKL Formulation

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i, \ \gamma \geq 0$

[Lanckriet et.al., 04]:

$$\min_{\gamma \geq 0} \ \max_{\alpha} \quad \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \sum_{i=1}^{n} \gamma_i \mathbf{K}_i \mathbf{Y} \alpha$$

$$\text{s.t.} \quad \mathbf{0} \leq \alpha \leq C\mathbf{1}, \ \mathbf{y}^\top \alpha = 0$$

$$\sum_{i=1}^{n} \gamma_i \text{trace}(\mathbf{K}_i) \leq d$$

MKL Formulation

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i,\ \gamma \geq 0$

[Lanckriet et.al., 04]:

$$\begin{aligned} \min_{\gamma \geq 0} \ \max_{\alpha} \quad & \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} \sum_{i=1}^{n} \gamma_i \mathbf{K}_i \mathbf{Y} \alpha \\ \text{s.t.} \quad & \mathbf{0} \leq \alpha \leq C\mathbf{1},\ \mathbf{y}^\top \alpha = 0 \\ & \sum_{i=1}^{n} \gamma_i \text{trace}(\mathbf{K}_i) \leq d \end{aligned}$$

- Unit-trace Normalization: $\mathbf{K}_i \mapsto \frac{\mathbf{K}_i}{\text{trace}(\mathbf{K}_i)}$

MKL Formulation

Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i, \; \gamma \geq 0$

[Lanckriet et.al., 04]:

$$\min_{\gamma \geq 0} \; \max_{\alpha} \quad \mathbf{1}^\top \alpha - \frac{d}{2} \alpha^\top \mathbf{Y} \sum_{i=1}^{n} \gamma_i \mathbf{K}_i \mathbf{Y} \alpha$$

$$\text{s.t.} \quad \mathbf{0} \leq \alpha \leq C\mathbf{1}, \; \mathbf{y}^\top \alpha = 0$$

$$\sum_{i=1}^{n} \gamma_i \leq 1$$

- Unit-trace Normalization: $\mathbf{K}_i \mapsto \frac{\mathbf{K}_i}{\text{trace}(\mathbf{K}_i)}$ (convex combination)

MKL Formulation

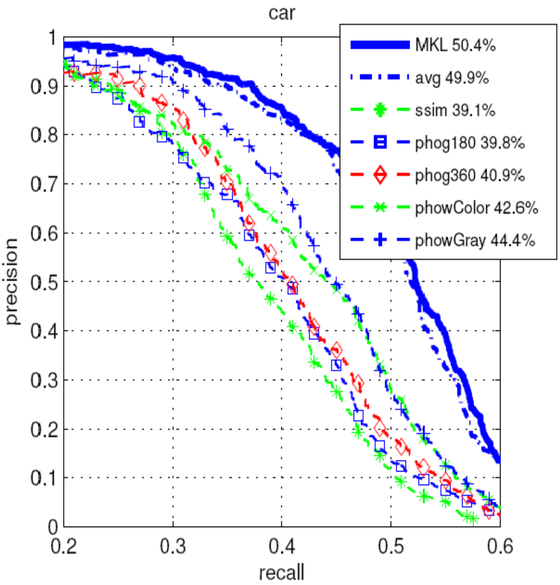
Deals with conic combination of kernels $k = \sum_{i=1}^{n} \gamma_i k_i,\ \gamma \geq 0$

[Lanckriet et.al., 04]:

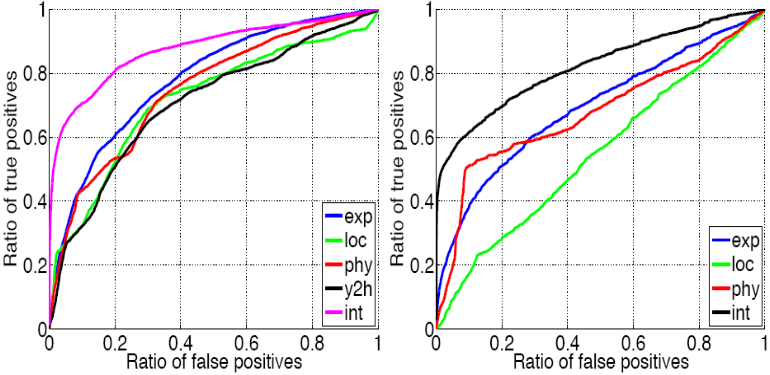
$$\begin{aligned} \min_{\gamma \geq 0} \ \max_{\alpha} \quad & \mathbf{1}^\top \alpha - \frac{d}{2} \alpha^\top \mathbf{Y} \sum_{i=1}^{n} \gamma_i \mathbf{K}_i \mathbf{Y} \alpha \\ \text{s.t.} \quad & \mathbf{0} \leq \alpha \leq C\mathbf{1},\ \mathbf{y}^\top \alpha = 0 \\ & \sum_{i=1}^{n} \gamma_i \leq 1 \end{aligned}$$

- Unit-trace Normalization: $\mathbf{K}_i \mapsto \frac{\mathbf{K}_i}{\text{trace}(\mathbf{K}_i)}$ (convex combination)
- Application of min-max thm. helps pose as QCQP
- Can be solved using `SeDuMi` or `Mosek`

Object Categorization Results



Source: [Vedaldi et.al., 09]



Source: [Bleakley et.al., 07]

Techniques for Solving MKL

- `SMO` algorithm [Bach et.al., 04]
- Pose as SILP, solve series of SVMs [Sonnenburg et.al., 06]

Techniques for Solving MKL

- `SMO` algorithm [Bach et.al., 04]
- Pose as SILP, solve series of SVMs [Sonnenburg et.al., 06]
- `SimpleMKL`: projected gradient descent [Rakotomamonjy et.al., 08]
- Extended level-set method [Xu et.al., 08]
- Mirror descent based alg. [Nath et.al., 09]

Techniques for Solving MKL

- `SMO` algorithm [Bach et.al., 04]
- Pose as SILP, solve series of SVMs [Sonnenburg et.al., 06]
- `SimpleMKL`: projected gradient descent [Rakotomamonjy et.al., 08]
- Extended level-set method [Xu et.al., 08]
- Mirror descent based alg. [Nath et.al., 09] highly scalable

Projected (Sub)Gradient Descent

- Extension of steepest descent alg. for constrained problems
- $\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ (f is convex, Lipschitz, $\mathcal{X}$ is compact)
- At iteration k:
  - f is approx. by linear func. $f(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top(\mathbf{x} - \mathbf{x}_k)$
  - valid only when $\|\mathbf{x} - \mathbf{x}_k\|_2$ is small

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}\in\mathcal{X}} \quad s_k \nabla f(\mathbf{x}_k)^\top(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_2^2 \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \quad \frac{1}{2}\|\mathbf{x} - (\mathbf{x}_k - s_k \nabla f(\mathbf{x}_k))\|_2^2 \\
&= \Pi_{\mathcal{X}}(\mathbf{x}_k - s_k \nabla f(\mathbf{x}_k)))
\end{aligned}
$$

- Convergence guarantees with some choices of step-sizes (s_k)
- "Optimal" for Euclidean geometry

$$\min_{\gamma \in \Delta_n} \max_{\alpha \in S_m(C)} \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \mathbf{Y} (\sum_{i=1}^{n} \gamma_i \mathbf{K}_i) \mathbf{Y} \alpha$$

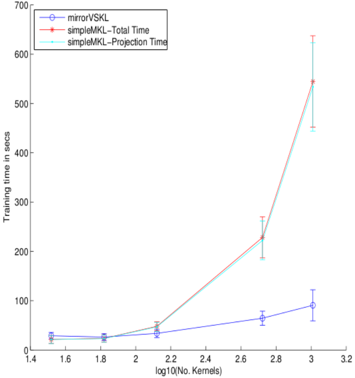
$$\min_{\gamma\in\Delta_n} \underbrace{\max_{\alpha\in S_m(C)} \mathbf{1}^\top\alpha - \frac{1}{2}\alpha^\top\mathbf{Y}(\sum_{i=1}^{n}\gamma_i\mathbf{K}_i)\mathbf{Y}\alpha}_{g(\gamma)}$$

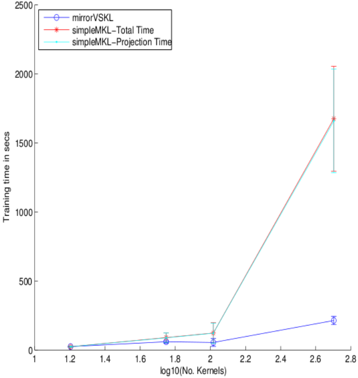
$$\min_{\gamma \in \Delta_n} \underbrace{\max_{\alpha \in S_m(C)} \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \mathbf{Y}(\sum_{i=1}^{n} \gamma_i \mathbf{K}_i)\mathbf{Y}\alpha}_{g(\gamma)}$$

- Danskin's theorem provides $\nabla g(\gamma)$ (need to solve SVM problem)
- Apply projected gradient descent
- Step-sizes chosen by line-search (involves some more SVM solving)

KEY ADVANTAGES [NATH ET.AL, 09]:

- No. iterations is $O(\log(n))$
- No expensive projection step
- Step-sizes can be easily computed





Mirror Descent

- Similar to projected gradient descent
- Per-step problem has Bregman divergence based regularizer:

- Similar to projected gradient descent
- Per-step problem has Bregman divergence based regularizer:

$$\mathbf{x}_{k+1} = \text{arg}\min_{\mathbf{x}\in\mathcal{X}} \quad s_k \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_2^2$$

- Similar to projected gradient descent
- Per-step problem has Bregman divergence based regularizer:

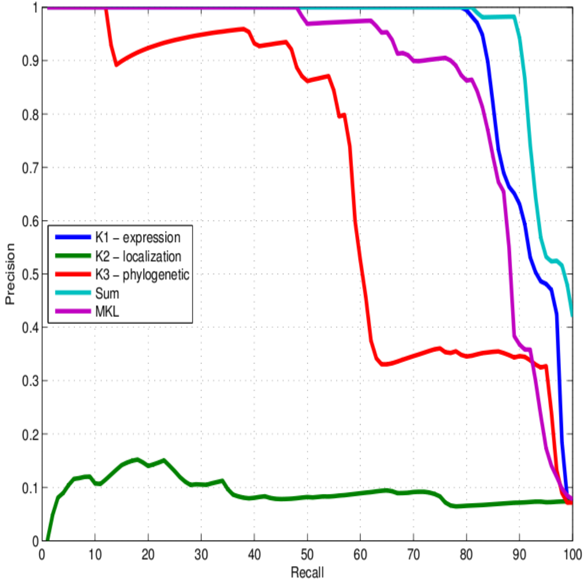
$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \quad s_k \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_2^2}_{Breg.Div.}$$

Mirror Descent

- Similar to projected gradient descent
- Per-step problem has Bregman divergence based regularizer:

$$\mathbf{x}_{k+1} = \text{arg}\min_{\mathbf{x}\in\mathcal{X}} \quad s_k \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{x}_k\|_2^2}_{Breg.Div.}$$

- Regularizer chosen such that per-step problem has closed form solution
  - For simplex geometry, entropy function based reg. can be employed [Ben-Tal & Nemirovski, 01]



Source: [Vert, 09]

MKL leads to sparse selection!

- Analyze the primal view [Bach et.al., 04; Rakotomamonjy et.al., 07]
- Consider $f(\mathbf{x}) = \text{sign}(< \mathbf{w}, \phi(\mathbf{x}) >_{\mathcal{H}} - b) = \text{sign}(\sum_{j=1}^{n} < \mathbf{w}_j, \phi_j(\mathbf{x}) >_{\mathcal{H}_j} - b)$
- MKL is same as:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\left(\sum_{j=1}^{n} \|\mathbf{w}_j\|_{\mathcal{H}_j}\right)^2 + C\sum_{i=1}^{m} \xi_i$$
$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{n} < \mathbf{w}_j, \phi_j(\mathbf{x}) >_{\mathcal{H}_j} - b\right) \geq 1 - \xi_i, \ \xi_i \geq 0$$

MKL leads to sparse selection!

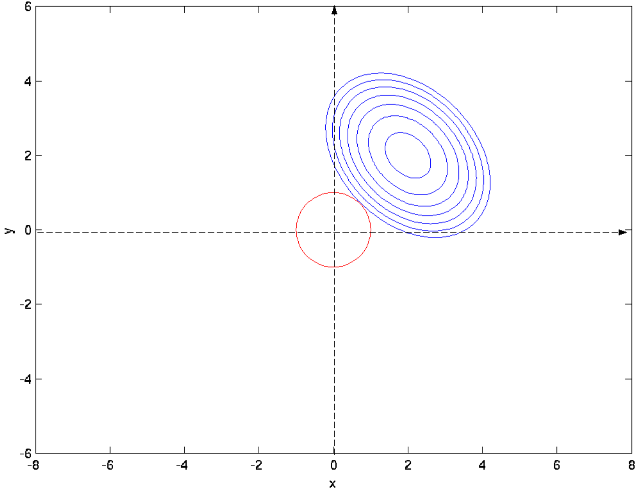
- Analyze the primal view [Bach et.al., 04; Rakotomamonjy et.al., 07]
- Consider $f(\mathbf{x}) = \text{sign}(< \mathbf{w}, \phi(\mathbf{x}) >_{\mathcal{H}} - b) = \text{sign}(\sum_{j=1}^{n} < \mathbf{w}_j, \phi_j(\mathbf{x}) >_{\mathcal{H}_j} - b)$
- MKL is same as:

$$\begin{array}{ll} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}(\sum_{j=1}^{n} \|\mathbf{w}_j\|_{\mathcal{H}_j})^2 + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} & y_i(\sum_{j=1}^{n} < \mathbf{w}_j, \phi_j(\mathbf{x}) >_{\mathcal{H}_j} - b) \geq 1 - \xi_i, \; \xi_i \geq 0 \end{array}$$

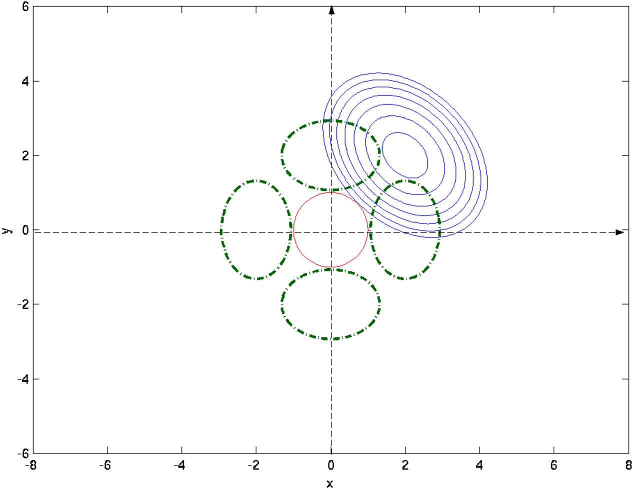
Key observations:

- $\|\mathbf{w}\|_{\mathcal{H}}^2 \neq (\sum_{j=1}^{n} \|\mathbf{w}_j\|_{\mathcal{H}_j})^2$
  - If regularizer were $\|\mathbf{w}\|_{\mathcal{H}}^2$, we would get back SVM i.e. $k = k_1 + k_2 + \ldots + k_n$!
- Current regularizer is l_1, l_2-norm (block lasso) hence promotes sparsity — selection of kernels!

Consider $\min_{\mathbf{x}:\|\mathbf{x}\|_2 \leq 1} f(\mathbf{x})$

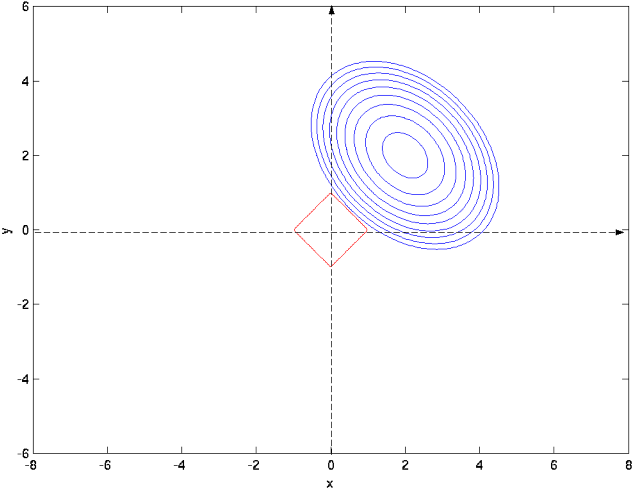


Consider $\min_{\mathbf{x}:\|\mathbf{x}\|_2\leq 1} f(\mathbf{x})$

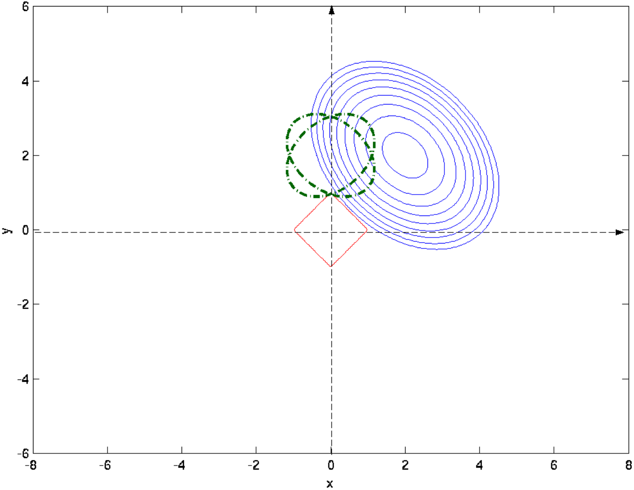


l_1 regularization (Lasso) leads to sparsity

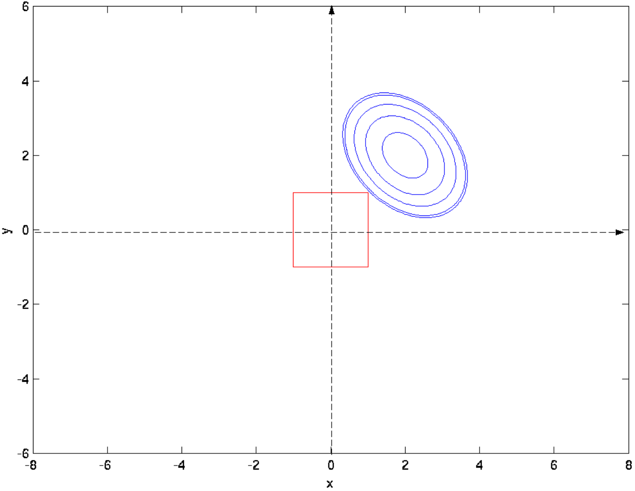
Consider $\min_{\mathbf{x}:\|\mathbf{x}\|_1 \leq 1} f(\mathbf{x})$



Consider $\min_{\mathbf{x}:\|\mathbf{x}\|_1 \leq 1} f(\mathbf{x})$



Consider $\min_{\mathbf{x}:\|\mathbf{x}\|_\infty\leq 1} f(\mathbf{x})$



MKL for non-linear feature selection

- Hierarchical Kernel Learning [Bach, 08]
- Composite Kernel Learning [Szafranski et.al., 08]

MKL for non-linear feature selection

- Hierarchical Kernel Learning [Bach, 08]
- Composite Kernel Learning [Szafranski et.al., 08]
- Multi-class MKL [Zien & Ong, 07]
- Feature Selection for Density Level-Sets [Kloft et.al., 09]

Non-sparse MKL

- l_2-regularization for learning kernels [Cortes et.al., 09]
- l_p-norm multiple kernel learning [Kloft et.al., 09]

[1] http://www.cse.iitb.ac.in/saketh/research.html

Non-sparse MKL

- l_2-regularization for learning kernels [Cortes et.al., 09]
- l_p-norm multiple kernel learning [Kloft et.al., 09]
- MKL for multi-modal tasks [Nath et.al., 09; Nath et.al., 10][1]

[1] http://www.cse.iitb.ac.in/saketh/research.html

Source: [Vert, 09]

- Kernels are generated from different sources (modes)
- Natural grouping:
  - Atleast one kernel in each group in important
  - Not all kernels in a group may be crucial
  - Each source may not be “equally” critical
- Propose an MKL formulation which exploits this group structure!
- Let there be n groups and n_j kernels in j^{th} group

- Kernels are generated from different sources (modes)
- Natural grouping:
  - Atleast one kernel in each group in important
  - Not all kernels in a group may be crucial
  - Each source may not be "equally" critical
- Propose an MKL formulation which exploits this group structure!
- Let there be n groups and n_j kernels in j^{th} group

New regularizer:

$\frac{1}{2}\left\{\sum_{j=1}^{n}\left\{\sum_{k=1}^{n_j}\|\mathbf{w}_{jk}\|_2\right\}^{2q}\right\}^{\frac{1}{q}},\ q \geq 1$

Variable Sparsity Kernel Learning Formulation

Primal form:

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

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

Variable Sparsity Kernel Learning Formulation

Primal form:

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

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

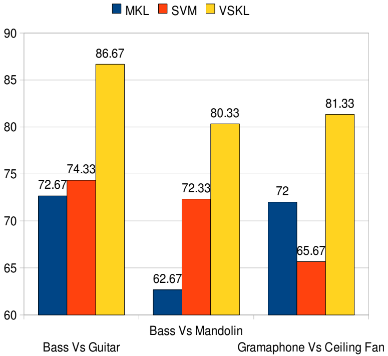
Dual form:

$$\underbrace{\min_{\lambda\in\Delta_{n_j}} \max_{\alpha\in S_m,\gamma\in\Delta_{n,q^*}} \overbrace{\mathbf{1}^T\alpha - \frac{1}{2}\alpha^T\mathbf{Y}\left(\sum_{j=1}^{n}\sum_{k=1}^{n_j}\frac{\lambda_{jk}K_{jk}}{\gamma_j}\right)\mathbf{Y}\alpha}^{f_\lambda(\alpha,\gamma)}}_{G(\lambda)}$$

Mirror descent based solver

- $\min_{\lambda_j \in \Delta_{n_j}} G(\lambda)$ (min. convex function over compact set)
- Entropy function based reg. also works for product of simplices
- Again, Danskin's theorem provides $\nabla G(\lambda)$
  - Need to solve $\max_{\alpha \in S_m, \gamma \in \Delta_{n,q^*}} f_\lambda(\alpha, \gamma)$
  - Alternating minimization alg. with convergence guarantee
  - In practice, solve 4-5 SVM problems
- Overall complexity $O(m^2 n_{tot} \log(n_{max}))$

Performance on Object Categorization



Performance on Object Categorization

	MKL	SVM	CKL	VSKL
Caltech-101	32.25%	33.47%	34.48%	35.62%
Caltech-5	92.76%	93.84%	94.88%	96.12%
Oxford flowers	81.76%	80.12%	80.65%	83.94%

Non-Linear Combinations

- DC-Programming algorithm [Argyriou et al., 05]
- Generalized MKL [Varma & Babu, 09]
- Polynomial combinations [Cortes et.al., 09]

Conclusions and Open Problems

- MKL is a powerful framework for learning kernels
- Great tool for non-linear feature selection
- Promise in combining kernels from multiple modes
- State-of-the-art performance in many applications

Conclusions and Open Problems

- MKL is a powerful framework for learning kernels
- Great tool for non-linear feature selection
- Promise in combining kernels from multiple modes
- State-of-the-art performance in many applications

- In some cases, performance comparable to simple addition of kernels
- Minimization of alternative bounds ?
- Better interpretation of mixed-norm from learning theory view ?
- Non-convexity issues in non-linear combinations of kernels

Questions ?

Thank You