

Learning Kernels -Tutorial

Part I: Introduction to Kernel Methods.

Corinna Cortes
Google Research
corinna@google.com

Mehryar Mohri
Courant Institute &
Google Research
mohri@cims.nyu.edu

Afshin Rostami
UC Berkeley
arostami@eecs.berkeley.edu

Outline

- Part I: Introduction to kernel methods.
- Part II: Learning kernel algorithms.
- Part III: Theoretical guarantees.
- Part IV: Software tools.

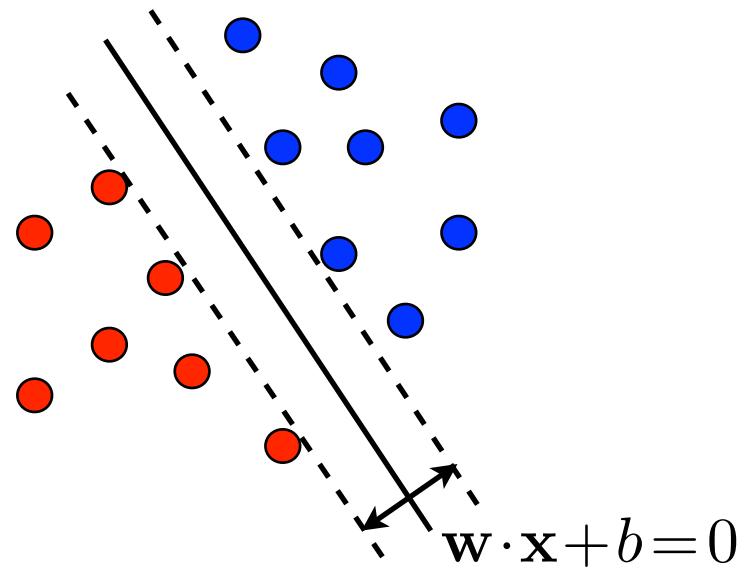
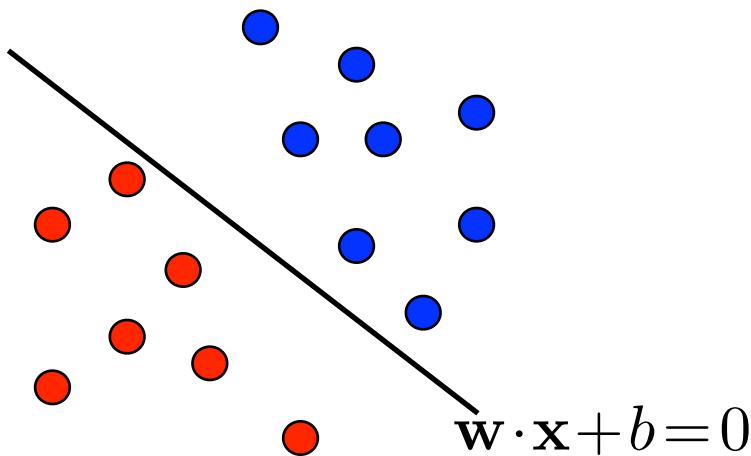
Binary Classification Problem

- **Training data:** sample drawn i.i.d. from set $X \subseteq \mathbb{R}^N$ according to some distribution D ,

$$S = ((x_1, y_1), \dots, (x_m, y_m)) \in X \times \{-1, +1\}.$$

- **Problem:** find hypothesis $h : X \mapsto \{-1, +1\}$ in H (classifier) with small generalization error $R_D(h)$.
- **Linear classification:**
 - Hypotheses based on hyperplanes.
 - Linear separation in high-dimensional space.

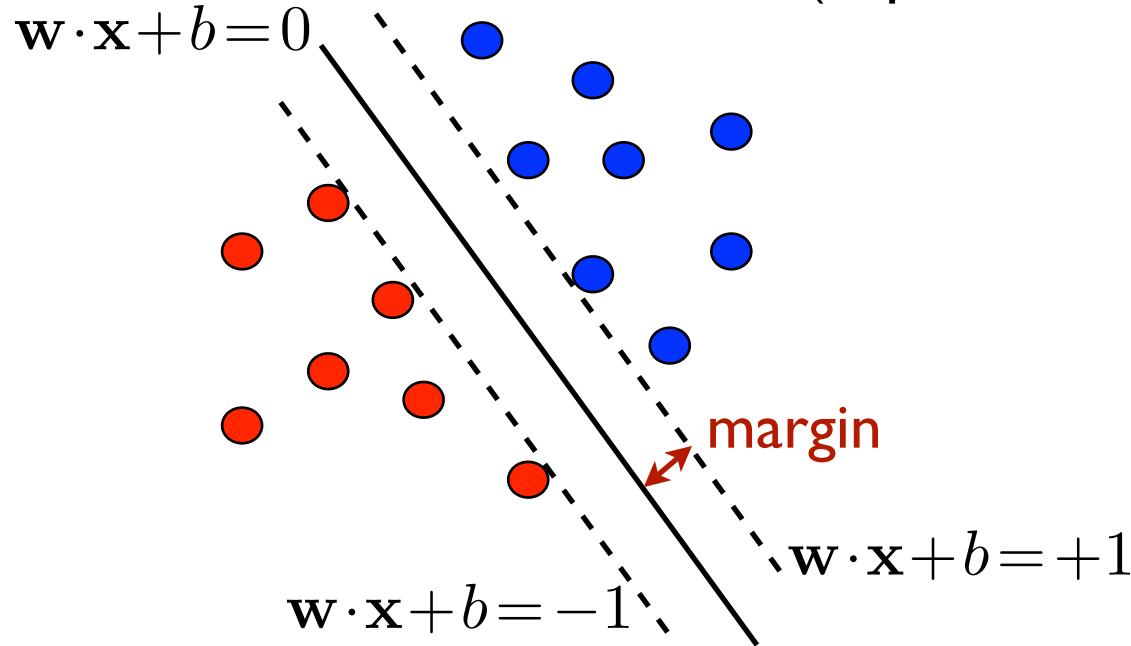
Linear Separation



- **Classifiers:** $H = \{x \mapsto \text{sgn}(w \cdot x + b) : w \in \mathbb{R}^N, b \in \mathbb{R}\}$.

Optimal Hyperplane: Max. Margin

(Vapnik and Chervonenkis, 1964)



- **Canonical hyperplane:** w and b chosen such that for closest points $|w \cdot x + b| = 1$.
- **Margin:** $\rho = \min_{x \in S} \frac{|w \cdot x + b|}{\|w\|} = \frac{1}{\|w\|}$.

Optimization Problem

■ Constrained optimization:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i \in [1, m]$.

■ Properties:

- Convex optimization (strictly convex).
- Unique solution for linearly separable sample.

Support Vector Machines

(Cortes and Vapnik, 1995)

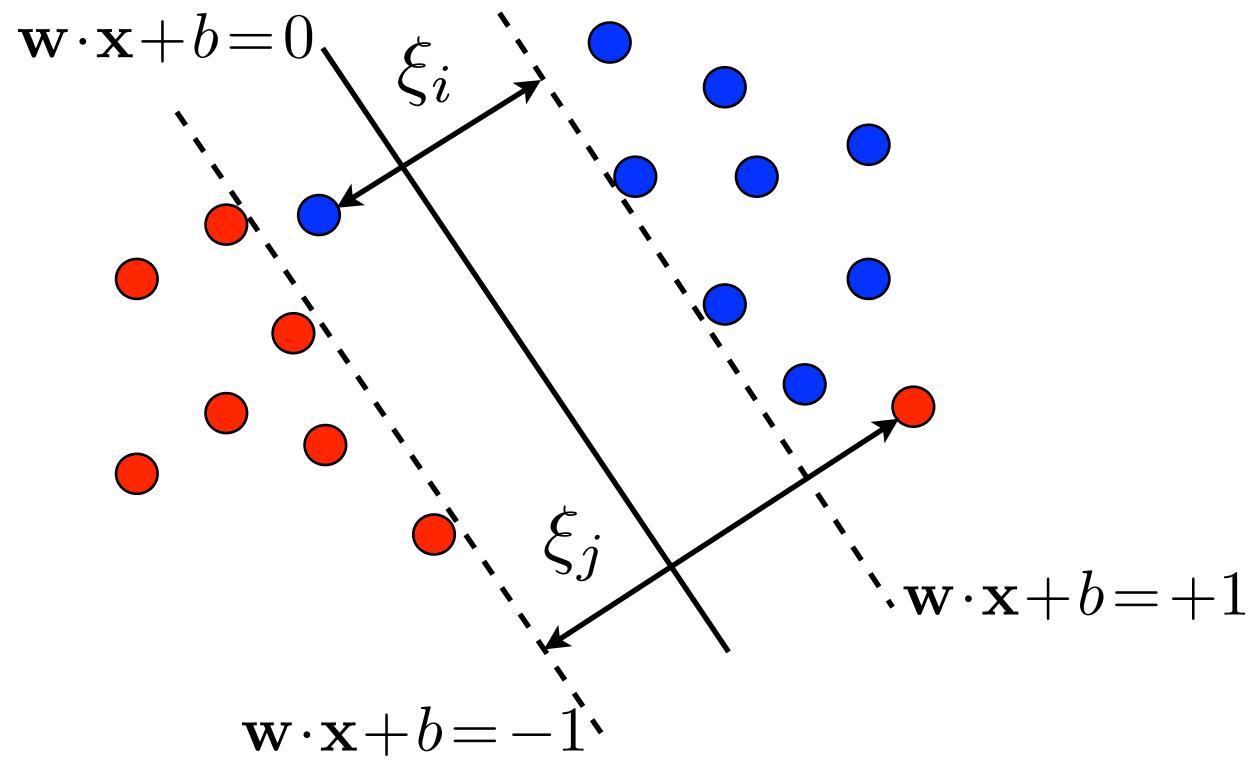
- **Problem:** data often not linearly separable in practice. For any hyperplane, there exists \mathbf{x}_i such that

$$y_i [\mathbf{w} \cdot \mathbf{x}_i + b] \not\geq 1.$$

- **Idea:** relax constraints using **slack variables** $\xi_i \geq 0$

$$y_i [\mathbf{w} \cdot \mathbf{x}_i + b] \geq 1 - \xi_i.$$

Soft-Margin Hyperplanes



- **Support vectors:** points along the margin or outliers.
- **Soft margin:** $\rho = 1/\|w\|$.

Optimization Problem

(Cortes and Vapnik, 1995)

■ Constrained optimization:

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

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \wedge \xi_i \geq 0, i \in [1, m]$.

■ Properties:

- $C \geq 0$ trade-off parameter.
- Convex optimization (strictly convex).
- Unique solution.

Dual Optimization Problem

■ Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

subject to: $0 \leq \alpha_i \leq C \wedge \sum_{i=1}^m \alpha_i y_i = 0, i \in [1, m]$.

■ Solution:

$$h(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b\right),$$

with $b = y_i - \sum_{j=1}^m \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ **for any** \mathbf{x}_i **with**
 $0 < \alpha_i < C$.

Kernel Methods

■ Idea:

- Define $K : X \times X \rightarrow \mathbb{R}$, called **kernel**, such that:

$$\Phi(x) \cdot \Phi(y) = K(x, y).$$

- K often interpreted as a similarity measure.

■ Benefits:

- **Efficiency:** K is often more efficient to compute than Φ and the dot product.
- **Flexibility:** K can be chosen arbitrarily so long as the existence of Φ is guaranteed (Mercer's condition).

Example - Polynomial Kernels

■ Definition:

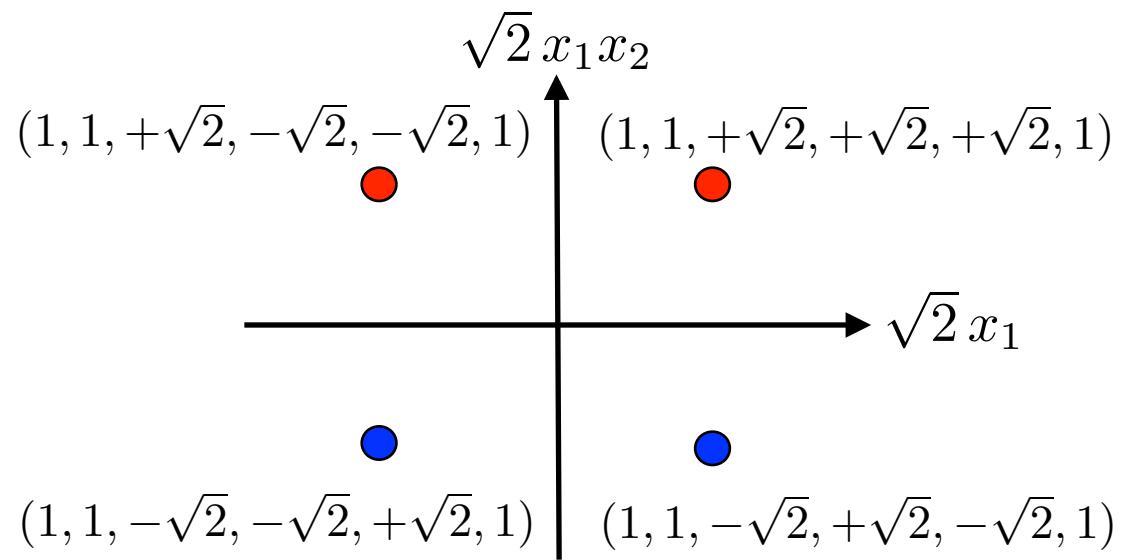
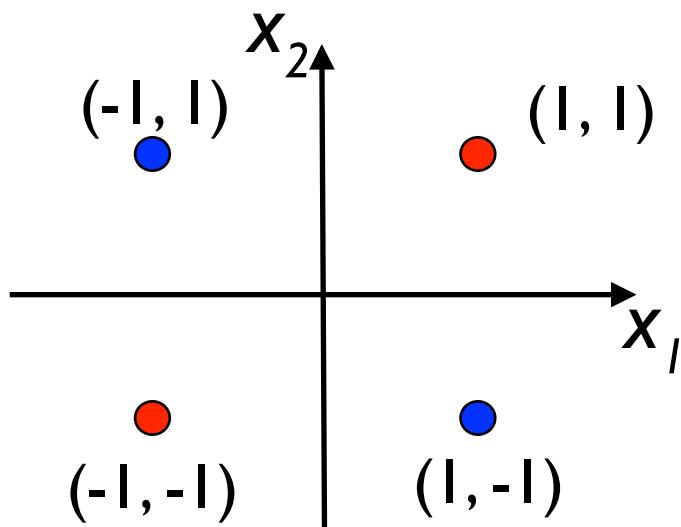
$$\forall x, y \in \mathbb{R}^N, K(x, y) = (x \cdot y + c)^d, \quad c > 0.$$

■ Example: for $N=2$ and $d=2$,

$$K(x, y) = (x_1 y_1 + x_2 y_2 + c)^2 \\ = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2c} x_1 \\ \sqrt{2c} x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2} y_1 y_2 \\ \sqrt{2c} y_1 \\ \sqrt{2c} y_2 \\ c \end{bmatrix}.$$

XOR Problem

- Use second-degree polynomial kernel with $c = 1$:



Linearly non-separable

Linearly separable by
 $x_1x_2 = 0$.

Other Standard PDS Kernels

- Gaussian kernels:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \quad \sigma \neq 0.$$

- Sigmoid Kernels:

$$K(x, y) = \tanh(a(x \cdot y) + b), \quad a, b \geq 0.$$

Consequence: SVMs with PDS Kernels

(Boser, Guyon, and Vapnik, 1992)

■ Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to: $0 \leq \alpha_i \leq C \wedge \sum_{i=1}^m \alpha_i y_i = 0, i \in [1, m]$.

■ Solution:

$$h(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i K(x_i, x) + b\right),$$

with $b = y_i - \sum_{j=1}^m \alpha_j y_j K(x_j, x_i)$ for any x_i with $0 < \alpha_i < C$.

SVMs with PDS Kernels

■ Constrained optimization:

$$\begin{aligned} & \max_{\alpha} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \alpha \\ \text{subject to: } & \mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0. \end{aligned}$$

■ Solution:

$$h = \operatorname{sgn}\left(\sum_{i=1}^m \alpha_i y_i K(x_i, \cdot) + b\right),$$

with $b = y_i - (\alpha \circ \mathbf{y})^\top \mathbf{K} \mathbf{e}_i$ for any x_i with
 $0 < \alpha_i < C$.

Regression Problem

- **Training data:** sample drawn i.i.d. from set X according to some distribution D ,

$$S = ((x_1, y_1), \dots, (x_m, y_m)) \in X \times Y,$$

with $Y \subseteq \mathbb{R}$ is a measurable subset.

- **Loss function:** $L: Y \times Y \rightarrow \mathbb{R}_+$ a measure of closeness, typically $L(y, y') = (y' - y)^2$ or $L(y, y') = |y' - y|^p$ for some $p \geq 1$.
- **Problem:** find hypothesis $h: X \rightarrow \mathbb{R}$ in H with small generalization error with respect to target f

$$R_D(h) = \mathbb{E}_{x \sim D} [L(h(x), f(x))].$$

Kernel Ridge Regression

(Saunders et al., 1998)

■ Optimization problem:

$$\max_{\alpha \in \mathbb{R}^m} -\lambda \alpha^\top \alpha + 2\alpha^\top \mathbf{y} - \alpha^\top \mathbf{K} \alpha$$

or $\max_{\alpha \in \mathbb{R}^m} -\alpha^\top (\mathbf{K} + \lambda \mathbf{I}) \alpha + 2\alpha^\top \mathbf{y}.$

■ Solution:

$$h(x) = \sum_{i=1}^m \alpha_i K(x_i, x),$$

with $\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}.$

Questions

- How should the user choose the kernel?
 - problem similar to that of selecting features for other learning algorithms.
 - poor choice → learning made very difficult.
 - good choice → even poor learners could succeed.
- The requirement from the user is thus critical.
 - can this requirement be lessened?
 - is a more automatic selection of features possible?

Learning Kernels -Tutorial

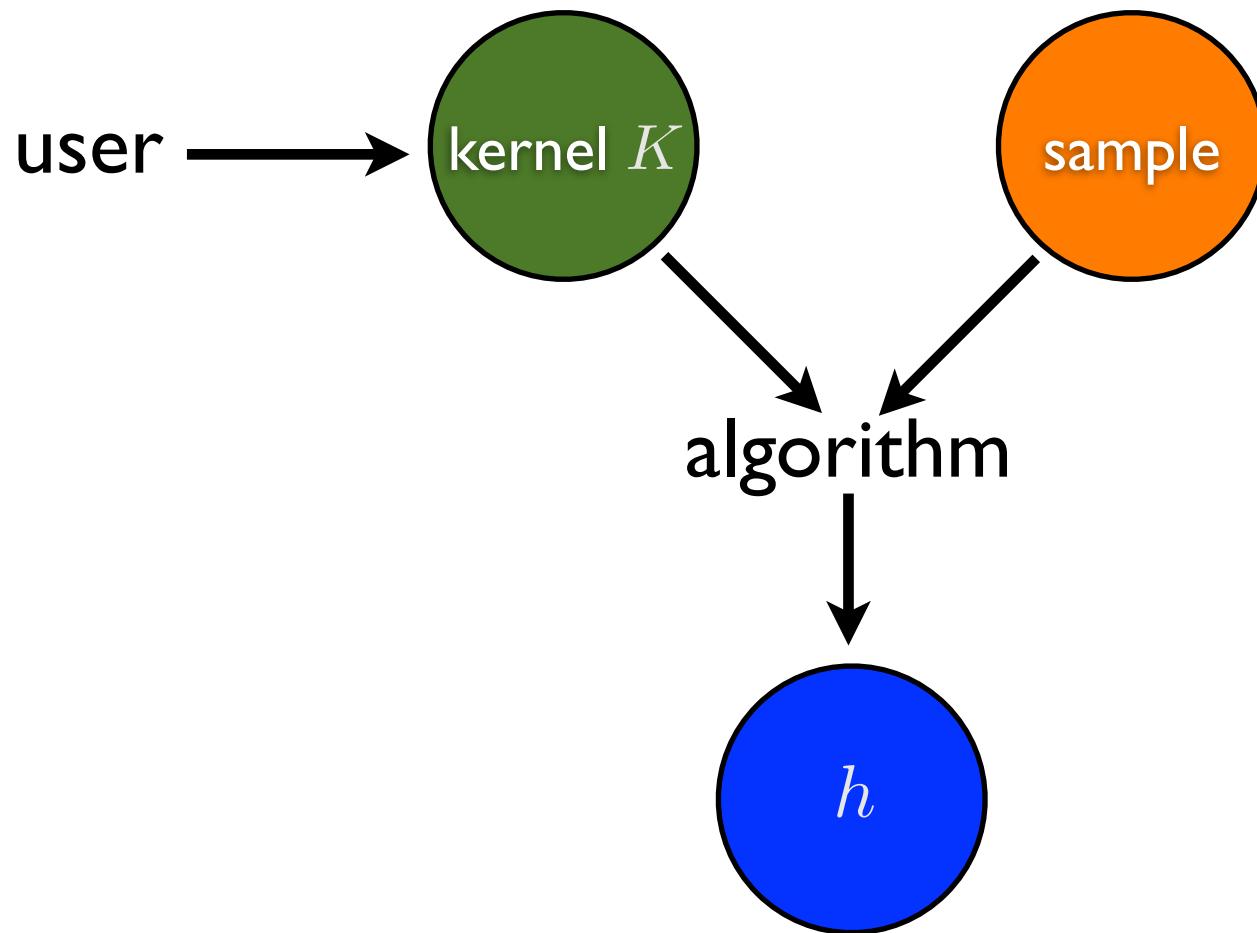
Part II: Learning Kernel Algorithms.

Corinna Cortes
Google Research
corinna@google.com

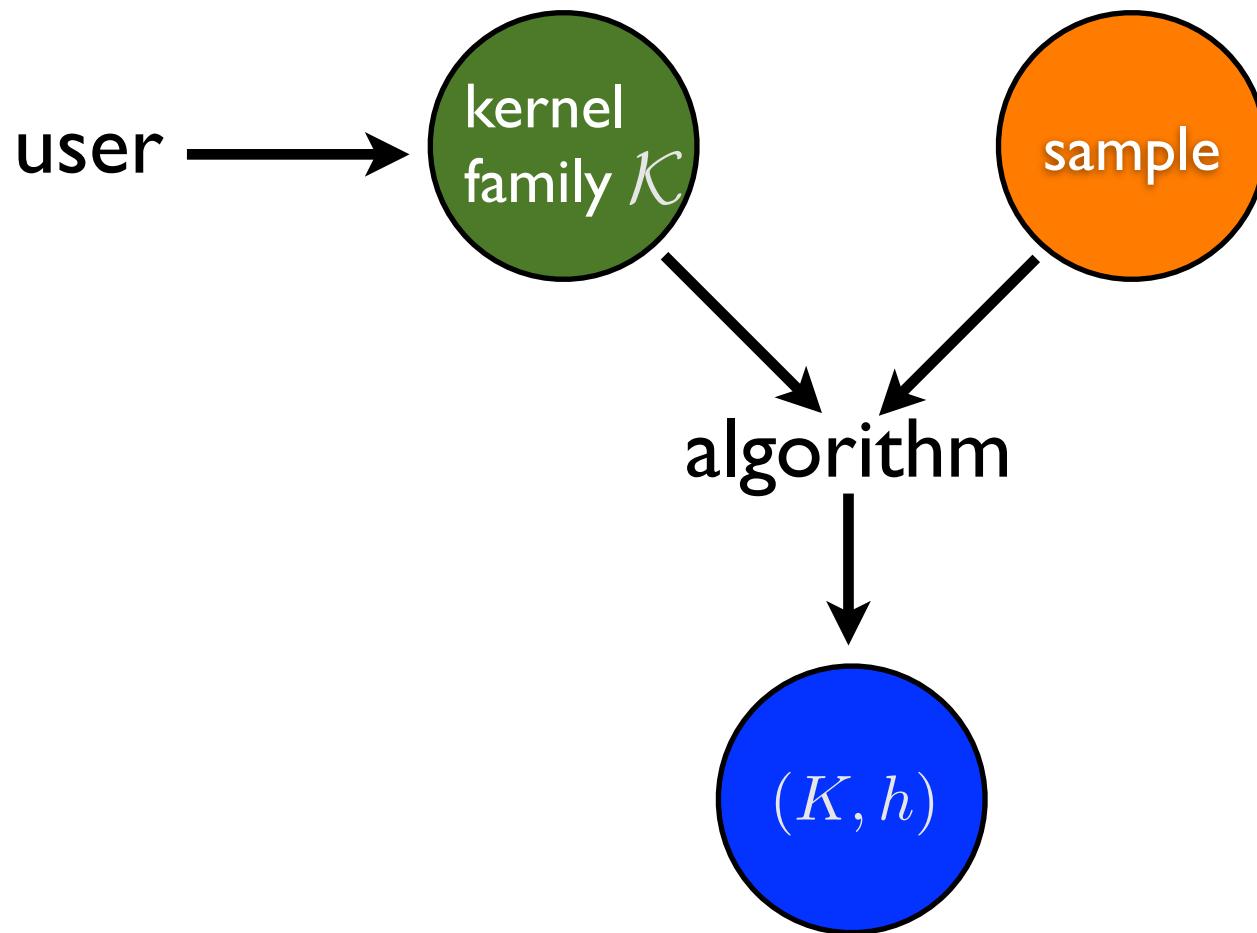
Mehryar Mohri
Courant Institute &
Google Research
mohri@cims.nyu.edu

Afshin Rostami
UC Berkeley
arostami@eecs.berkeley.edu

Standard Learning with Kernels



Learning Kernel Framework



This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

Minimize Different Criteria

(Weston et al., 2000; Chapelle et al., 2002)

- **Wrapper method:** alternate a call to an SVM solver and an update of the kernel parameters.

- solve SVM to get α^*
- gradient step over criterion T to select kernel parameters:
 - margin criterion $T = R^2/\rho^2$
 - span criterion $T = \frac{1}{m} \sum_{i=1}^m \Theta(\alpha_i^* S_i^2 - 1)$.

Reality Check

(Chapelle et al., 2002)

Selecting the width of a Gaussian kernel and the SVM parameter C.

Accuracy:

	Cross-validation	R^2/γ^2	Span-bound
Breast cancer	26.04 ± 4.74	26.84 ± 4.71	25.59 ± 4.18
Diabetis	23.53 ± 1.73	23.25 ± 1.7	23.19 ± 1.67
Heart	15.95 ± 3.26	15.92 ± 3.18	16.13 ± 3.11
Thyroid	4.80 ± 2.19	4.62 ± 2.03	4.56 ± 1.97
Titanic	22.42 ± 1.02	22.88 ± 1.23	22.5 ± 0.88

Speed:

	Cross-validation	R^2/γ^2	Span-bound
Breast cancer	500	14.2	7
Diabetis	500	12.2	9.8
Heart	500	9	6.2
Thyroid	500	3	11.6
Titanic	500	6.8	3.4

Kernel Learning & Feature Selection

■ Linear kernels:

$$K(x_i, x_j) = \sum_{k=1}^p \mu_k x_i^k x_j^k, \quad \mu_k \geq 0, \quad \sum_{k=1}^p (\mu_k)^q \leq \Lambda$$

■ Polynomial kernels:

$$K(x_i, x_j) = \left(1 + \sum_{k=1}^p \mu_k x_i^k x_j^k\right)^d, \quad \mu_k \geq 0, \quad \sum_{k=1}^p (\mu_k)^q \leq \Lambda$$

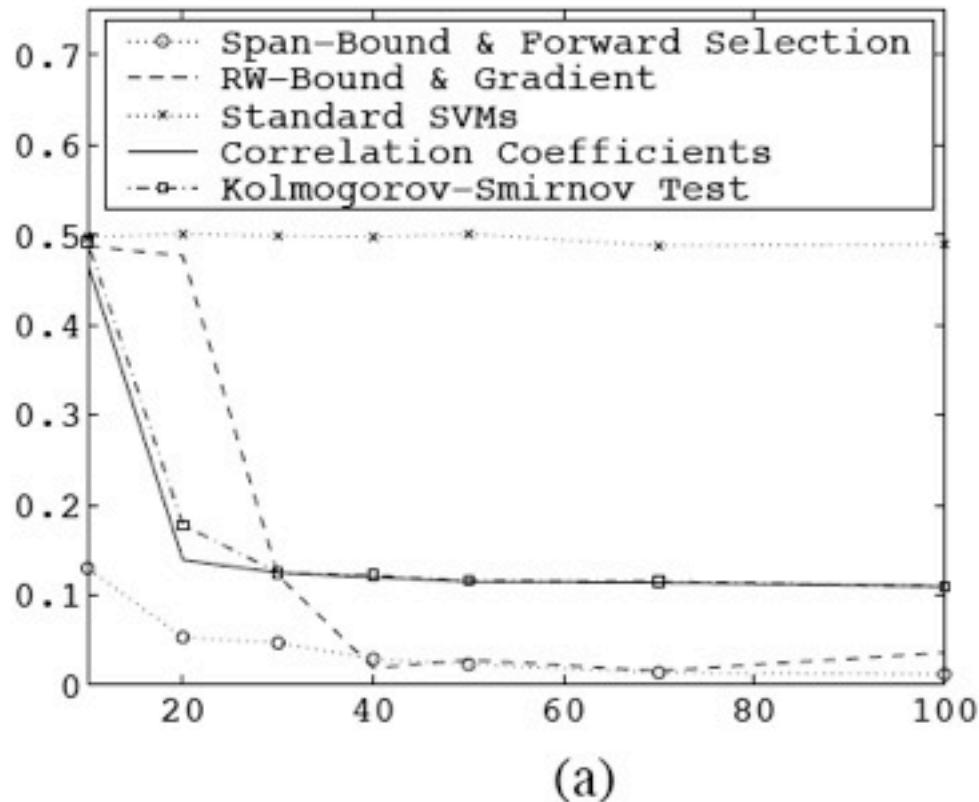
■ Alternate between solving SVM and gradient step

- the margin bound: R^2/ρ^2 , (Weston et al., 2000)
- the SVM dual: $2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha$, (Grandvalet & Canu, 2003).

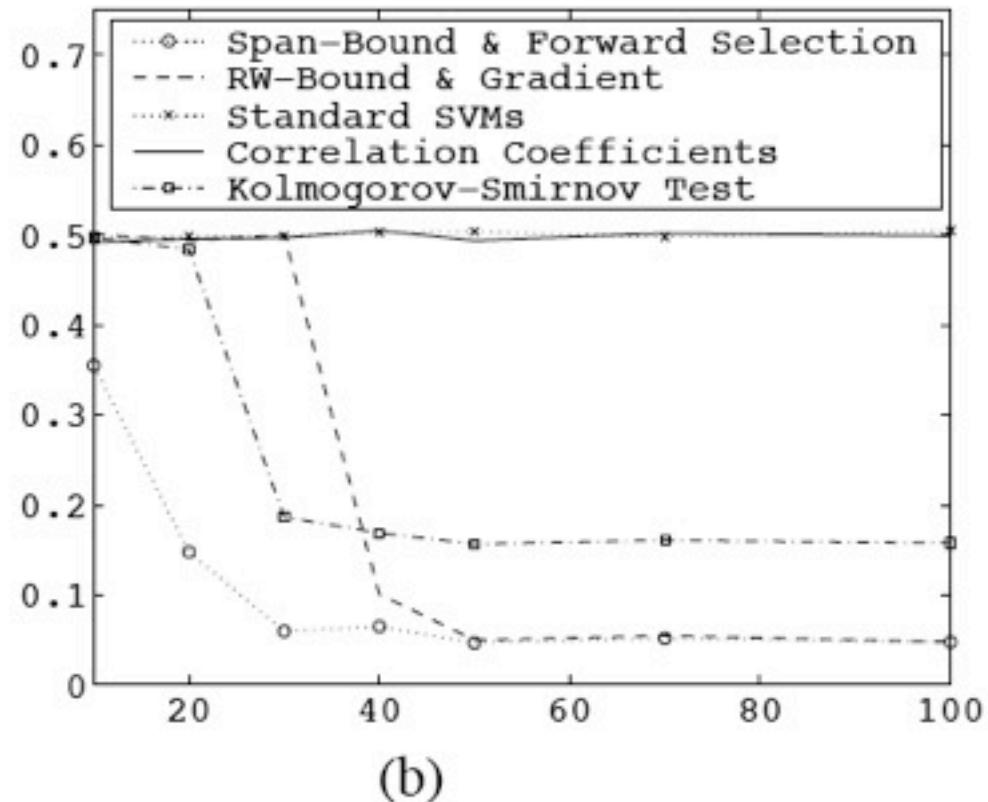
Feature Selection: Reality Check

(Weston et al., 2000; Chapelle et al., 2002)

■ Comparison with existing methods:



(a)



(b)

Figure 1: A comparison of feature selection methods on (a) a linear problem and (b) a nonlinear problem both with many irrelevant features. The x -axis is the number of training points, and the y -axis the test error as a fraction of test points.

This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

Overview

■ LK formulations:

- (Lanckriet et al., 2004): SVM, L_1 regularization, general, linear, or non-negative combinations.
- (Cortes et al., 2009): KRR, L_2 regularization, non-negative combinations.
- (Kloft et al., 2009): SVM, L_p regularization, linear, or non-negative combinations.

General LK Formulation - SVMs

■ Notation:

- \mathcal{K} set of PDS kernel functions.
- $\overline{\mathcal{K}}$ kernel matrices associated to \mathcal{K} , assumed convex.
- $\mathbf{Y} \in \mathbb{R}^{m \times m}$ diagonal matrix with $\mathbf{Y}_{ii} = \mathbf{y}_i$.

■ Optimization problem:

$$\min_{\mathbf{K} \in \overline{\mathcal{K}}} \max_{\boldsymbol{\alpha}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \boldsymbol{\alpha}$$

subject to: $\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0$.

- convex problem: function linear in \mathbf{K} , convexity of pointwise maximum.

General LK Formulation - SVMs

- Consider the maximization problem:

$$\begin{aligned} & \max_{\alpha} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \alpha \\ \text{subject to: } & \mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0. \end{aligned}$$

- The corresponding Lagrange function is

$$L = 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \alpha + 2\beta^\top \alpha - 2\gamma^\top (\alpha - \mathbf{C}) - 2\delta \alpha^\top \mathbf{y}.$$

$$\text{and } \nabla_\alpha L = 0 \iff \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \alpha = \mathbf{1} + \beta - \gamma - \delta \mathbf{y}.$$

- Thus, $(\mathbf{Y}^\top \mathbf{K} \mathbf{Y})^\dagger (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})$ is one solution.
Plugging that in gives the dual problem

$$\min_{\beta \geq \mathbf{0}, \gamma \geq \mathbf{0}, \delta} (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})^\top (\mathbf{Y}^\top \mathbf{K} \mathbf{Y})^\dagger (\mathbf{1} + \beta - \gamma - \delta \mathbf{y}) + 2\gamma^\top \mathbf{C}.$$

General LK Formulation - SVMs

- The problem can now be rewritten as

$$\min_{t, \beta, \gamma, \delta} t$$

subject to: $t \geq (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})^\top (\mathbf{Y}^\top \mathbf{K} \mathbf{Y})^\dagger (\mathbf{1} + \beta - \gamma - \delta \mathbf{y}) + 2\gamma^\top \mathbf{C}$
 $(\beta \geq \mathbf{0}) \wedge (\gamma \geq \mathbf{0}).$

- Now, by the property of the Schur complement with a singular matrix (Boyd and Vandenberghe, 2004), this is equivalent to

$$\min_{t, \beta, \gamma, \delta} t$$

subject to: $\begin{bmatrix} \mathbf{Y}^\top \mathbf{K} \mathbf{Y} & \mathbf{1} + \beta - \gamma - \delta \mathbf{y} \\ (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})^\top & t - 2\gamma^\top \mathbf{C} \end{bmatrix} \succeq \mathbf{0}$
 $(\beta \geq \mathbf{0}) \wedge (\gamma \geq \mathbf{0}).$

General LK Formulation - SVMs

■ Optimization problem:

$$\begin{array}{ll} \min_{\mathbf{K} \in \overline{\mathcal{K}}, t, \beta, \gamma, \delta} & t \\ \text{subject to:} & \left[\begin{array}{cc} \mathbf{Y}^\top \mathbf{K} \mathbf{Y} & 1 + \beta - \gamma - \delta \mathbf{y} \\ (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})^\top & t - 2 \gamma^\top \mathbf{C} \end{array} \right] \succeq \mathbf{0} \\ & (\beta \geq \mathbf{0}) \wedge (\gamma \geq \mathbf{0}). \end{array}$$

- the minimization over t, β, γ, δ is a semi-definite program (SDP).
- if $\overline{\mathcal{K}} = \{\mathbf{K}: (\mathbf{K} \succeq \mathbf{0}) \wedge \text{Tr}[\mathbf{K}] = 1\}$ the full program is an SDP.

Notes

■ Comments on (Lanckriet et al., 2004):

- full proof that problem is equivalent to an SDP not given. The proof given implicitly assumes $(K + \tau I)$ invertible for $\tau \geq 0$ which in general does not hold. In particular, for $\tau = 0$, K is in general not invertible.
- the paper deals exclusively with transductive scenario. Thus, instead of minimizing over kernel functions, it minimizes over kernel matrices.
- paper has been the basis for large part of the work done in LK area.

Parameterized LK Formulation

■ Notation:

- $(K_\mu)_{\mu \in \Delta}$ parameterized set of PDS kernel functions.
- Δ convex set, $\mu \mapsto K_\mu$ concave function.
- $Y \in \mathbb{R}^{m \times m}$ diagonal matrix with $Y_{ii} = y_i$.

■ Optimization problem:

$$\min_{\mu \in \Delta} \max_{\alpha} 2\alpha^\top 1 - \alpha^\top Y^\top K_\mu Y \alpha$$

subject to: $0 \leq \alpha \leq C \wedge \alpha^\top y = 0$.

- convex problem: function convex in μ , convexity of pointwise maximum.

Linear Combinations

- $p \geq 1$ base PDS kernel functions K_1, \dots, K_p .
- Kernel family:

$$\mathcal{K}_{\text{lin}} = \left\{ K_{\boldsymbol{\mu}} = \sum_{k=1}^p \mu_k K_k : \boldsymbol{\mu} \in \Delta_{\text{lin}} \right\}$$

with $\Delta_{\text{lin}} = \left\{ \boldsymbol{\mu} \in \mathbb{R}^p : \sum_{k=1}^p \mu_k = 1 \wedge \mathbf{K}_{\boldsymbol{\mu}} \succeq \mathbf{0} \right\}.$

- Hypothesis sets:

$$H_{\text{lin}} = \left\{ h \in \mathbb{H}_K : K \in \mathcal{K}_{\text{lin}}, \|h\|_{\mathbb{H}_K} \leq 1 \right\}.$$

Linear Combinations

(Lanckriet et al., 2004)

- Assuming trace-normalized base kernel matrices:

$$\text{Tr}[\mathbf{K}_\mu] = \sum_{k=1}^p \mu_k \text{Tr}[\mathbf{K}_k] = \sum_{k=1}^p \mu_k.$$

- Optimization problem: semi-definite program (SDP).

$$\begin{aligned} & \min_{\mu, t} \quad t \\ \text{subject to: } & \begin{bmatrix} \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} & 1 + \beta - \gamma - \delta \mathbf{y} \\ (\mathbf{1} + \beta - \gamma - \delta \mathbf{y})^\top & t - 2\gamma^\top \mathbf{C} \end{bmatrix} \succeq 0 \\ & (\beta \geq \mathbf{0}) \wedge (\gamma \geq \mathbf{0}) \\ & \left(\sum_{k=1}^p \mu_k = 1 \right) \wedge \left(\mathbf{K}_\mu = \sum_{k=1}^p \mu_k \mathbf{K}_k \right) \wedge \left(\mathbf{K}_\mu \succeq \mathbf{0} \right). \end{aligned}$$

Non-Negative Combinations

- $p \geq 1$ base PDS kernel functions K_1, \dots, K_p .
- Kernel family:

$$\mathcal{K}_q = \left\{ K_{\mu} = \sum_{k=1}^p \mu_k K_k : \mu \in \Delta_q \right\}$$

with $\Delta_q = \left\{ \mu \in \mathbb{R}^p : \|\mu\|_q \leq 1, \mu \geq \mathbf{0} \right\}$.

- Hypothesis sets:

$$H_q = \left\{ h \in \mathbb{H}_K : K \in \mathcal{K}_q, \|h\|_{\mathbb{H}_K} \leq 1 \right\}.$$

Non-Negative Combinations

- By von Neumann's generalized minimax theorem (convexity wrt μ , concavity wrt α , Δ_1 convex and compact, \mathcal{A} convex and compact):

$$\begin{aligned} & \min_{\mu \in \Delta_1} \max_{\alpha \in \mathcal{A}} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha \\ &= \max_{\alpha \in \mathcal{A}} \min_{\mu \in \Delta_1} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha \\ &= \max_{\alpha \in \mathcal{A}} 2\alpha^\top \mathbf{1} - \max_{\mu \in \Delta_1} \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha \\ &= \max_{\alpha \in \mathcal{A}} 2\alpha^\top \mathbf{1} - \max_{k \in [1, p]} \alpha^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \alpha. \end{aligned}$$

Non-Negative Combinations

(Lanckriet et al., 2004)

- **Optimization problem:** in view of the previous analysis, the problem can be rewritten as the following QCQP.

$$\max_{\alpha, t} 2\alpha^\top \mathbf{1} - t$$

$$\text{subject to: } \forall k \in [1, p], t \geq \alpha^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \alpha;$$
$$\mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0.$$

- complexity (interior-point methods): $O(pm^3)$.

N-Neg. Comb. - Primal Formulation

- Optimization problem: equivalent primal.

$$\min_{w, \mu \in \Delta_q} \frac{1}{2} \sum_{k=1}^p \frac{\|\mathbf{w}_k\|_2^2}{\mu_k} + \sum_{i=1}^m \max \left\{ 0, 1 - y_i \left(\sum_{k=1}^p \mathbf{w}_k^\top \Phi_k(x_i) \right) \right\}.$$

Rank-One Base Kernels

- Optimization problem: reduces to simple QP.

$$\max_{\alpha} 2\alpha^\top \mathbf{1} - t^2$$

subject to: $\forall k \in [1, p], -t \leq \alpha^\top \mathbf{Y}^\top \mathbf{X}_k \leq t;$

$$\mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0.$$

- $\mathbf{K}_k = \mathbf{X}_k \mathbf{X}_k^\top$.
- application to learning sequence kernels (Cortes et al., 2008).

Solving Non-Negative Combinations

- **Wrapper methods:** interleaving a call to an SVM solver and an update of the kernel parameters.
- **Beyond wrapper methods:** methods that avoid the call to the SVM solver.
- **SMO methods:** methods that re-write the SVM solver and find the optimal kernel parameters.
- **Experimental comparison.**

Wrapper Methods

- Alternate steps between solving the SVM and updating the kernel parameters using:
 - SILP
 - Steepest descent
 - Reduced gradient
 - Newton's method
 - Mirror descent

SILP

■ What is a Semi-Infinite Linear Program?

$$\max_{\mathbf{y}} \mathbf{b}^\top \mathbf{y}$$

$$\text{subject to: } \mathbf{a}_\alpha^\top \mathbf{y} \leq c_\alpha, \forall \alpha \in \mathcal{A},$$

- where $\mathbf{y}, \mathbf{b}, \mathbf{a}_\alpha \in \mathbb{R}^m$, $c_\alpha \in \mathbb{R}$, and $\alpha \in \mathcal{A}$, with \mathcal{A} typically a compact (infinite) set.
- Efficient for large-scale problems when used with constraint generating methods.

SILP

- QCQP for non-negative combinations rewritten as (changing sign in objective function):

$$\max_{\beta} \min_{\alpha} \sum_{k=1}^p \beta_k (\alpha^\top Y^\top K_k Y \alpha - 2\alpha^\top 1)$$

subject to: $(\sum_{k=1}^p \beta_k = 1) \wedge (\beta \geq 0)$

$$(\mathbf{0} \leq \alpha \leq \mathbf{C}) \wedge (\alpha^\top \mathbf{y} = 0).$$

SILP - Formulation

(Sonnenburg et al., 2006)

- Optimization problem: semi-infinite linear program (SILP), e.g., LP with infinitely many constraints.

$$\begin{aligned} & \max_{\beta, \theta} \theta \\ \text{subject to: } & \theta \leq \sum_{k=1}^p \beta_k (\alpha^\top Y^\top K_k Y \alpha - 2\alpha^\top \mathbf{1}) \\ & \left(\sum_{k=1}^p \beta_k = 1 \right) \wedge (\beta \geq \mathbf{0}) \\ & (\mathbf{0} \leq \alpha \leq \mathbf{C}) \wedge (\alpha^\top \mathbf{y} = 0). \end{aligned}$$

SILP - Algorithm

(Sonnenburg et al., 2006)

- **Algorithm:** repeat following operations.
 - solve LP with finite number of constraints.
 - add new (most violating constraint), that is for a fixed β , find $\alpha \in \mathcal{A}$ minimizing
$$\sum_{k=1}^p \beta_k (\alpha^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \alpha - 2\alpha^\top \mathbf{1}) = \alpha^\top \mathbf{Y}^\top \mathbf{K}_\beta \mathbf{Y} \alpha - 2\alpha^\top \mathbf{1},$$
which coincides with solving dual SVM.
- Many other heuristics: e.g., chunking for SVM problem, removing inactive constraints for LP.
- No clear convergence rate guarantee, but handles large samples (e.g., 1M points, 20 kernels).

Reduced Gradient

■ Optimization problem:

$$\min_{\mu \in \Delta} \max_{\alpha} 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha$$

subject to: $\mathbf{0} \leq \alpha \leq \mathbf{C} \wedge \alpha^\top \mathbf{y} = 0$.

- Kernel family: $\mathcal{K} = \left\{ K_\mu = \sum_{k=1}^p \mu_k K_k : \mu \in \Delta \right\}$.

■ Reduced gradient:

Let $J = 2\alpha^\top \mathbf{1} - \alpha^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \alpha$

$$\nabla_{red} J_k = \frac{\partial J}{\partial \mu_k} - \frac{\partial J}{\partial \mu_m}, k \neq m \quad \nabla_{red} J_m = \sum_{k \neq m} \left(\frac{\partial J}{\partial \mu_m} - \frac{\partial J}{\partial \mu_k} \right).$$

Reduced Gradient: SimpleMKL

(Rakotomamonjy et al., 2008)

■ SimpleMKL algorithm

Algorithm 1 SimpleMKL algorithm

set $d_m = \frac{1}{M}$ for $m = 1, \dots, M$

while stopping criterion not met **do**

 compute $J(d)$ by using an SVM solver with $K = \sum_m d_m K_m$

 compute $\frac{\partial J}{\partial d_m}$ for $m = 1, \dots, M$ and descent direction D (12).

 set $\mu = \operatorname{argmax}_m d_m, J^\dagger = 0, d^\dagger = d, D^\dagger = D$

while $J^\dagger < J(d)$ **do** {descent direction update}

$d = d^\dagger, D = D^\dagger$

$v = \operatorname{argmin}_{\{m|D_m < 0\}} -d_m/D_m, \gamma_{\max} = -d_v/D_v$

$d^\dagger = d + \gamma_{\max} D, D_\mu^\dagger = D_\mu - D_v, D_v^\dagger = 0$

 compute J^\dagger by using an SVM solver with $K = \sum_m d_m^\dagger K_m$

end while

 line search along D for $\gamma \in [0, \gamma_{\max}]$ {calls an SVM solver for each γ trial value}

$d \leftarrow d + \gamma D$

end while

Newton's Method

■ Optimization problem:

$$\min_{\mu \in \Delta} F(\mu)$$

■ Approximate F :

$$G_t(\mu) = F(\mu^t) + (\mu - \mu^t)^\top \nabla_\mu F(\mu)|_{\mu^t} + \frac{1}{2}(\mu - \mu^t)^\top \underbrace{\nabla_\mu^2 F(\mu)|_{\mu^t}}_{\mathbf{H}(\mu^t)}(\mu - \mu^t).$$

■ Solving for μ :

$$\begin{aligned} \nabla G_t(\mu) = 0 &\Leftrightarrow \nabla_\mu F(\mu)|_{\mu^t} + \mathbf{H}(\mu^t)(\mu - \mu^t) = 0 \\ &\Leftrightarrow \Delta\mu = -\mathbf{H}^{-1}(\mu^t)\nabla F(\mu)|_{\mu^t}. \end{aligned}$$

Newton's Method: L_q-Norm

(Kloft et al., 2009)

■ Optimization problem:

$$\min_{\mu \in \Delta_q, \mathbf{w}, b, \xi} \frac{1}{2} \sum_{k=1}^p \frac{\|\mathbf{w}_k\|_2^2}{\mu_k} + C\|\xi\|_1$$

subject to: $\forall i \ y_i \left(\sum_{k=1}^p \mathbf{w}_k^\top \psi(\mathbf{x})_i + b \right) \geq 1 - \xi_i, \xi \geq \mathbf{0}, \mu \geq \mathbf{0}, \|\mu\|_q^q \leq 1.$

- Kernel family $\mathcal{K}_q = \left\{ K_\mu = \sum_{k=1}^p \mu_k K_k : \mu \in \Delta_q \right\}.$

■ Lagrange function:

$$L = \sum_{k=1}^p \frac{\|\mathbf{w}_k\|_2^2}{\mu_k} + \delta \left(\sum_{k=1}^p \mu_k^q - 1 \right).$$

Newton's Method: L_q-Norm

(Kloft et al., 2009)

■ Computing the derivatives:

$$\frac{\partial L}{\partial \mu_k} = -\frac{1}{2} \frac{\mathbf{w}_k^\top \mathbf{w}_k}{\mu_k^2} + \delta \mu_k^{q-1} \quad \frac{\partial^2 L}{\partial \mu_k^2} = \frac{\mathbf{w}_k^\top \mathbf{w}_k}{\mu_k^3} + (q-1)\delta \mu_k^{q-2}$$

■ Hessian diagonal:

$$\Delta \mu_k = \frac{\frac{1}{2} \mu_k \mathbf{w}_k^\top \mathbf{w}_k - \delta \mu_k^{q+2}}{\mathbf{w}_k^\top \mathbf{w}_k + (q-1)\delta \mu_k^{q+1}}$$

- various techniques used to enforce non-negative parameters.

Mirror Descent

■ Optimization problem:

$$\min_{\mu \in \Delta} F(\mu)$$

■ Approximate F :

$$G_t(\mu) = F(\mu^t) + (\mu - \mu^t)^\top \nabla_\mu F(\mu)|_{\mu^t} + \frac{1}{s_t} B_\Omega(\mu^t \| \mu).$$

- strictly convex function $\Omega(\mu)$ and

$$B_\Omega(\mu^t \| \mu) = \Omega(\mu) - \Omega(\mu^t) - (\mu - \mu^t)^\top \nabla_\mu \Omega|_{\mu^t}$$

Bregman divergence defined by $\Omega(\mu)$.

Mirror Descent

■ Solving $\nabla_{\mu} G_t(\mu) = 0$

gives $\nabla_{\mu} \Omega|_{\mu} - \nabla_{\mu} \Omega|_{\mu^t} = -s_t \nabla_{\mu} F(\mu)|_{\mu^t}$

and the next value of μ given by

$$\mu^{t+1} = [\nabla_{\mu} \Omega]^{-1} \left(\nabla_{\mu} \Omega|_{\mu^t} - s_t \nabla_{\mu} F(\mu)|_{\mu^t} \right).$$

■ Examples

coordinate function inversion.

$$\Omega(\mu) = \frac{1}{2} \|\mu\|_2^2 \Rightarrow \mu^{t+1} = \mu^t - s_t \nabla_{\mu} F(\mu)|_{\mu^t}.$$

$$\Omega(\mu) = \mu^\top \log(\mu) \Rightarrow \mu^{t+1} = \mu^t \exp(-s_t \nabla_{\mu} F(\mu)|_{\mu^t}).$$

vector of coord. $\log(\mu_k)$.

Mirror Descent: Mixed-Norm MKL

(Nath et al., 2009)

■ Optimization problem:

$$\max_{\forall j, \boldsymbol{\mu}_j \in \Delta_{n_j}} \max_{\boldsymbol{\alpha} \in S_m(C), \boldsymbol{\gamma} \in \Delta_n} \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \left[\sum_{j=1}^n \frac{\sum_{k=1}^{n_j} \mu_{jk} \mathbf{K}_{jk}}{\gamma_j} \right] \boldsymbol{\alpha}.$$

● Kernel family:

$$\mathcal{K}_q = \left\{ K_{\boldsymbol{\mu}} = \sum_{k=1}^p \sum_{l=1}^{n_k} \frac{\mu_{kl}}{\gamma_k} K_{kl} : \boldsymbol{\mu}_k \in \Delta_{n_k}, \boldsymbol{\gamma} \in \Delta_p \right\}$$

$$\Omega(\boldsymbol{\mu}) = \sum_{k=1}^p \sum_{l=1}^{n_k} \left(\frac{\mu_{kl}}{p} + \frac{\delta}{pn_k} \right) \log \left(\frac{\mu_{kl}}{p} + \frac{\delta}{pn_k} \right).$$

Mirror Descent: Mixed-Norm MKL

(Nath et al., 2009)

- Update of kernel parameter

$$\boldsymbol{\mu}_{kl}^{t+1} = \frac{\mu_{kl}^t \exp(-ps_t[\nabla F|_{u^t}]_{kl})}{\sum_{l=1}^{n_k} \mu_{kl}^t \exp(-ps_t[\nabla F|_{u^t}]_{kl})}$$

$$F = 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \boldsymbol{\alpha}$$

$$\mathcal{K}_q = \left\{ K_{\boldsymbol{\mu}} = \sum_{k=1}^p \sum_{l=1}^{n_k} \frac{\mu_{kl}}{\gamma_k} K_{kl} : \boldsymbol{\mu}_k \in \Delta_{n_k}, \gamma \in \Delta_p \right\}$$

- Specific step-size gives bound on the number of iterations.

Beyond Wrappers

- Avoiding call to SVM:
 - Online methods, L_q -norm.
 - Projected gradient, KRR.

Online Methods - L_q-Norm

(Orabona & Jie, 2011)

■ Optimization problem:

$$\min_{\bar{w}} \Omega(\bar{w}) + \frac{1}{N} \sum_{i=1}^N \ell(\bar{w}, \phi(x_i, \cdot), y_i) ,$$

$$\Omega(\bar{w}) := \lambda/2 \|\bar{w}\|_{2, \frac{2 \log F}{2 \log F - 1}}^2 + \alpha \|\bar{w}\|_{2,1},$$

- where ℓ is the hinge loss for the case of SVMs.

■ Use Mirror Descent algorithm to update w:

$$\mathbf{w}^{t+1} = \nabla_{\mathbf{w}} \Omega^{-1}(\nabla_{\mathbf{w}} \Omega|_{\mathbf{w}^t} - s_t \nabla_{\mathbf{w}} l(\mathbf{w})|_{\mathbf{w}^t})$$

- where $\nabla_{\mathbf{w}} l(\mathbf{w})|_{\mathbf{w}^t}$ is determined by sampling.

Projected Gradient, KRR

■ Kernel family:

- non-negative combinations.
- L_q regularization.

■ Optimization problem:

$$\min_{\mu} \max_{\alpha} -\lambda \alpha^\top \alpha - \sum_{k=1}^p \mu_k \alpha^\top \mathbf{K}_k \alpha + 2\alpha^\top \mathbf{y}$$

subject to: $\mu \geq 0 \wedge \|\mu - \mu_0\|_q \leq \Lambda$.

- convex optimization: linearity in μ and convexity of pointwise maximum.

Projected Gradient, KRR

- Solving maximization problem in α , closed-form solution $\alpha = (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y}$, reduces problem to

$$\min_{\mu} \mathbf{y}^\top (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y}$$

subject to: $\mu \geq 0 \wedge \|\mu - \mu_0\|_2 \leq \Lambda$.

- Convex optimization problem, one solution using projection-based gradient descent:

$$\begin{aligned}\frac{\partial F}{\partial \mu_k} &= \text{Tr} \left[\frac{\partial \mathbf{y}^\top (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y}}{\partial (\mathbf{K}_\mu + \lambda \mathbf{I})} \frac{\partial (\mathbf{K}_\mu + \lambda \mathbf{I})}{\partial \mu_k} \right] \\ &= - \text{Tr} \left[(\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y} \mathbf{y}^\top (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \frac{\partial (\mathbf{K}_\mu + \lambda \mathbf{I})}{\partial \mu_k} \right] \\ &= - \text{Tr} \left[(\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y} \mathbf{y}^\top (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{K}_k \right] \\ &= - \mathbf{y}^\top (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{K}_k (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y} = -\boldsymbol{\alpha}^\top \mathbf{K}_k \boldsymbol{\alpha}. \quad \square\end{aligned}$$

Projected Gradient, KRR - L₂ Reg.

(Cortes et al., 2009)

PROJECTIONBASEDGRADIENTDESCENT($((\mathbf{K}_k)_{k \in [1, p]}, \mu_0)$)

```
1    $\mu \leftarrow \mu_0$ 
2    $\mu' \leftarrow \infty$ 
3   while  $\|\mu' - \mu\| > \epsilon$  do
4        $\mu \leftarrow \mu'$ 
5        $\alpha \leftarrow (\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y}$ 
6        $\mu' \leftarrow \mu + \eta (\alpha^\top \mathbf{K}_1 \alpha, \dots, \alpha^\top \mathbf{K}_p \alpha)^\top$ 
7       for  $k \leftarrow 1$  to  $p$  do
8            $\mu'_k \leftarrow \max(0, \mu'_k)$ 
9            $\mu' \leftarrow \mu_0 + \Lambda \frac{\mu' - \mu_0}{\|\mu' - \mu_0\|}$ 
10  return  $\mu'$ 
```

Interpolated Step, KRR - L₂ Reg.

(Cortes et al., 2009)

INTERPOLATEDITERATIVEALGORITHM($((\mathbf{K}_k)_{k \in [1, p]}, \mu_0)$)

```
1  $\alpha \leftarrow \infty$ 
2  $\alpha' \leftarrow (\mathbf{K}_{\mu_0} + \lambda \mathbf{I})^{-1} \mathbf{y}$ 
3 while  $\|\alpha' - \alpha\| > \epsilon$  do
4      $\alpha \leftarrow \alpha'$ 
5      $\mathbf{v} \leftarrow (\alpha^\top \mathbf{K}_1 \alpha, \dots, \alpha^\top \mathbf{K}_p \alpha)^\top$ 
6      $\mu \leftarrow \mu_0 + \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|}$ 
7      $\alpha' \leftarrow \eta \alpha + (1 - \eta)(\mathbf{K}_\mu + \lambda \mathbf{I})^{-1} \mathbf{y}$ 
8 return  $\alpha'$ 
```

Simple and very efficient: few iterations (less than 15).

SMO Solutions

- MKL and SMO - (Bach et al., 2004)
 - Moreau-Yosida regularization to form smooth problem for L_1 -regularization.
- MKL and SMO - (Vishwanathan et al., 2010)
 - Squared L_q -norm results in smooth problem in dual.

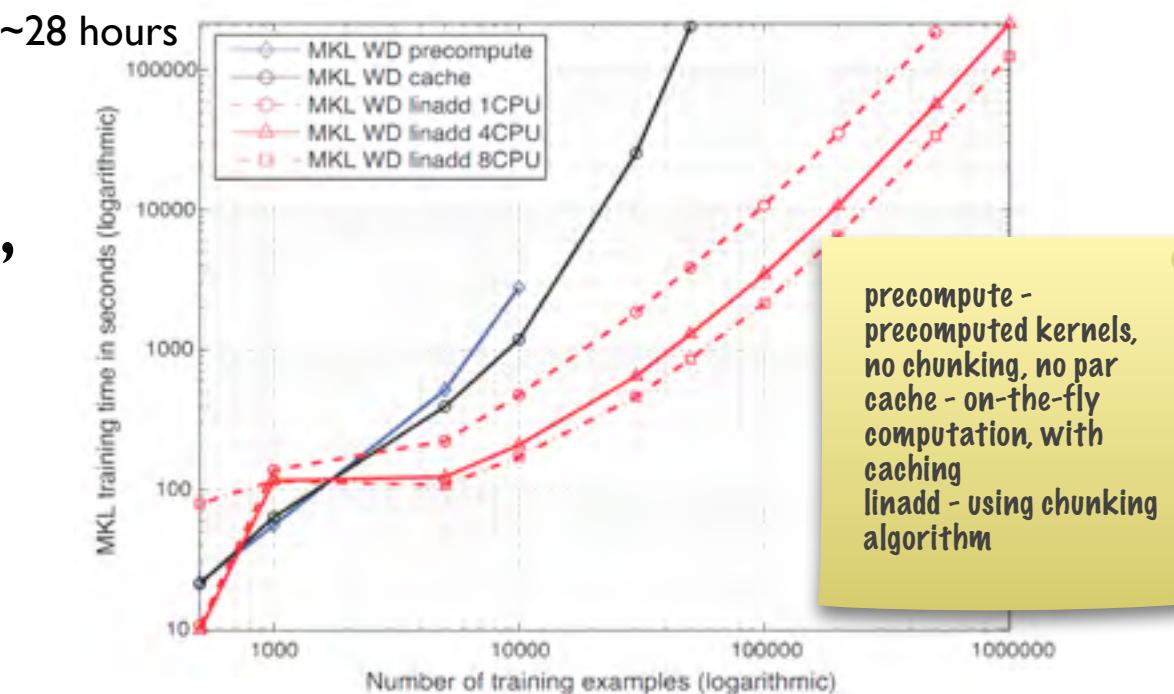
Experimental Results

- Solving the same problem.
 - only difference is the norm of the regularization.
- Compare speed for different norms.
- Compare accuracy for different norms.

SILP Algorithm

(Sonnenburg et al., 2006)

- Semi-infinite linear programming (SILP) approach for convex combinations.
- 20 base kernels, 1,000,000 training points (human splice dataset).
- Requires on-the-fly kernel computation, employs caching, chunking and parallelization.



SimpleMKL

(Rakotomamonjy et al., 2006)

- Reduced gradient method for solving L_1 -regularized MKL.
- In regimes of small scale data, but 100's of kernels, SimpleMKL show improvement over SILP method.

Pima $\ell = 538$ $M = 117$

Algorithm	# Kernel	Accuracy	Time (s)	# SVM eval	# Gradient eval
SILP	11.6 ± 1.0	76.5 ± 2.3	224 ± 37	95.6 ± 13	95.6 ± 13
SimpleMKL	14.7 ± 1.4	76.5 ± 2.6	79.0 ± 13	314 ± 44	24.3 ± 4.8
Grad. Desc.	14.8 ± 1.4	75.5 ± 2.5	219 ± 24	873 ± 147	118 ± 8.7

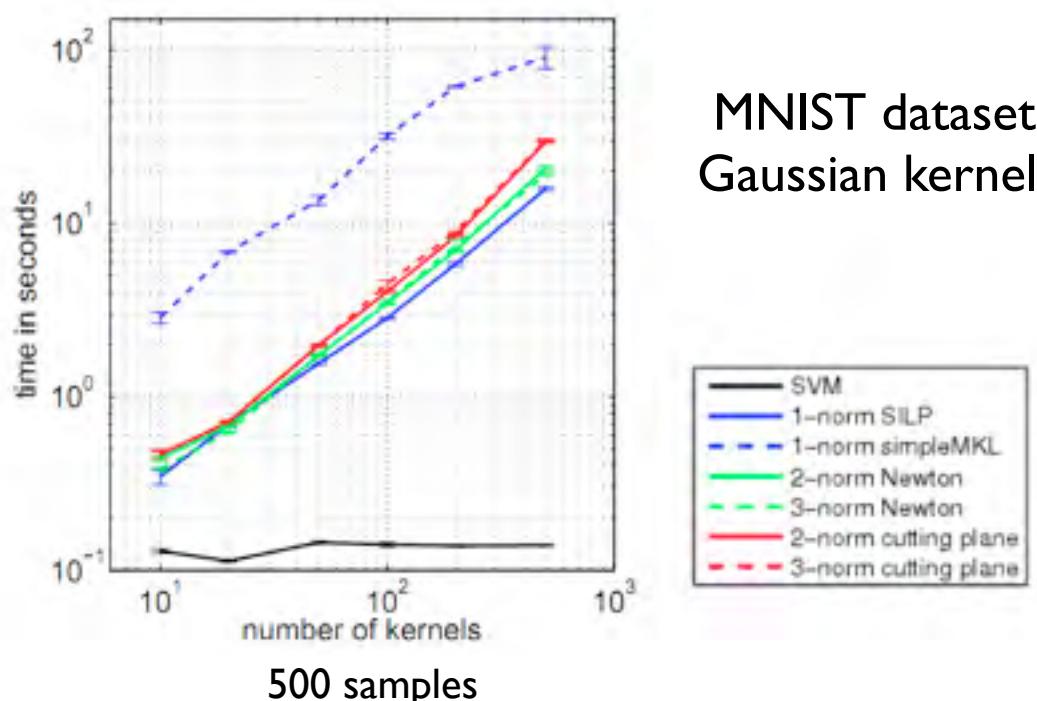
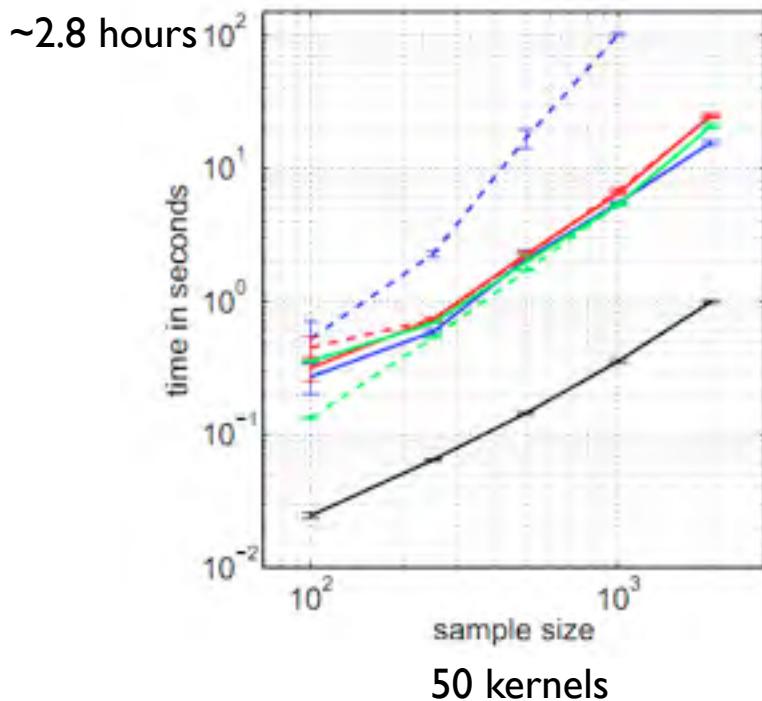
Sonar $\ell = 146$ $M = 793$

Algorithm	# Kernel	Accuracy	Time (s)	# SVM eval	# Gradient eval
SILP	33.5 ± 3.8	80.5 ± 5.1	2290 ± 864	903 ± 187	903 ± 187
SimpleMKL	36.7 ± 5.1	80.6 ± 5.1	163 ± 93	2770 ± 1560	115 ± 66
Grad. Desc.	35.7 ± 3.9	80.2 ± 4.7	469 ± 90	7630 ± 2600	836 ± 99

Efficient L_p Regularized

(Kloft et al., 2009)

- Wrapper methods for L_p -regularized combinations: Newton or Cutting Plane + SVM.
- Allows for efficient computation of non-sparse combinations of kernel.

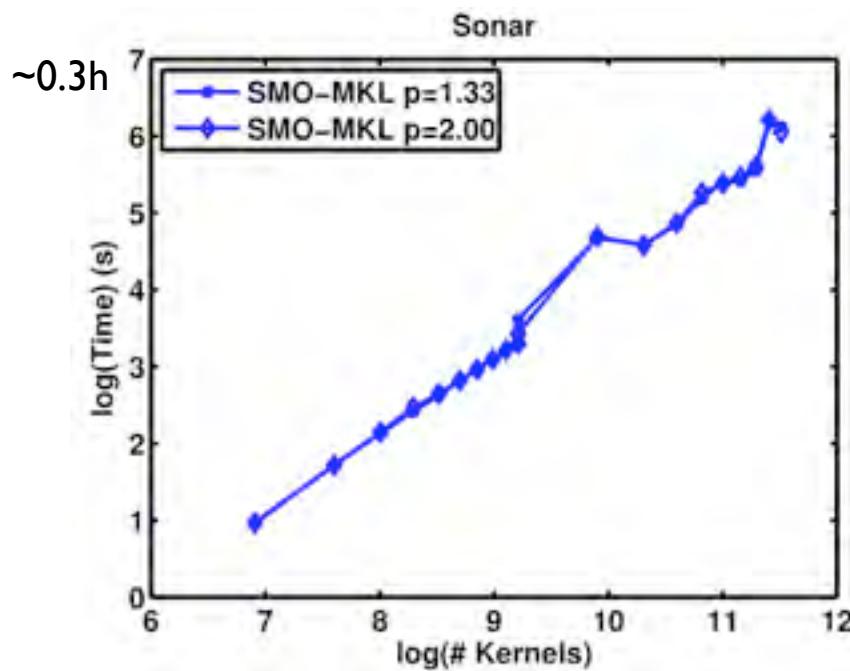


MNIST dataset,
Gaussian kernels

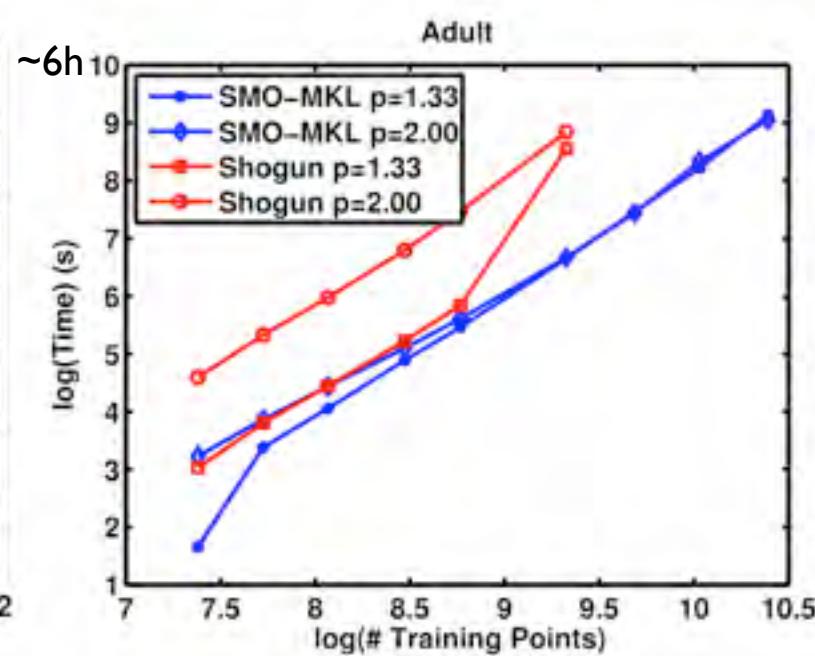
SMO Optimization

(Vishwanathan et al., 2009)

- SMO for L_p -regularization.
- Found to scale better with training size than (Kloft et al., 2009).



166 points

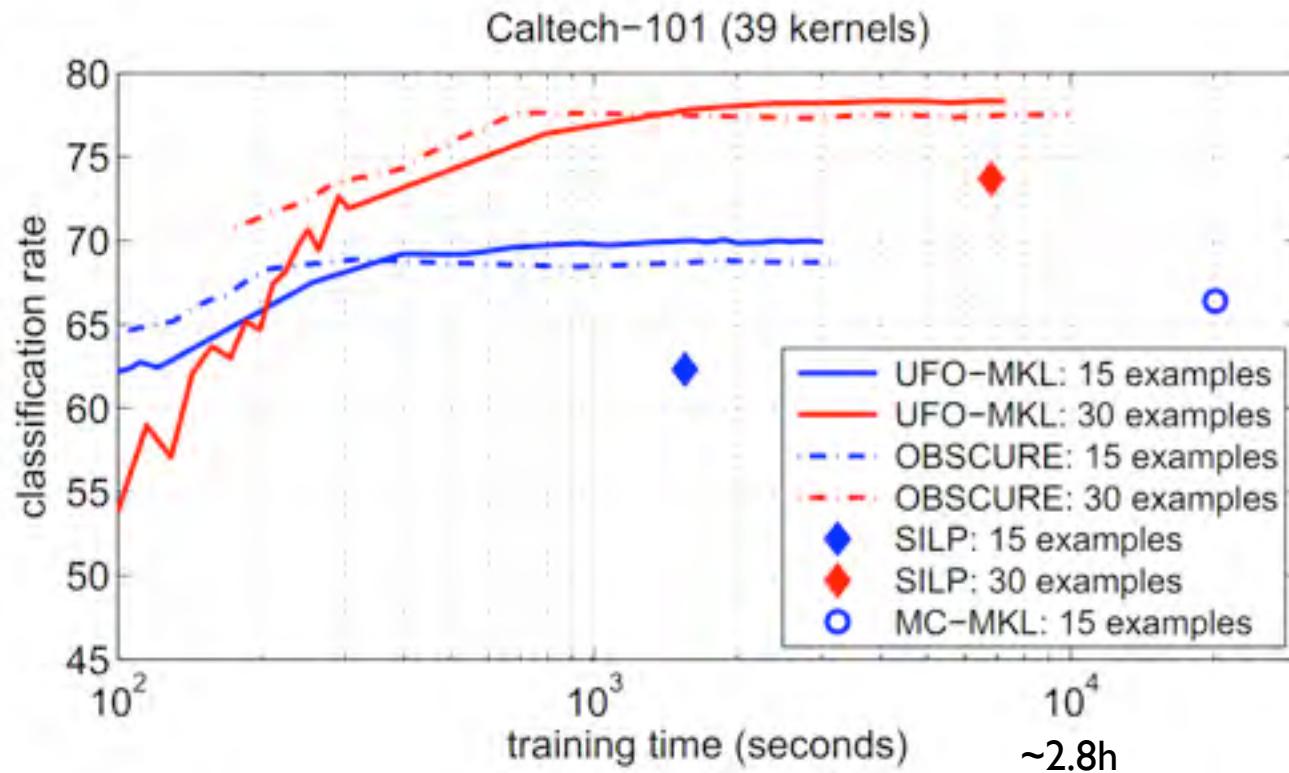


50 kernels

Stochastic Gradient Descent

(Orabona et al., 2010 & 2011)

- OBSCURE and UFO-MKL for L_p -regularization.
- Primal formulation allows for general loss functions, e.g. multi-class classification.



L_1 -Regularized Combinations

(Lanckriet et al., 2004)

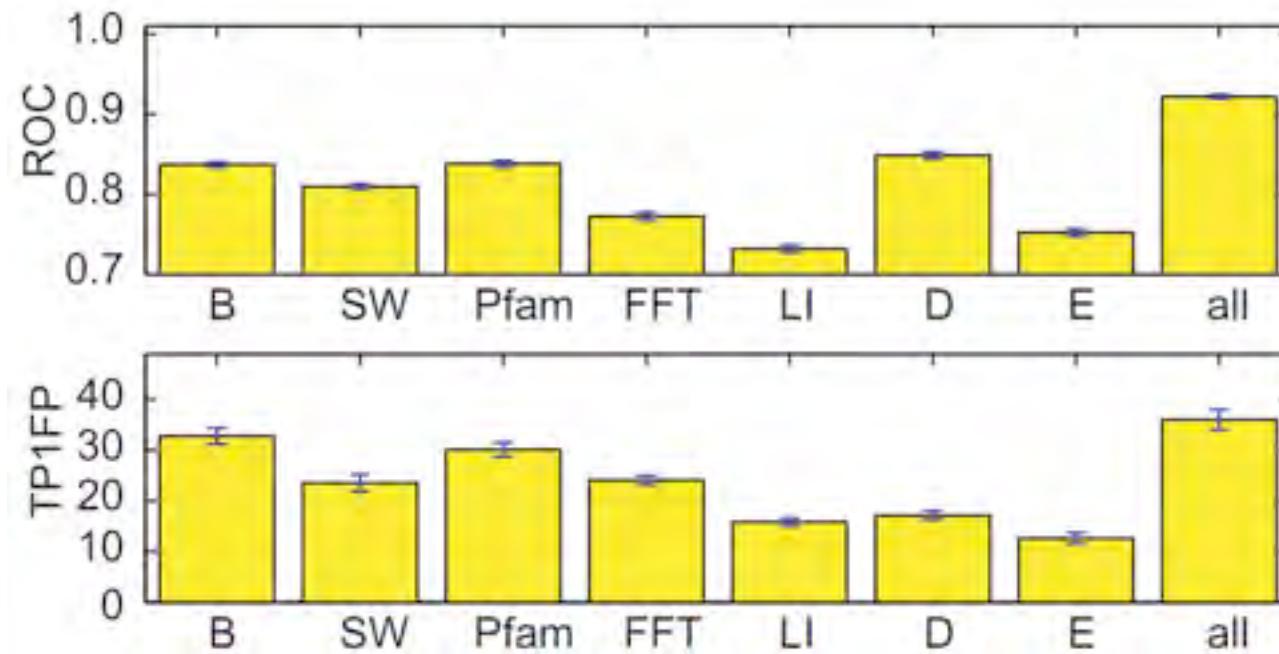
- Learn with sparse linear combinations of kernels.
- Combining kernels can help performance, but do simple uniform combinations suffice?

	$\mu_{1,+}$	$\mu_{2,+}$	$\mu_{3,+}$	$\mu_{4,+}$	$\mu_{5,+}$	TSA	SM2,C	TSA best c/v	RBF
Breast Cancer	0	0	3.24	0.94	0.82	97.1 %		96.8 %	
Ionosphere	0.85	0.85	2.63	0.68	0	94.5 %		94.2 %	
Heart	0	3.89	0.06	1.05	0	84.1 %		83.2 %	
Sonar	0	3.93	1.07	0	0	84.8 %		84.2 %	
2-norm	0.49	0.49	0	3.51	0	96.5 %		97.2 %	

L_1 -Regularized Combinations

(Lanckriet et al., Bioinformatics 2004)

- Yeast protein classification, 7 domain specific kernels, 2318 samples.

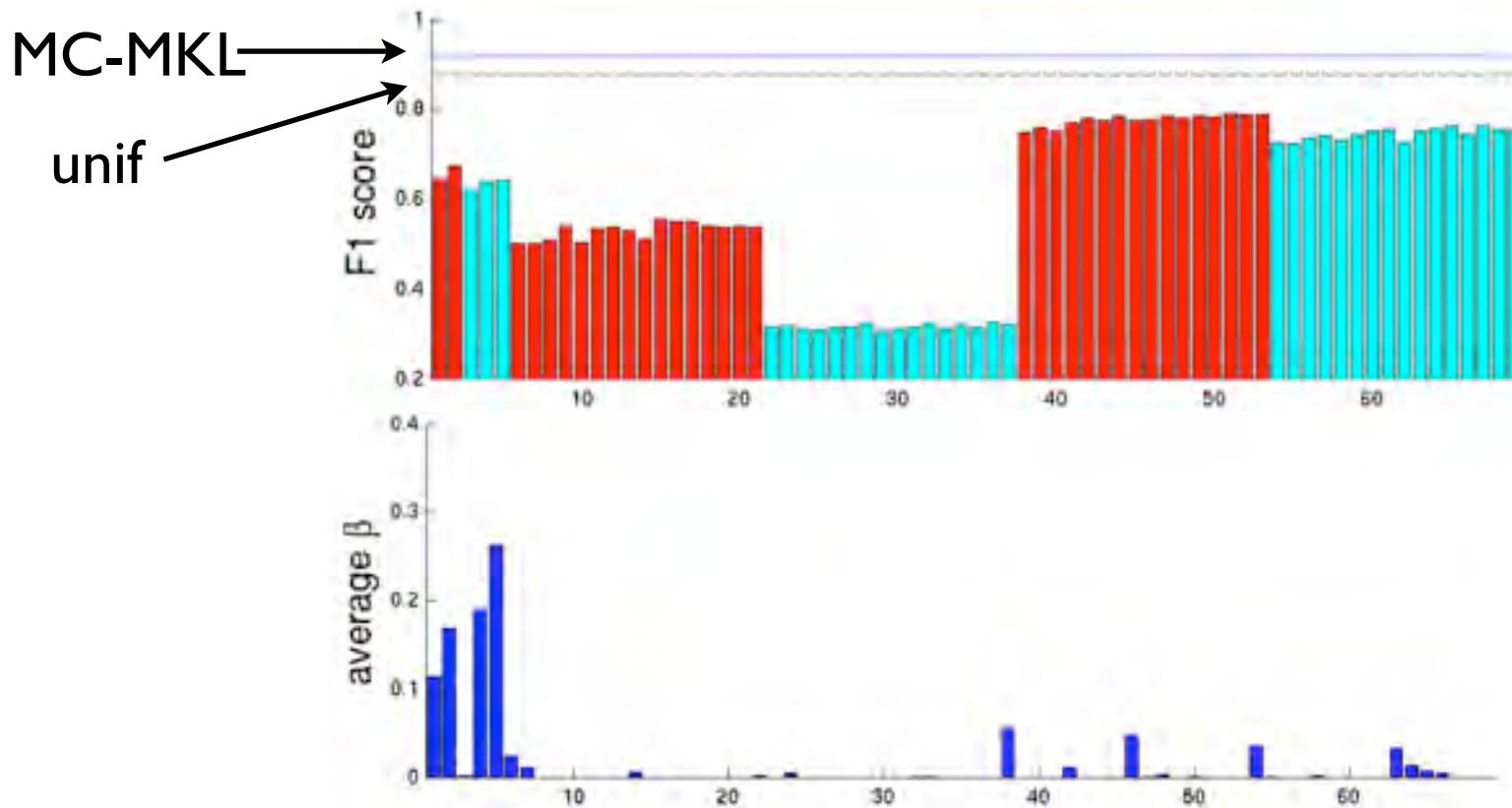


K_B	K_{SW}	K_D	K_E	K_{R1}	K_{R2}	K_{R3}	K_{R4}	TP1FP (%)	ROC
1.81	1.05	0.73	0.42	—	—	—	—	35.71 ± 2.13	0.9196 ± 0.0023
3.30	1.98	1.31	0.79	0.08	0.17	0.21	0.17	34.14 ± 2.09	0.9145 ± 0.0026
1.00	1.00	1.00	1.00	—	—	—	—	33.87 ± 2.20	0.9180 ± 0.0026
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	26.24 ± 1.39	0.8627 ± 0.0033

Multi-Class L₁-Regularized

(Zien & Ong., 2007; Ong & Zien, 2008)

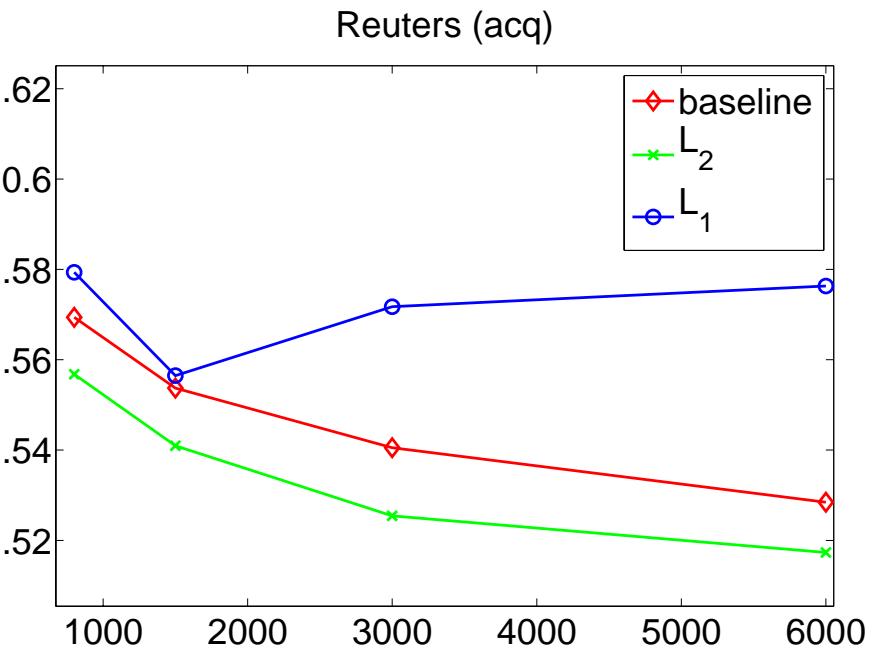
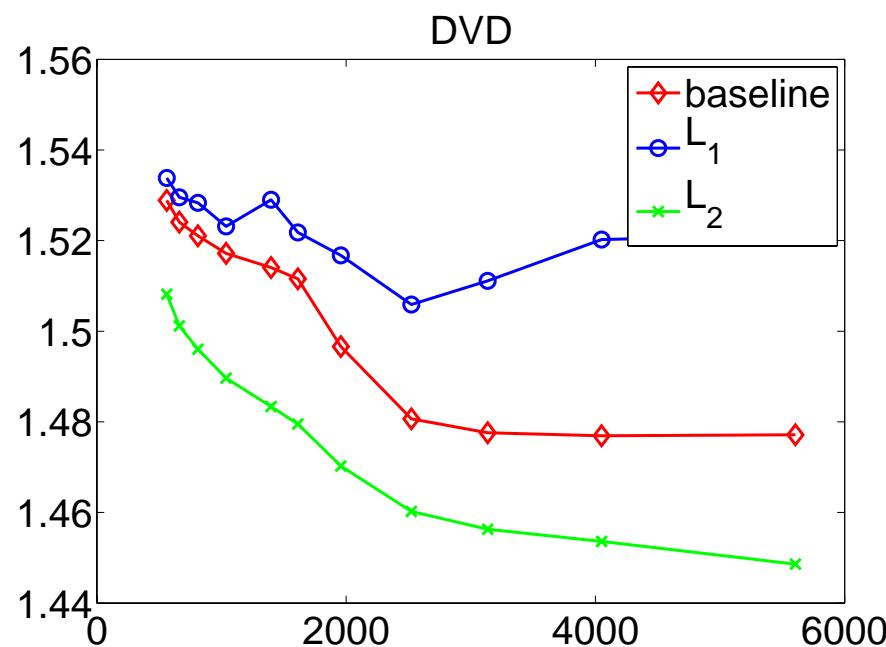
- Predict subcellular localization (TargetP dataset), 5 classes, 69 base kernels.
- Multi-class SVM with L₁-regularization.



L_2 -Regularized Combinations

(Cortes et al., 2009)

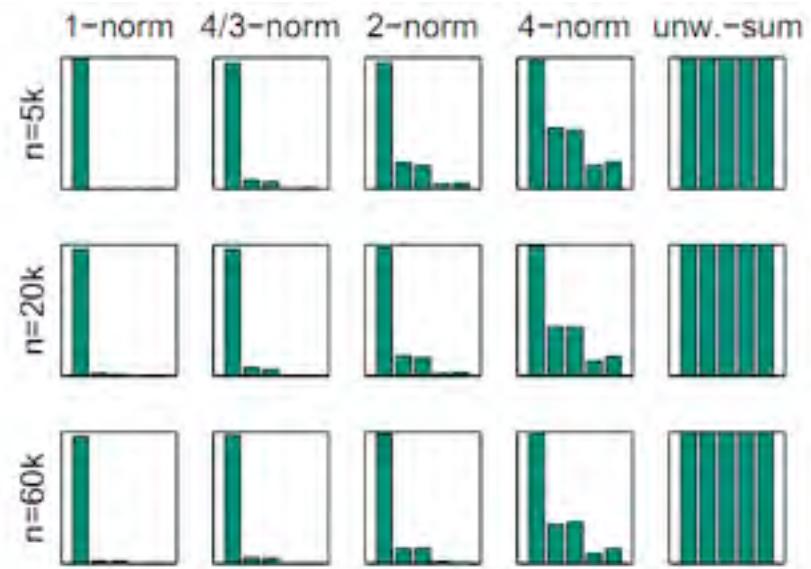
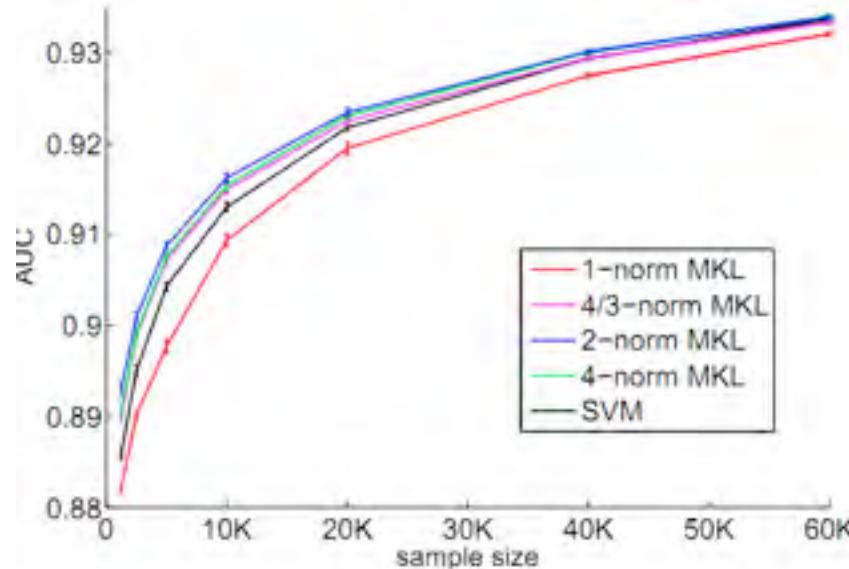
- Dense combinations are beneficial when using many kernels.
- Combining kernels based on single features, can be viewed as principled feature weighting.



L_p -Regularized Combinations

(Sonnenburg et al., Bioinformatics 2006; Kloft et al., 2009)

- Non-sparse combination are found to be more effective (in terms of AUC) for transcription start site (TSS) recognition.
- 5 kernels, up to 60,000 training examples.



This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

Non-Linear Combinations and Alternative Formulations

- Gaussian and polynomial kernels
 - DC-Programming algorithm (Argyriou et al., 2005)
 - Generalized MKL (Varma & Babu, 2009)
 - Polynomial kernels - KRR (Cortes et al., 2009)
- Hierarchical kernels (Bach, 2008)
- Hyperkernels (Ong et al., 2005)
- Radius-based kernel learning (Gai et al., 2010)

Gaussian and Polynomial Kernels

(Weston et al, 2000; Argyriou et al., 2005; Varma and Babu, 2009)

- Optimize over a continuously parameterized set of

Gaussians: $\mathbf{K}_\mu(x_i, x_j) = \prod_{k=1}^p \exp(-\mu_k(x_{ik} - x_{jk})^2)$

Polynomials: $\mathbf{K}_{\mu,d}(x_i, x_j) = (1 + \sum_{k=1}^p \mu_k x_{ik} x_{jk})^d$

- Wrapper method:

- (Argyriou et al., 2005): squared loss, DC (difference of convex functions) to find new parameters.
- (Chapelle et al., 2000; Varma & Babu, 2009): hinge loss, steepest descent + projection onto feasible set.

GMKL: Reality Check

(Varma and Babu, 2009)

- **Feature selection:** train MKL and rank features according to weights. Retrain with top-k weights. Compare to other feature selection algorithms:

Ionosphere: $N = 246$, $M = 34$, Uniform MKL = 89.9 ± 2.5 , Uniform GMKL = 94.6 ± 2.0								
N_d	AdaBoost	OWL-QN	LP-SVM	S-SVM	BAHSIC	MKL	GMKL	
5	75.2 ± 6.9	84.0 ± 6.0	86.7 ± 3.1	87.0 ± 3.1	87.1 ± 3.6	85.1 ± 3.2	90.9 ± 1.9	
10	—	87.6 ± 2.2	90.6 ± 3.4	90.2 ± 3.5	90.2 ± 2.6	87.8 ± 2.4	93.7 ± 2.1	
15	—	89.1 ± 1.9	93.0 ± 2.1	91.9 ± 2.0	92.6 ± 3.0	87.7 ± 2.2	94.1 ± 2.1	
20	—	89.2 ± 1.8	92.8 ± 3.0	92.4 ± 2.5	93.4 ± 2.6	87.8 ± 2.8	—	
25	—	89.1 ± 1.9	92.6 ± 2.7	92.4 ± 2.7	94.0 ± 2.2	87.9 ± 2.7	—	
30	—	—	92.6 ± 2.6	92.9 ± 2.5	94.3 ± 1.9	—	—	
34	—	—	92.6 ± 2.6	92.9 ± 2.5	94.6 ± 2.0	—	—	
		75.1 (9.8)	89.2 (25.2)	92.6 (34.0)	92.9 (34.0)	—	88.1 (29.3)	94.4 (16.9)

$$\text{MKL + } l_1\text{-reg: } K_{\mu}(x_i, x_j) = \sum_{k=1}^p \mu_k \exp(-\gamma_k (x_{ik} - x_{jk})^2)$$

$$\text{GMKL + } l_1\text{-reg: } K_{\mu}(x_i, x_j) = \prod_{k=1}^p \exp(-\mu_k (x_{ik} - x_{jk})^2)$$

Unknown how γ_k is chosen...

GMKL: Reality Check

(Varma and Babu, 2009)

■ Accuracy:

Database	SimpleMKL	GMKL
Sonar	80.6 ± 5.1 (793)	82.3 ± 4.8 (60)
Wpbc	76.7 ± 1.2 (442)	79.0 ± 3.5 (34)
Ionosphere	91.5 ± 2.5 (442)	93.0 ± 2.1 (34)
Liver	65.9 ± 2.3 (091)	72.7 ± 4.0 (06)
Pima	76.5 ± 2.6 (117)	77.2 ± 2.1 (08)

Database	N	M	HKL	GMKL
Magic04	1024	10	84.4 ± 0.8	86.2 ± 1.2
Spambase	1024	57	91.9 ± 0.7	93.2 ± 0.8
Mushroom	1024	22	99.9 ± 0.2	100 ± 0.0

$$\text{HKL} + \ell_1\text{-reg: } \mathbf{K}_{\mu,4}(x_i, x_j) = \prod_{k=1}^p (1 + \mu_k x_{ik} x_{jk})^4$$

$$\text{GMKL} + \ell_1\text{-reg: } \mathbf{K}_{\mu,2}(x_i, x_j) = (1 + \sum_{k=1}^p \mu_k x_{ik} x_{jk})^2$$

Polynomial Kernels - KRR

(Cortes et al., 2010)

- $p \geq 1$ base PDS kernel functions K_1, \dots, K_p .
- Kernel family: polynomial degree $d \geq 2$.

$$\mathcal{K}_q = \left\{ K_{\boldsymbol{\mu}} = \left(\sum_{k=1}^p \mu_k K_k \right)^d : \boldsymbol{\mu} \in \Delta_q \right\}$$

with $\Delta_q = \left\{ \boldsymbol{\mu} \in \mathbb{R}^p : \|\boldsymbol{\mu}\|_q \leq 1, \boldsymbol{\mu} \geq \mathbf{0} \right\}$.

- Hypothesis sets:

$$H_q = \left\{ h \in \mathbb{H}_K : K \in \mathcal{K}_q, \|h\|_{\mathbb{H}_K} \leq 1 \right\}.$$

Polynomial Kernels - KRR

- Optimization problem: case $d=2$.

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \sum_{k,l=1}^p \mu_k \mu_l \boldsymbol{\alpha}^\top (\mathbf{K}_k \circ \mathbf{K}_l) \boldsymbol{\alpha} + 2 \boldsymbol{\alpha}^\top \mathbf{y}$$

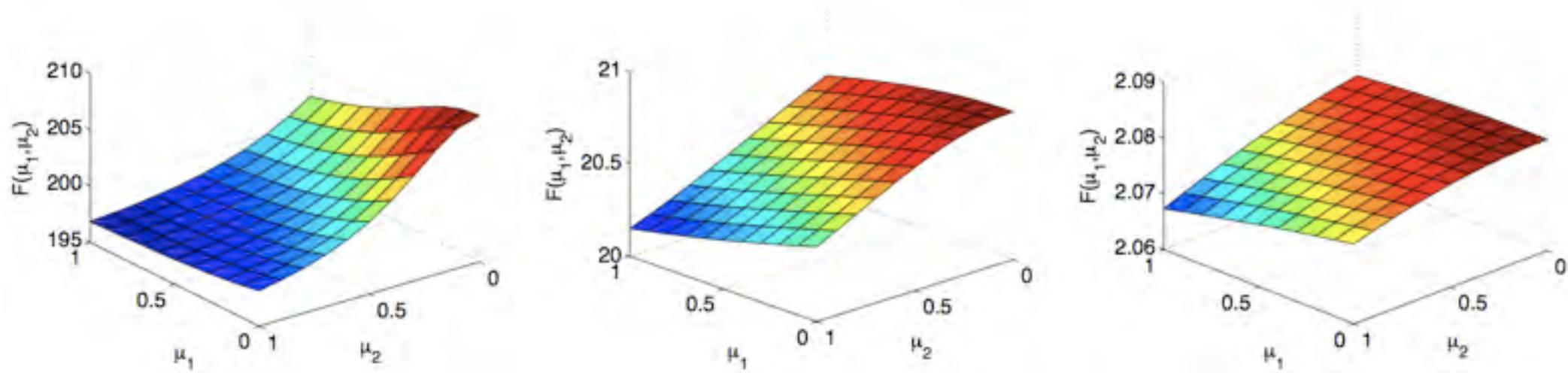
subject to: $\boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_q \leq \Lambda$.

- Closed-form solution $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})^{-1} \mathbf{y}$ leads to:

$$\min_{\boldsymbol{\mu}} F(\boldsymbol{\mu}) = \mathbf{y}^\top \left(\sum_{k,l=1}^p \mu_k \mu_l \mathbf{K}_k \circ \mathbf{K}_l + \lambda \mathbf{I} \right)^{-1} \mathbf{y}$$

subject to: $\boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_q \leq \Lambda$.

Function Properties



- Two properties:
 - decreasing.
 - no interior stationary points
→ optimal solution at the boundary.

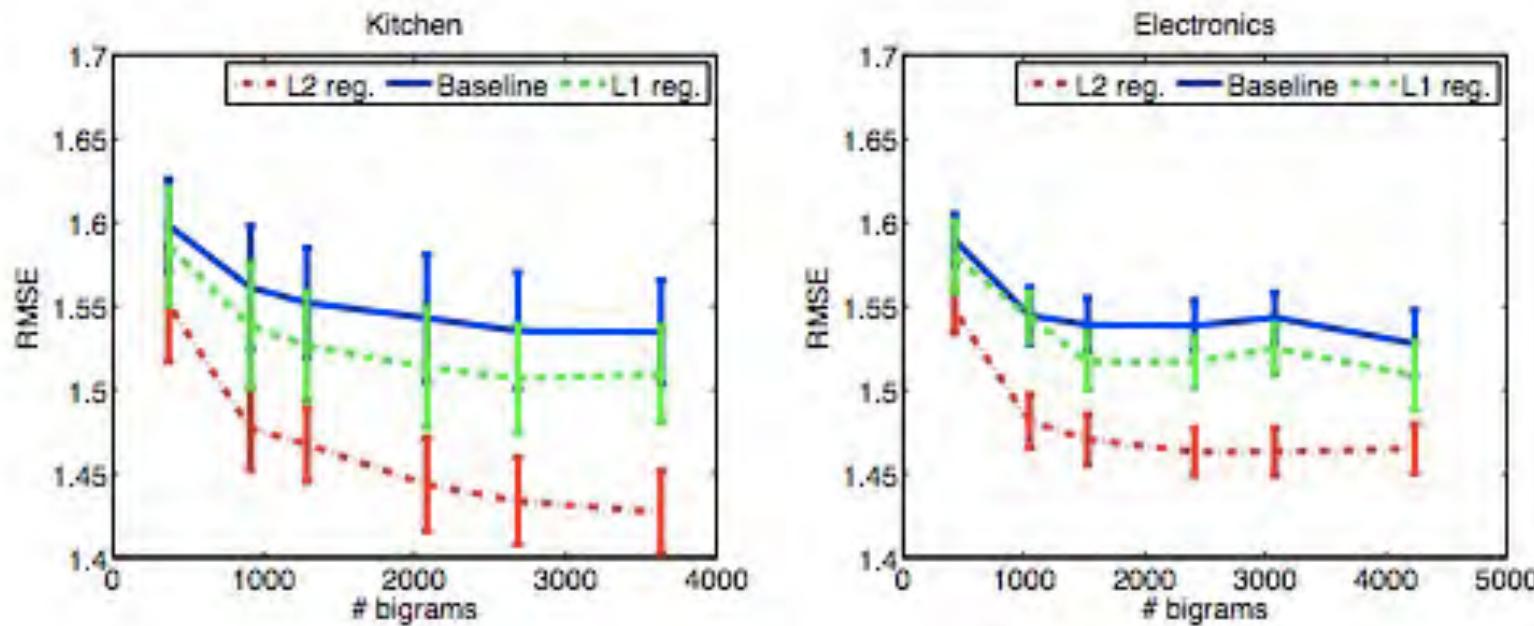
$$\begin{aligned}\forall \mu, \nabla F(\mu) &\leq 0 \\ \forall \mu > 0, \nabla F(\mu) &\neq 0\end{aligned}$$

- Convex regions exist under certain conditions.

Pol. Kernels - KRR: Reality Check

(Cortes et al., 2010)

- Sentiment dataset (Blitzer et al.).



- Polynomial kernels with $d=2$, and L_1 and L_2 regularization. Baseline is a uniformly weighted quadratic kernel.

Hierarchical Kernel Learning

(Bach, 2008)

■ Example:

- Sub kernel:

$$K_{i,j}(x_i, x'_i) = \binom{q}{j} (1 + x_i x'_i)^j, \quad i \in [1, p], \quad j \in [0, q]$$

- Full kernel:

$$K(x, x') = \prod_{i=1}^p (1 + x_i x'_i)^q$$

- Convex optimization problem under some assumptions, complexity polynomial in the number of kernels selected, sparsity through L_1 regularization and hierarchical selection criteria.

HKL: Reality Check

(Bach, 2008)

Regression:
Normalized
Mean Squared
Error x 100

dataset	n	p	k	$\#(V)$	L2	MKL	HKL
abalone	4177	10	pol4	$\approx 10^7$	44.2 ± 1.3	44.5 ± 1.1	43.3 ± 1.0
abalone	4177	10	rbf	$\approx 10^{10}$	43.0 ± 0.9	43.7 ± 1.0	43.0 ± 1.1
bank-32fh	8192	32	pol4	$\approx 10^{22}$	40.1 ± 0.7	38.7 ± 0.7	38.9 ± 0.7
bank-32fh	8192	32	rbf	$\approx 10^{31}$	39.0 ± 0.7	38.4 ± 0.7	38.4 ± 0.7
bank-32fm	8192	32	pol4	$\approx 10^{22}$	6.0 ± 0.1	6.1 ± 0.3	5.1 ± 0.1
bank-32fm	8192	32	rbf	$\approx 10^{31}$	5.7 ± 0.2	5.9 ± 0.2	4.6 ± 0.2
bank-32nh	8192	32	pol4	$\approx 10^{22}$	44.3 ± 1.2	46.0 ± 1.2	43.6 ± 1.1
bank-32nh	8192	32	rbf	$\approx 10^{31}$	44.3 ± 1.2	46.1 ± 1.1	43.5 ± 1.0
bank-32nm	8192	32	pol4	$\approx 10^{22}$	17.2 ± 0.6	21.0 ± 0.7	16.8 ± 0.6
bank-32nm	8192	32	rbf	$\approx 10^{31}$	16.9 ± 0.6	20.9 ± 0.7	16.4 ± 0.6
boston	506	13	pol4	$\approx 10^9$	17.1 ± 3.6	22.2 ± 2.2	18.1 ± 3.8
boston	506	13	rbf	$\approx 10^{12}$	16.4 ± 4.0	20.7 ± 2.1	17.1 ± 4.7
pumadyn-32fh	8192	32	pol4	$\approx 10^{22}$	57.3 ± 0.7	56.4 ± 0.7	56.4 ± 0.8
pumadyn-32fh	8192	32	rbf	$\approx 10^{31}$	57.7 ± 0.6	56.5 ± 0.8	55.7 ± 0.7
pumadyn-32fm	8192	32	pol4	$\approx 10^{22}$	6.9 ± 0.1	7.0 ± 0.1	3.1 ± 0.0
pumadyn-32fm	8192	32	rbf	$\approx 10^{31}$	5.0 ± 0.1	7.1 ± 0.1	3.4 ± 0.0
pumadyn-32nh	8192	32	pol4	$\approx 10^{22}$	84.2 ± 1.3	83.6 ± 1.3	36.7 ± 0.4
pumadyn-32nh	8192	32	rbf	$\approx 10^{31}$	56.5 ± 1.1	83.7 ± 1.3	35.5 ± 0.5
pumadyn-32nm	8192	32	pol4	$\approx 10^{22}$	60.1 ± 1.9	77.5 ± 0.9	5.5 ± 0.1
pumadyn-32nm	8192	32	rbf	$\approx 10^{31}$	15.7 ± 0.4	77.6 ± 0.9	7.2 ± 0.1

Hyperkernels

(Ong et al, 2005)

- Kernels over kernels, \underline{K}
- Representer theorem: m^2 Lagrange multipliers:

$$K(x, x') = \sum_{i,j=1}^m \beta_{i,j} \underline{K}((x_i, x_j), (x, x')) \quad \forall x, x' \in X, \quad \beta_{i,j} \geq 0$$

- Hyperkernel example:

$$\underline{K}\left((x, x'), (x'', x''')\right) = \prod_{j=1}^d \frac{1 - \lambda}{1 - \lambda \exp\left(-\sigma_j((x_j - x'_j)^2 + (x''_j - x'''_j)^2)\right)}$$

- For fixed σ_j SDP problem similar to Lanckriet, SeDuMi.

Hyperkernels: Reality Check

(Ong et al, 2005)

Data	C -SVM	ν -SVM	Lag-SVM	Best other	CV Tuned SVM (C)
syndata	2.8 ± 2.4	1.9 ± 1.9	2.4 ± 2.2	NA	$5.9 \pm 5.4 (10^8)$
pima	23.5 ± 2.0	27.7 ± 2.1	23.6 ± 1.9	23.5	$24.1 \pm 2.1 (10^4)$
ionosph	6.6 ± 1.8	6.7 ± 1.8	6.4 ± 1.9	5.8	$6.1 \pm 1.8 (10^3)$
wdbc	3.3 ± 1.2	3.8 ± 1.2	3.0 ± 1.1	3.2	$5.2 \pm 1.4 (10^6)$
heart	19.7 ± 3.3	19.3 ± 2.4	20.1 ± 2.8	16.0	$23.2 \pm 3.7 (10^4)$
thyroid	7.2 ± 3.2	10.1 ± 4.0	6.2 ± 3.1	4.4	$5.2 \pm 2.2 (10^5)$
sonar	14.8 ± 3.7	15.3 ± 3.7	14.7 ± 3.6	15.4	$15.3 \pm 4.1 (10^3)$
credit	14.6 ± 1.8	13.7 ± 1.5	14.7 ± 1.8	22.8	$15.3 \pm 2.0 (10^8)$
glass	6.0 ± 2.4	8.9 ± 2.6	6.0 ± 2.2	NA	$7.2 \pm 2.7 (10^3)$

$$\underline{K}\left((x, x'), (x'', x''')\right) = \prod_{j=1}^d \frac{1 - \lambda}{1 - \lambda \exp\left(-\sigma_j((x_j - x'_j)^2 + (x''_j - x'''_j)^2)\right)}$$

σ_j is fixed.

Radius-based Kernel Learning, RKL

(Gai et al, 2010)

- Slack term $O(\sqrt{R^2/\rho^2})$.
- For fixed kernel, the radius is constant, but for combinations of kernels it varies.
- Primal: $R^2(K) = \min_{y,c} y, \quad \text{s.t.} \quad y \geq \|\phi_K(x_i) - c\|^2$
- Dual: $R^2(K) = \max_{\beta_i} \sum_{i=1}^m \beta_i K(x_i, x_i) - \sum_{i,j=1}^m \beta_i \beta_j K(x_i, x_j), \quad \text{s.t.} \quad \beta_i \geq 0, \sum_{i=1}^m \beta_i = 1$
- RKL optimization

$$\min_{\theta} g(\theta),$$

where $g(\theta) = \left\{ \max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2r^2(\theta)} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{i,j}(\theta), \quad \text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \right\}$,

where $r^2(\theta) = \left\{ \max_{\beta_i} \sum_i \beta_i K_{i,i}(\theta) - \sum_{i,j} \beta_i K_{i,j}(\theta) \beta_j, \quad \text{s.t.} \quad \sum_i \beta_i = 1, \quad \beta_i \geq 0 \right\}$.

RKL: Reality Check

(Gai et al, 2010)

Index	1 Unif	2 MKL L_1	3 KL-C L_1	4 Ours L_1	5 MKL L_2	6 KL-C L_2	7 Ours L_2	8 Ours No					
Constraint													
Data set	Acc.	Nk	Acc.	Nk	Acc.	Nk	Acc.	Nk	Acc.	Nk	Acc.	Nk	
Ionosphere	94.0(1.4)	20	92.9(1.6)	3.8	86.0(1.9)	4.0	95.7 (0.9)	2.8	94.3(1.5)	20	84.4(1.6)	18	
Splice	51.7(0.1)	20	79.5(1.9)	1.0	80.5(1.9)	2.8	86.5 (2.4)	3.2	82.0(2.2)	20	74.0(2.6)	14	
Liver	58.0(0.0)	20	59.1(1.4)	4.2	62.9(3.5)	4.0	64.1 (4.2)	3.6	67.0(3.8)	20	64.1(3.9)	11	
Fourclass	81.2(1.9)	20	97.7(1.2)	7.0	94.0(1.2)	2.0	100	(0.0)	1.0	97.3(1.6)	20	94.0(1.3)	17
Heart	83.7(6.1)	20	84.1(5.7)	7.4	83.3(5.9)	1.8	84.1(5.7)	5.2	83.7(5.8)	20	83.3(5.1)	19	
Germannum	70.0(0.0)	20	70.0(0.0)	7.2	71.9(1.8)	9.8	73.7 (1.6)	4.8	71.5(0.8)	20	71.6(2.1)	13	
Musk1	61.4(2.9)	20	85.5(2.9)	1.6	73.9(2.9)	2.0	93.3 (2.3)	4.0	87.4(3.0)	20	61.9(3.1)	19	
Wdbc	94.4(1.8)	20	97.0(1.8)	1.2	97.4(2.3)	4.6	97.4(1.6)	6.2	96.8(1.6)	20	97.4(2.0)	11	
Wpbc	76.5(2.9)	20	76.5(2.9)	7.2	52.2(5.9)	9.6	76.5(2.9)	17	75.9(1.8)	20	51.0(6.6)	17	
Sonar	76.5(1.8)	20	82.3(5.6)	2.6	80.8(5.8)	7.4	86.0 (2.6)	2.6	85.2(2.9)	20	80.2(5.9)	11	
Coloncancer	67.2(11)	20	82.6(8.5)	13	74.5(4.4)	11	84.2 (4.2)	7.2	76.5(9.0)	20	76.0(3.6)	15	

‘KL-C’ is (Chapelle et al. 2002).

Norm reg.: $L_1 \sim \sum \beta_i = 1$, $L_2 \sim \sum \beta_i^2 = 1$, or unconstrained (No).

Change of reg. changes relationship to radius.

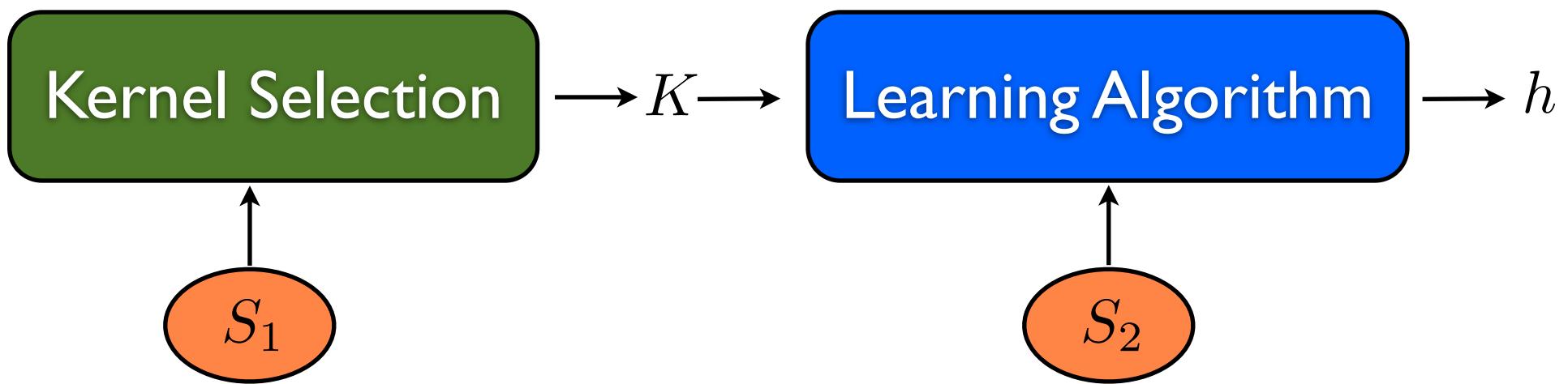
This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

Centered Alignment-Based LK

(Cortes et al., 2010)

- Two stages:



- Outperforms uniform baseline and previous algorithms.
- Centered alignment is key: different from notion used by (Cristianini et al., 2001).

Centered Alignment

(Cortes et al., 2010)

■ Definition:

$$\rho(K, K') = \frac{\mathbf{E}[K_c K'_c]}{\sqrt{\mathbf{E}[K_c^2] \mathbf{E}[K'_c^2]}},$$

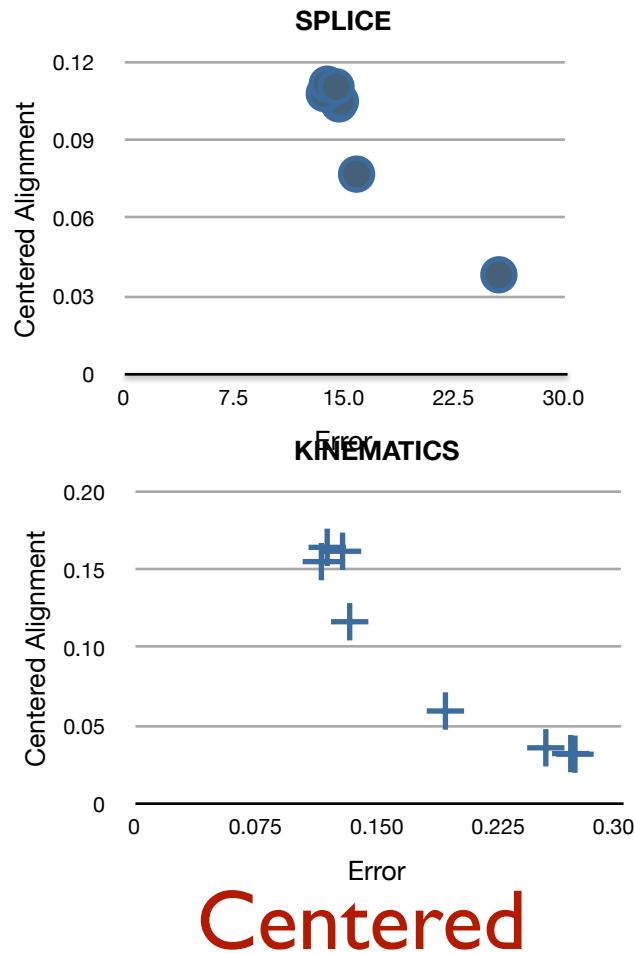
with $K_c(x, x') = (\Phi(x) - \mathbf{E}_x[\Phi])^\top (\Phi(x') - \mathbf{E}_{x'}[\Phi]).$

■ Idea: choose $K \in \mathcal{K}$ maximizing alignment with the labeling kernel (target kernel):

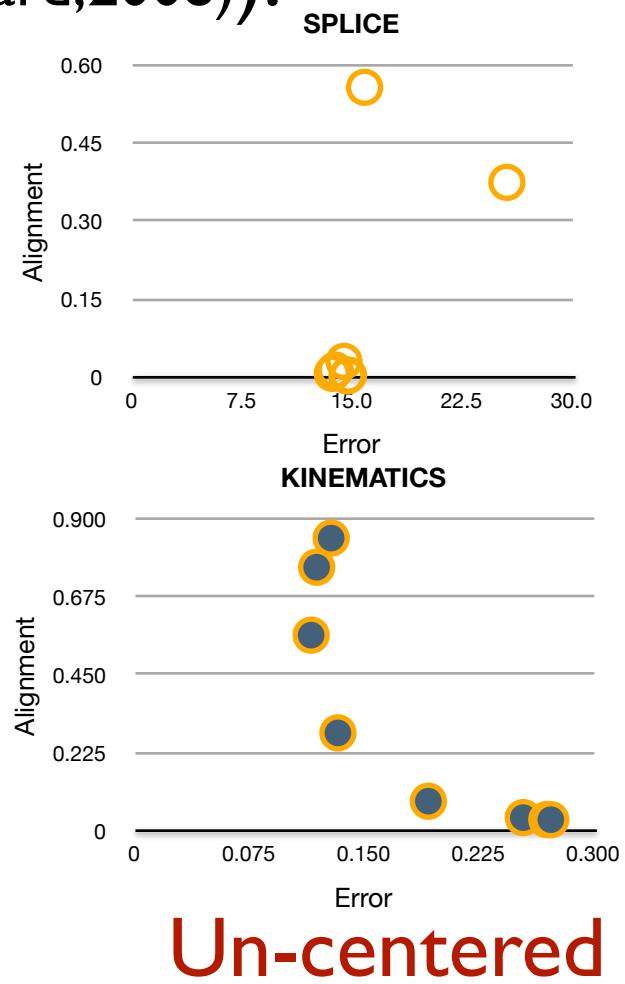
$$K_Y(x, x') = f(x) f(x').$$

Centering

- Centering crucial for correlation with error. See also (Meila et al., 2003; Pothin & Richard, 2008)).



correlation:
-0.95 0.45
-0.96 -0.86



Notes

- (Cortes et al., 2010) comment on (Cristianini, Shawe-Taylor, Elisseeff, Kandola, 2001) and related papers by the same authors:
 - alignment definition **does not correlate well with performance.**
 - thus, poor empirical performance.
 - main **proof** of the paper about the existence of good classifiers is **incorrect**.
 - concentration bound not directly on quantities of interest.

Existence of Good Predictor

- Theorem: let h^* be the hypothesis defined for all x by

$$h^*(x) = \frac{\mathbf{E}_{x'}[y' K_c(x, x')]}{\sqrt{\mathbf{E}[K_c^2]}},$$

and assume normalized labels: $\mathbf{E}[y^2] = 1$. Then,

$$\text{error}(h^*) = \mathbf{E}_x[(h^*(x) - y)^2] \leq 2(1 - \rho(K, K_Y)).$$

Proof

$$\begin{aligned}\mathbf{E}_x[h^{*2}(x)] &= \mathbf{E}_x\left[\frac{\mathbf{E}_{x'}[y'K_c(x, x')]^2}{\mathbf{E}[K_c^2]}\right] \\ &\leq \mathbf{E}_x\left[\frac{\mathbf{E}_{x'}[y'^2]\mathbf{E}_{x'}[K_c^2(x, x')]}{\mathbf{E}[K_c^2]}\right] \\ &= \frac{\mathbf{E}_{x,x'}[K_c^2(x, x')]}{\mathbf{E}[K_c^2]} = 1.\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{E}[(y - h^*(x))^2] &= \mathbf{E}_x[h^*(x)^2] + \mathbf{E}_x[y^2] - 2\mathbf{E}_x[yh^*(x)] \\ &\leq 1 + 1 - 2\rho(K, K_Y).\end{aligned}$$

→ But, alignment between kernel functions unavailable!

Empirical Centered Alignment

■ Definition:

$$\widehat{\rho}(\mathbf{K}, \mathbf{K}') = \frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{\|\mathbf{K}_c\|_F \|\mathbf{K}'_c\|_F}.$$

■ Concentration bound: with probability at least $1 - \delta$,

$$|\rho(K, K') - \widehat{\rho}(\mathbf{K}, \mathbf{K}')| \leq 6\beta \left[\frac{3}{m} + \sqrt{\frac{2 \log \frac{6}{\delta}}{m}} \right],$$

with $\beta = \max(R^4/\mathbf{E}[K_c^2], R^4/\mathbf{E}[K'_c]^2)$.

Algorithm

■ Empirical alignment maximization:

$$\mu^* = \operatorname{argmax}_{\mu \in \Delta_1} \hat{\rho}(\mathbf{K}_\mu, \mathbf{y}\mathbf{y}^\top) = \boxed{\operatorname{argmax}_{\mu \in \Delta_1} \frac{\langle \mathbf{K}_{\mu_c}, \mathbf{y}\mathbf{y}^\top \rangle_F}{\|\mathbf{K}_{\mu_c}\|_F}}$$

with $\mathbf{K}_\mu = \sum_{k=1}^p \mu_k \mathbf{K}_k$.

■ Reduces to simple QP: $\mu^* = \frac{\mathbf{v}^*}{\|\mathbf{v}^*\|}$,

$$\boxed{\mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \geq 0} \mathbf{v}^\top \mathbf{M} \mathbf{v} - 2\mathbf{v}^\top \mathbf{a}},$$

$$\mathbf{a} = (\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \dots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top \rangle_F)^\top, \quad \mathbf{M}_{kl} = \langle \mathbf{K}_{k_c}, \mathbf{K}_{l_c} \rangle_F.$$

Alternative Algorithm

- Based on independent base kernel alignments:

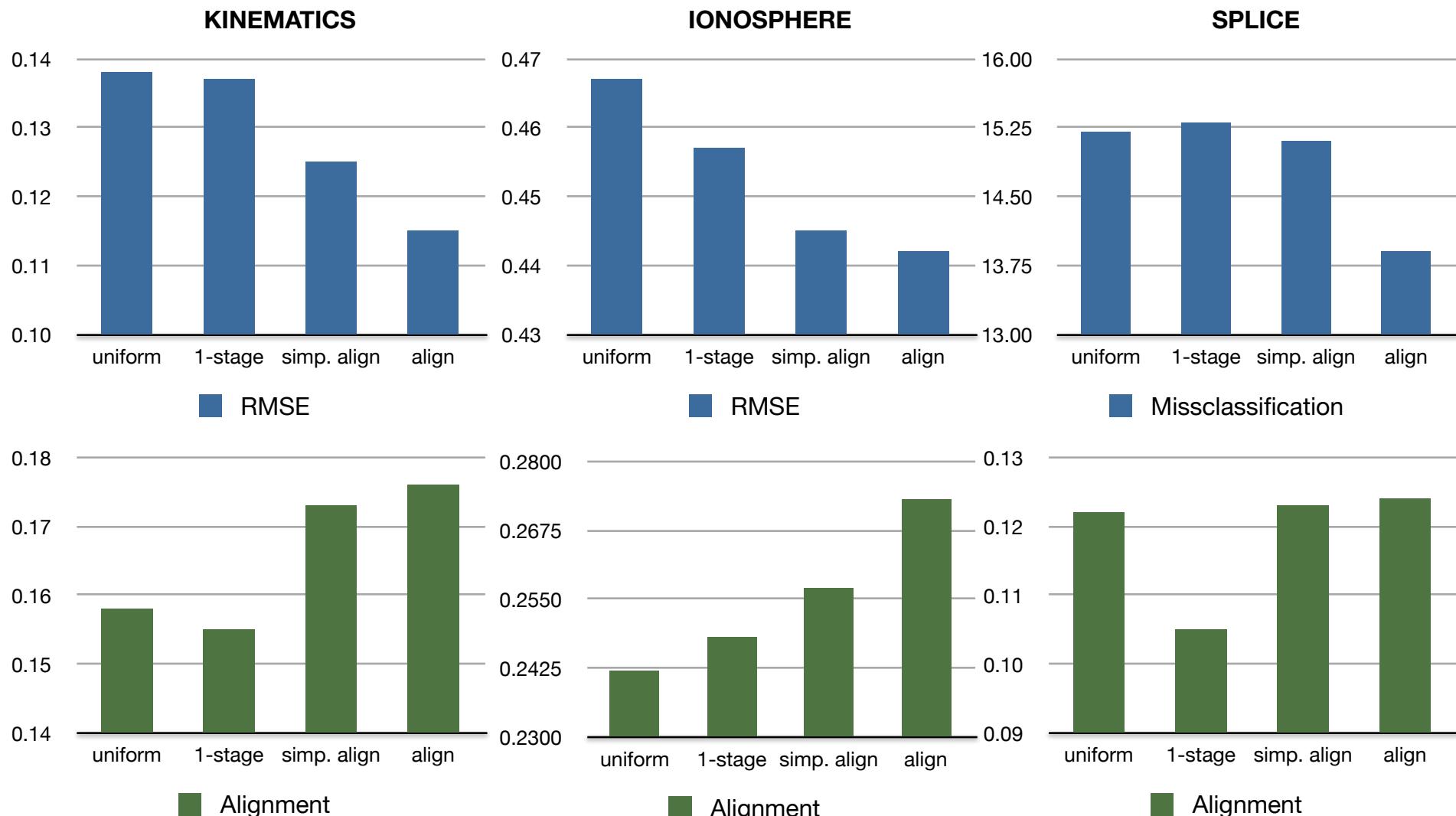
$$\mathbf{K}_\mu \propto \sum_{k=1}^p \hat{\rho}(\mathbf{K}_k, \mathbf{K}_Y) \mathbf{K}_k.$$

- Easily scales to very large numbers of kernels.

Centered Alignment: Reality Check

(Cortes et al., 2010)

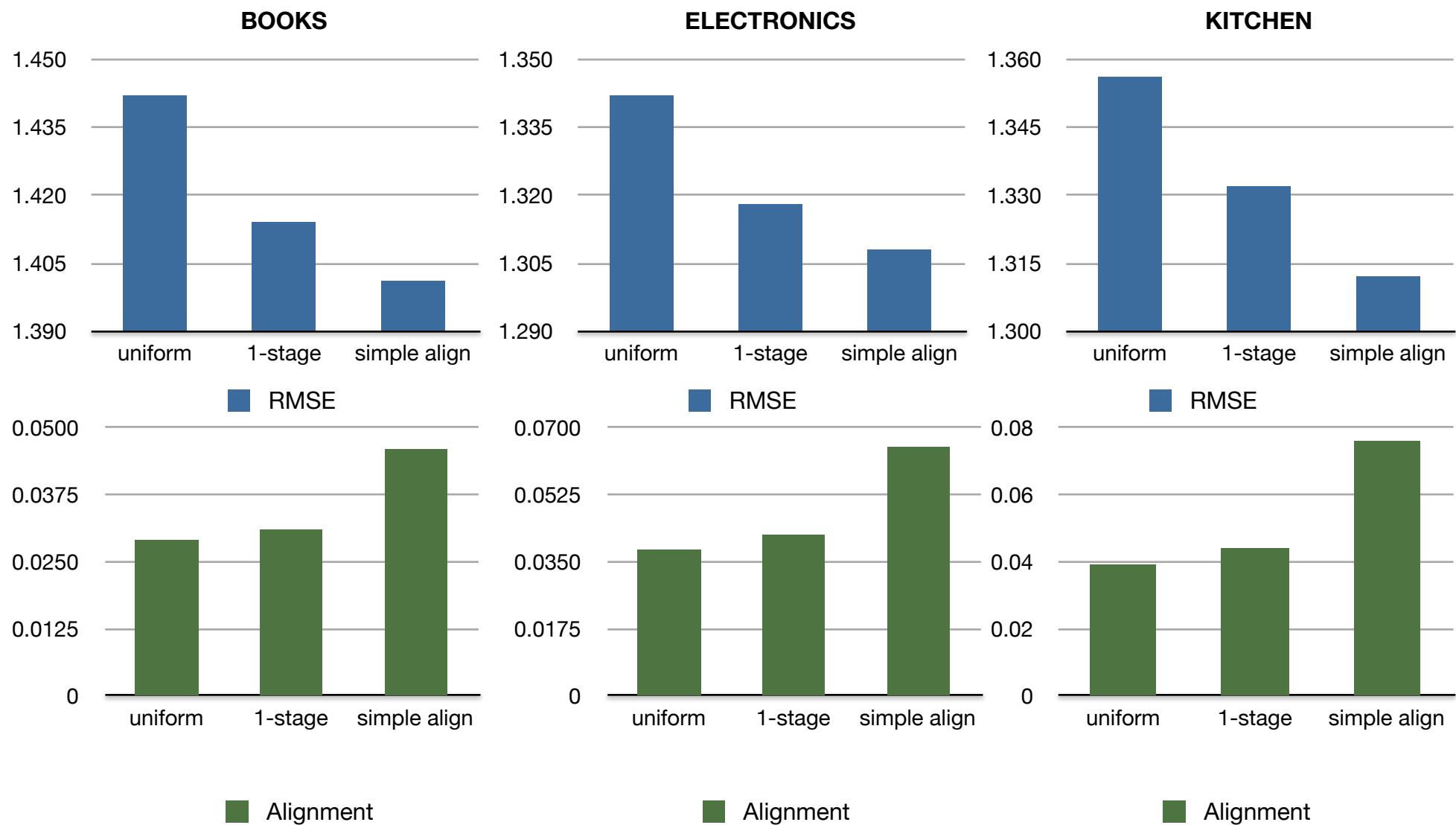
Gaussian base kernels with varying bandwidth.



Centered Alignment: Reality Check

(Cortes et al., 2010)

4,000 rank-1 base kernels.



Centered Alignment-Based LK

- Properties:
 - outperforms uniform combination.
 - based on new definition of centered alignment.
 - effective in classification and regression.
 - proof of existence of good predictors.
 - concentration bound for centered alignment.
 - stability-based generalization bound.
 - algorithm reduced to a simple QP.
- Question: better criterion for first stage?

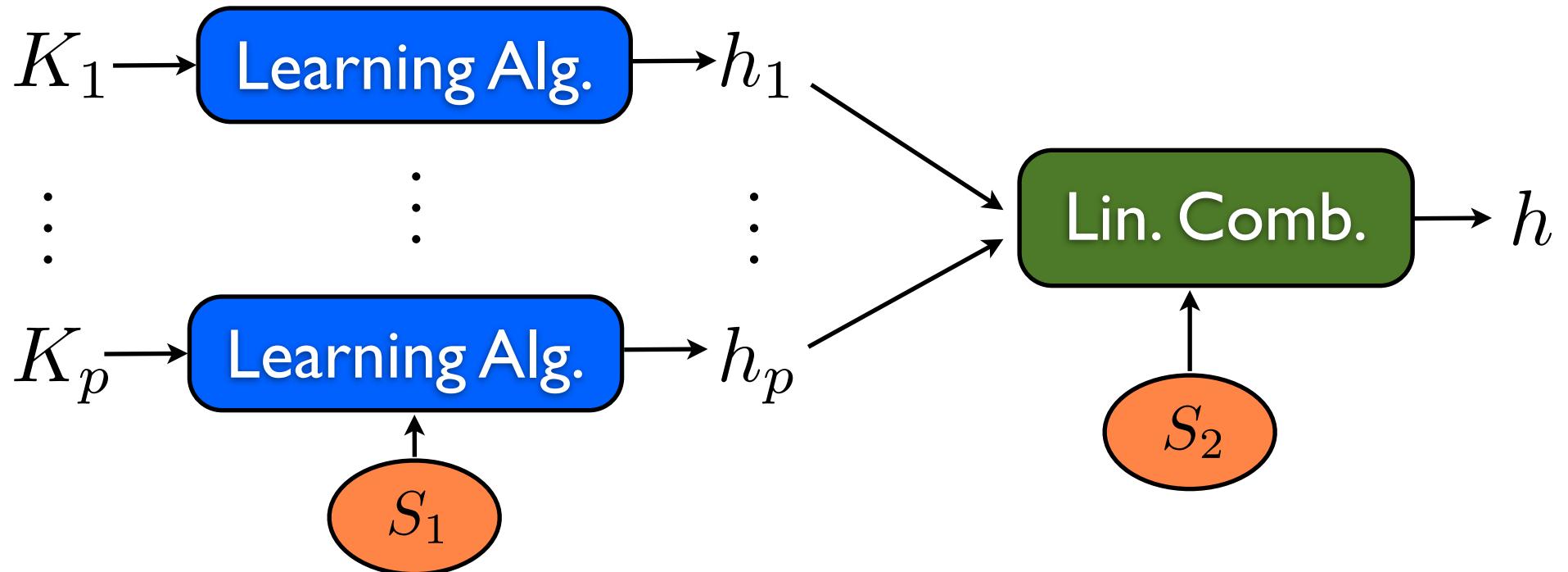
This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

Ensemble Combinations

(Gehler & Nowozin, 2009; Cortes et al., 2011)

- Two stages:



- Standard Learning algorithm in first stage.
- Second stage linearly combines predictions from the first stage, $h(x) = \sum_{i=1}^p \mu_i h_i(x)$.

Ensemble Hypothesis Class

- L_q regularized ensemble:

$$\mathcal{E}_p^q = \left\{ \sum_{k=1}^p \mu_k h_k : \|h_k\|_{\mathbb{H}_k} \leq \Lambda_k, k \in [1, p], \mu \in \Delta_q \right\}.$$

- Note, difference in regularization.
- How do learning kernel (LK) and ensemble kernel (EK) methods compare?
 - Hypothesis complexity.
 - Empirical performance.

Rademacher Complexity

- Let $\eta_0 = 23/22$ and $\Lambda_\star = \max_{k \in [1, p]} \Lambda_k$ furthermore assume, $\forall k \in [1, p], \forall x \in \mathcal{X}$ $K_k(x, x) \leq R^2$, then

$$\widehat{\mathfrak{R}}_S(\mathcal{E}_p^1) \leq \sqrt{\frac{\eta_0 e \lceil \log p \rceil \Lambda_\star^2 R^2}{m}}$$

Same as LK!

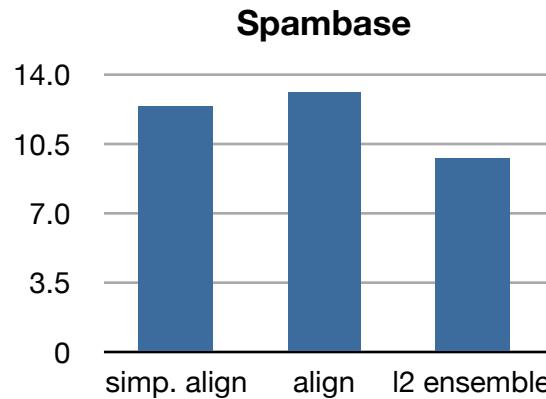
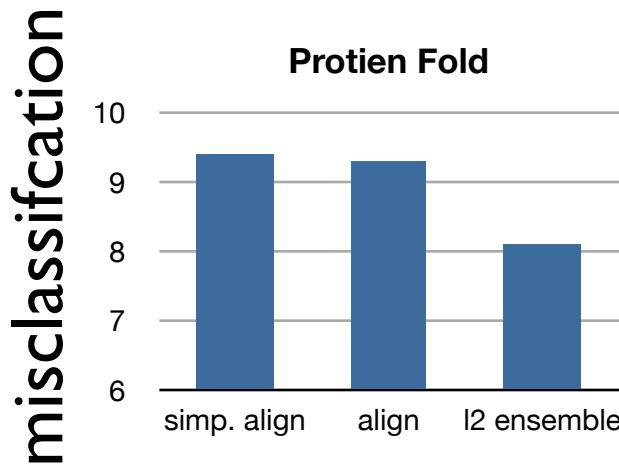
$$\widehat{\mathfrak{R}}_S(\mathcal{E}_p^q) \leq \sqrt{\frac{\eta_0 r p^{\frac{2}{r}} \Lambda_\star^2 R^2}{m}}$$

Differs by
 $p^{1/(2r)}$ factor.

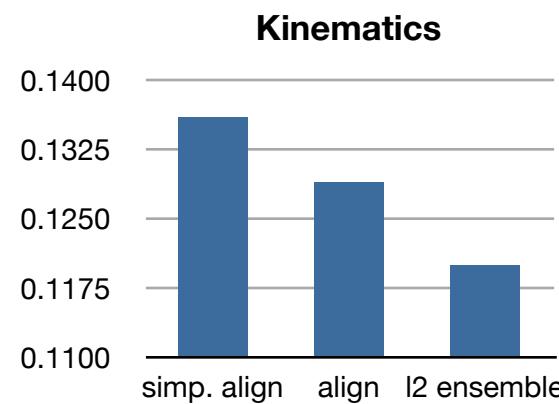
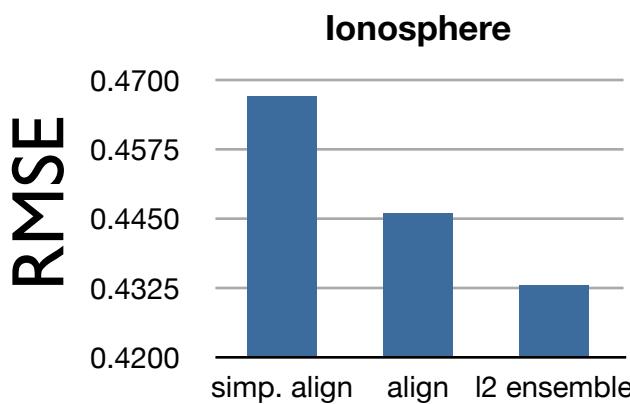
Ensemble Comb.: Reality Check

(Cortes et al., 2011)

Gaussian base kernels



stage 1: SVM
stage 2: L₂-reg SVM



stage 1: KRR
stage 2: KRR

One-Stage Ensemble

- Minimize the error of the ensemble hypothesis:

$$\min_{\mu \in \Delta_q} \min_{h \in \bar{\mathcal{H}}_\mu} \sum_{k=1}^p \lambda_k \|h_k\|_{K_k}^2 + \sum_{i=1}^m L\left(\sum_{k=1}^p \mu_k h_k(x_i), y_i\right)$$

- For $q=1$ optimization reduces to two-stage problem.
- In general, not practical due to cross-validation needed over λ_k for all k .

Multi-Class LPBoost

- (Gehler & Nowozin, 2009) use a multi-class LPBoost based second stage optimization:

$$\min_{\mu, \xi, \rho} -\rho + \frac{1}{\nu N} \sum_{i=1}^m \xi_i$$

$$\text{s.t. } \sum_{k=1}^p \mu_k h_{k,y_i}(x_i) - \underset{y_j \neq y_i}{\operatorname{argmax}} \sum_{k=1}^p \mu_k h_{k,y_j}(x_i) + \xi_i \geq \rho$$

$$\sum_{k=1}^p \mu_k = 1, \mu_k \geq 0$$

Multi-Class LPBoost

- A more complex formulation allows for separate weights for each class:

$$\min_{\mu, \xi, \rho} -\rho + \frac{1}{\nu N} \sum_{i=1}^m \xi_i$$

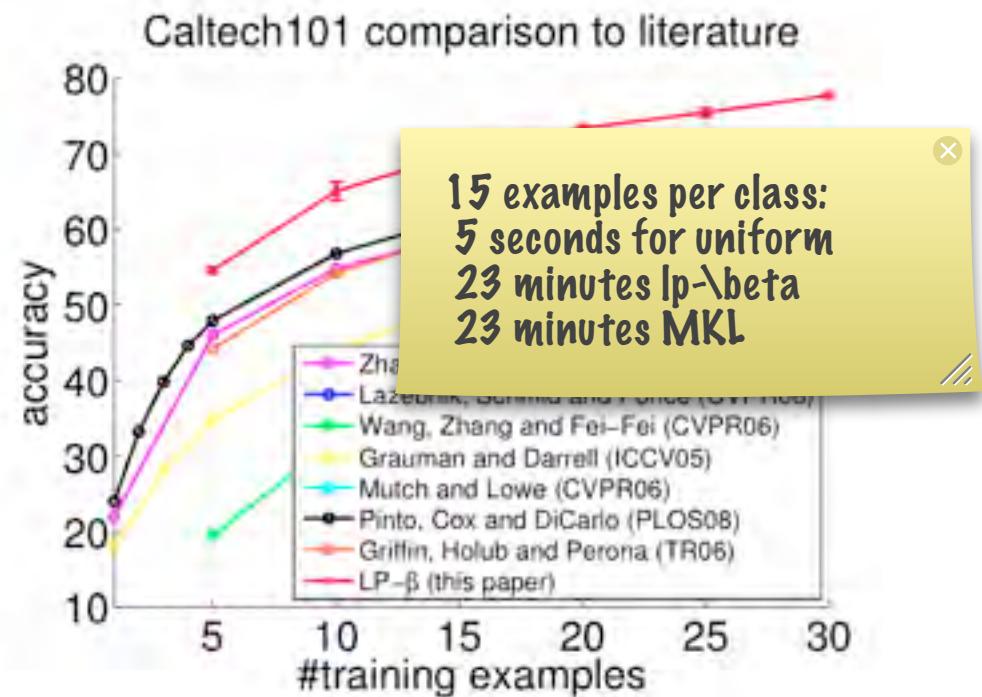
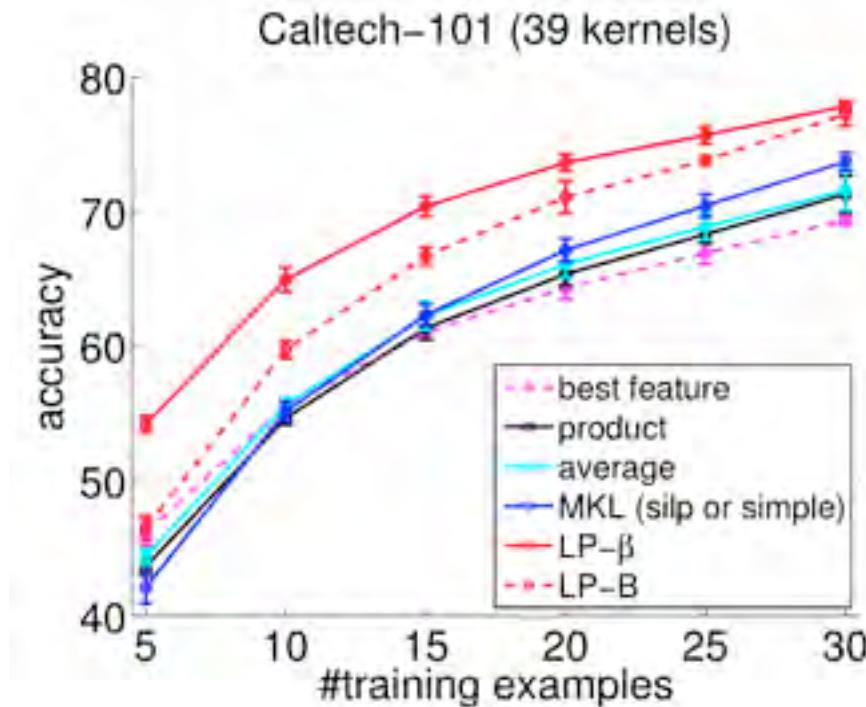
$$\text{s.t. } \sum_{k=1}^p \mu_k^{y_i} h_{k,y_i}(x_i) - \underset{y_j \neq y_i}{\operatorname{argmax}} \sum_{k=1}^p \mu_k^{y_j} h_{k,y_j}(x_i) + \xi_i \geq \rho$$

$$\forall c \in [1, C], \quad \sum_{k=1}^p \mu_k^c = 1, \mu_k^c \geq 0$$

LP- β and LP- β : Reality Check

(Gehler and Nowozin, 2009)

- State-of-the-art performance in multi-class classification for Caltech-101 dataset.
- Two-stage algorithm, combine classifiers trained on individual kernels (39 kernels).



This Part

- Early attempts
- General learning kernel formulation
 - linear, non-negative combinations
 - non-linear combinations and alternative formulations
- Alignment-based algorithms
- Ensemble combinations

DONE

Learning Kernels -Tutorial

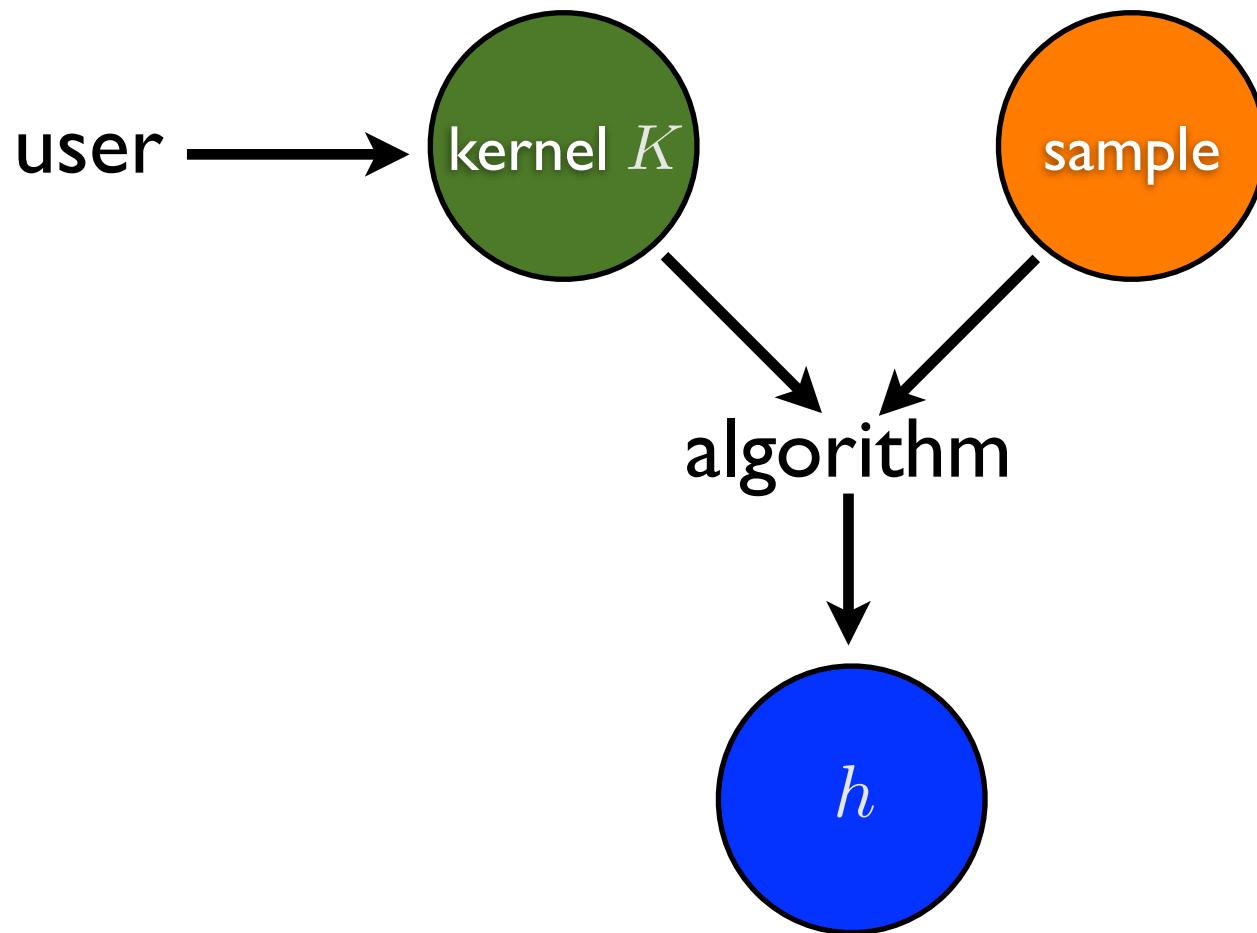
Part III: Theoretical Guarantees.

Corinna Cortes
Google Research
corinna@google.com

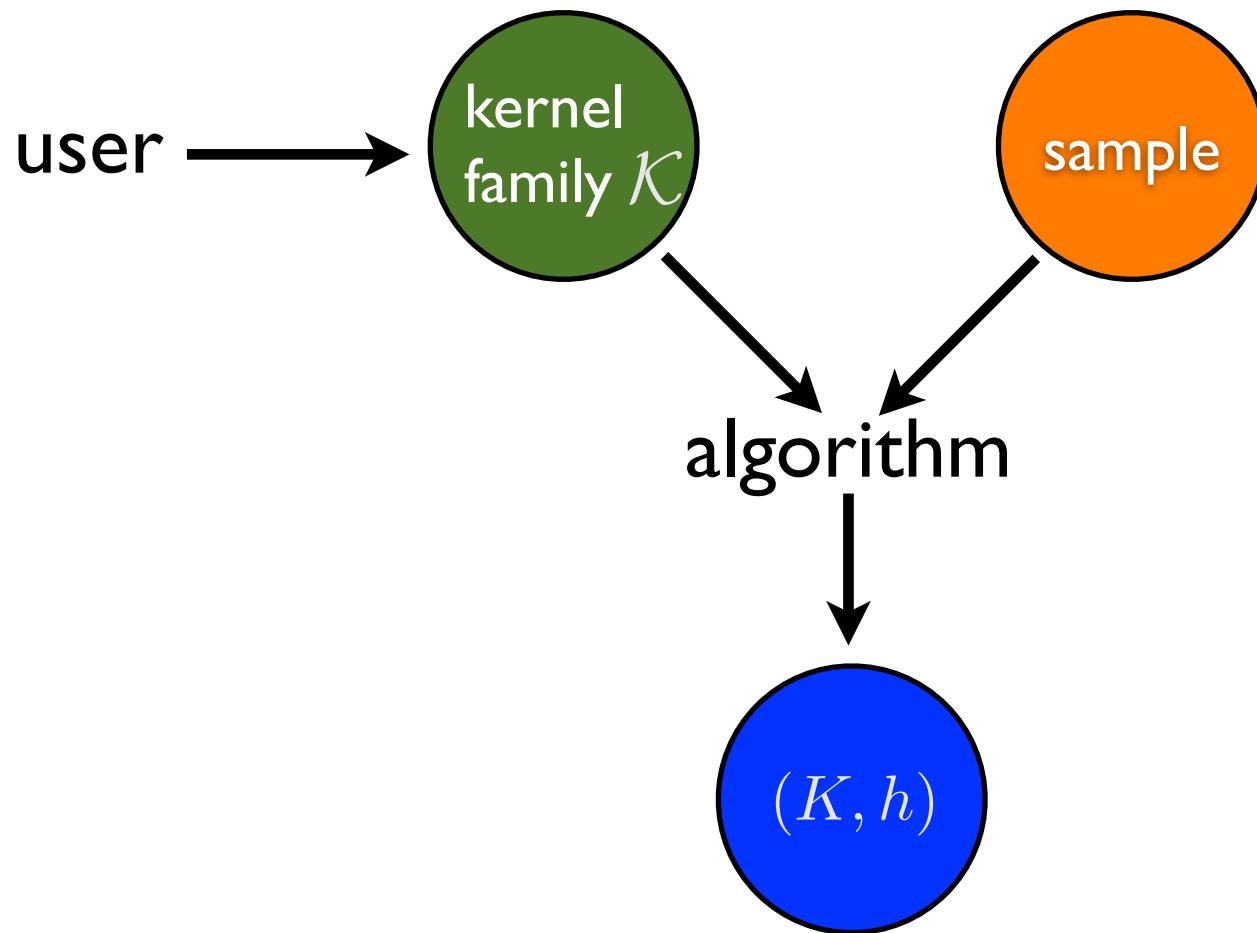
Mehryar Mohri
Courant Institute &
Google Research
mohri@cims.nyu.edu

Afshin Rostami
UC Berkeley
arostami@eecs.berkeley.edu

Standard Learning with Kernels



Learning Kernel Framework



Learning Kernels

■ Theoretical questions:

- what is the price to pay for relaxing the requirement from the user to specify a kernel?
- how does the choice of the kernel family affect generalization?

Part III

- Non-negative combinations.
- General case.

Kernel Families

- Most frequently used kernel families, $q \geq 1$,

$$\mathcal{K}_q = \left\{ \sum_{k=1}^p \mu_k K_k : \boldsymbol{\mu} \in \Delta_q \right\}$$

with $\Delta_q = \left\{ \boldsymbol{\mu} : \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_q = 1 \right\}$.

- Hypothesis sets:

$$H_q = \left\{ h \in \mathbb{H}_K : K \in \mathcal{K}_q, \|h\|_{\mathbb{H}_K} \leq 1 \right\}.$$

Rademacher Complexity

- Empirical Rademacher complexity of H : for a sample $S = (x_1, \dots, x_m)$,

$$\widehat{\mathfrak{R}}_S(H) = \underset{\sigma}{\text{E}} \left[\sup_{h \in H} \frac{1}{m} \sum_{i=1}^m \sigma_i h(x_i) \right],$$

where σ_i s are independent uniform random variables taking values in $\{-1, +1\}$.

- Rademacher complexity of H :

$$\mathfrak{R}_m(H) = \underset{S \sim D^m}{\text{E}} [\widehat{\mathfrak{R}}_S(H)].$$

Single Kernel Margin Bound

- **Theorem** (Koltchinskii and Panchenko, 2002): fix $\rho > 0$. Assume that $K(x, x) \leq R^2$ for all x , then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_1$,

$$R(h) \leq \hat{R}_\rho(h) + 2\sqrt{\frac{R^2/\rho^2}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Early Learning Kernel Bounds

(Bousquet and Herrmann 2003; Lanckriet et al., 2004)

- For any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_1$,

$$R(h) \leq \hat{R}_\rho(h) + \frac{1}{\sqrt{m}} \left[\sqrt{\frac{\max_{k=1}^p \text{Tr}(\mathbf{K}_k) \max_{k=1}^p \frac{\|\mathbf{K}_k\|}{\text{Tr}(\mathbf{K}_k)}}{\rho^2}} + 4 + \sqrt{2 \log \frac{1}{\delta}} \right].$$

- but, bound always greater than one (Srebro and Ben-David, 2006)!
- other bound of (Lanckriet et al., 2004) for linear combination case also always greater than one!

Multiplicative Learning Bound

(Lanckriet et al., 2004)

- Assume that for all $k \in [1, p]$, $K_k(x, x) \leq R^2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_1$,

$$R(h) \leq \hat{R}_\rho(h) + O\left(\sqrt{\frac{p R^2 / \rho^2}{m}}\right).$$

- bound multiplicative in p (number of kernels).

Additive Learning Bound

(Srebro and Ben-David, 2006)

- Assume that for all $k \in [1, p]$, $K_k(x, x) \leq R^2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_1$,

$$R(h) \leq \hat{R}_\rho(h) + \sqrt{8 \frac{2 + p \log \frac{128em^3R^2}{\rho^2p} + 256 \frac{R^2}{\rho^2} \log \frac{\rho em}{8R} \log \frac{128mR^2}{\rho^2} + \log(1/\delta)}{m}}.$$

- bound additive in p (modulo log terms).
- not informative for $p > m$.
- based on pseudo-dimension of kernel family.
- similar guarantees for other families.

New Data-Dependent Bound

(CC, MM, and AR, 2010)

- **Theorem:** for any sample S of size m , and positive integer r ,

$$\widehat{\mathfrak{R}}_S(H_1) \leq \frac{\sqrt{\frac{23}{22}r\|\mathbf{u}\|_r}}{m},$$

with $\mathbf{u} = (\text{Tr}[\mathbf{K}_1], \dots, \text{Tr}[\mathbf{K}_p])^\top$.

- similarity with single kernel bound.
- can be used directly to derive an algorithm.

New Data-Dependent Bound

■ Proof: Let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$.

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H_q) &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in H_q} \sum_{i=1}^m \sigma_i h(x_i) \right] \\ &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{\mu} \in \Delta_q, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \leq 1} \sum_{i,j=1}^m \sigma_i \alpha_j K_{\boldsymbol{\mu}}(x_i, x_j) \right] \\ &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{\mu} \in \Delta_q, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \leq 1} \boldsymbol{\sigma}^\top \mathbf{K}_{\boldsymbol{\mu}} \boldsymbol{\alpha} \right] = \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{\mu} \in \Delta_q, \|\boldsymbol{\alpha}\|_{\mathbf{K}^{1/2}} \leq 1} \langle \boldsymbol{\sigma}, \boldsymbol{\alpha} \rangle_{\mathbf{K}^{1/2}} \right] \\ &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{\mu} \in \Delta_q} \sqrt{\boldsymbol{\sigma}^\top \mathbf{K}_{\boldsymbol{\mu}} \boldsymbol{\sigma}} \right] \quad (\text{Cauchy-Schwarz}) \\ &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{\mu} \in \Delta_q} \sqrt{\boldsymbol{\mu} \cdot \mathbf{u}_{\boldsymbol{\sigma}}} \right] \quad [\mathbf{u}_{\boldsymbol{\sigma}} = (\boldsymbol{\sigma}^\top \mathbf{K}_1 \boldsymbol{\sigma}, \dots, \boldsymbol{\sigma}^\top \mathbf{K}_p \boldsymbol{\sigma})^\top] \\ &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r} \right]. \quad (\text{definition of dual norm})\end{aligned}$$

New Data-Dependent Bound

■ Proof: in the following, $r \geq 1$ is arbitrary integer.

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H_1) &= \frac{1}{m} \underset{\boldsymbol{\sigma}}{\mathbb{E}} \left[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_{\infty}} \right] \\ &\leq \frac{1}{m} \underset{\boldsymbol{\sigma}}{\mathbb{E}} \left[\sqrt{\|\mathbf{u}_{\boldsymbol{\sigma}}\|_r} \right] \quad (\forall r \geq 1, \|\mathbf{u}_{\boldsymbol{\sigma}}\|_{\infty} \leq \|\mathbf{u}_{\boldsymbol{\sigma}}\|_r) \\ &= \frac{1}{m} \underset{\boldsymbol{\sigma}}{\mathbb{E}} \left[\left[\sum_{k=1}^p (\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r \right]^{\frac{1}{2r}} \right] \\ &\leq \frac{1}{m} \left[\underset{\boldsymbol{\sigma}}{\mathbb{E}} \left[\sum_{k=1}^p (\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r \right] \right]^{\frac{1}{2r}} \text{ (Jensen's inequality)} \\ &= \frac{1}{m} \left[\sum_{k=1}^p \underset{\boldsymbol{\sigma}}{\mathbb{E}} \left[(\boldsymbol{\sigma}^\top \mathbf{K}_k \boldsymbol{\sigma})^r \right] \right]^{\frac{1}{2r}} \\ &\leq \frac{1}{m} \left[\sum_{k=1}^p \left(\frac{23}{22} r \operatorname{Tr}[\mathbf{K}_k] \right)^r \right]^{\frac{1}{2r}} = \frac{\sqrt{\frac{23}{22} r \|\mathbf{u}\|_r}}{m}. \quad \text{(lemma)}\end{aligned}$$

Key Lemma

- **Lemma:** Let \mathbf{K} be a kernel matrix for a finite sample. Then, for any integer r ,

$$\underset{\boldsymbol{\sigma}}{\text{E}} \left[(\boldsymbol{\sigma}^\top \mathbf{K} \boldsymbol{\sigma})^r \right] \leq \left(\frac{23}{22} r \text{Tr}[\mathbf{K}] \right)^r.$$

- proof based on combinatorial argument.

New Learning Bound - LI

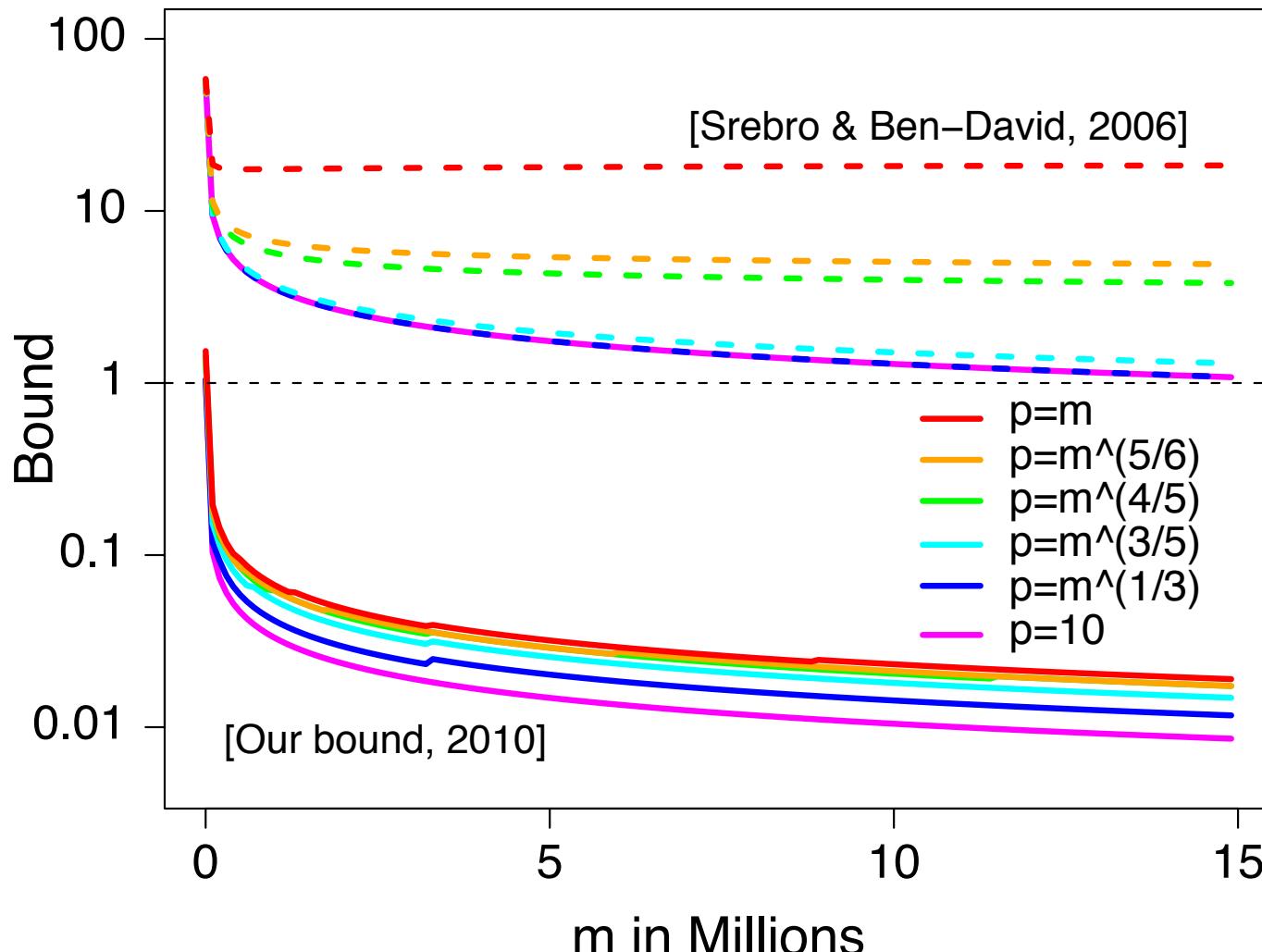
(CC, MM, and AR, 2010)

- **Theorem:** assume that for all $k \in [1, p]$, $K_k(x, x) \leq R^2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_1$,

$$R(h) \leq \widehat{R}_\rho(h) + 2\sqrt{\frac{\frac{23}{22}e[\log p]R^2/\rho^2}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- very weak dependency on p , no extra \log terms.
- analysis based on Rademacher complexity.
- bound valid for $p \gg m$.
- see also (Kakade et al., 2010).

Comparison



$$\rho/R = .2$$

Lower Bound

■ Tight bound:

- dependency $\sqrt{\log p}$ cannot be improved.
- argument based on VC dimension or example.

■ Observations: case $\mathcal{X} = \{-1, +1\}^p$.

- canonical projection kernels $K_k(\mathbf{x}, \mathbf{x}') = x_k x'_k$.
- H_1 contains $J_p = \{\mathbf{x} \mapsto s x_k : k \in [1, p], s \in \{-1, +1\}\}$.
- $\text{VCdim}(J_p) = \Omega(\log p)$.
- for $\rho = 1$ and $h \in J_p$, $\widehat{R}_\rho(h) = \widehat{R}(h)$.
- VC lower bound: $\Omega(\sqrt{\text{VCdim}(J^p)/m})$.

New Paper

- **Recent claim** (Hussain and Shawe-Taylor, AISTATS 2011): additive bound in terms of $\log p$, instead of multiplicative.
 - main proof incorrect: probabilistic bound on Rademacher complexity, but slack term left out of proof of theorem 8. Adding it → **multiplicative bound**.
 - however: authors are preparing new version (private communication: J. Shawe-Taylor).

New Learning Bound - Lq

(CC, MM, and AR, 2010)

- **Theorem:** let $q, r \geq 1$ with $\frac{1}{q} + \frac{1}{r} = 1$ and r integer.
Assume that for all $k \in [1, p]$, $K_k(x, x) \leq R^2$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_q$,

$$R(h) \leq \hat{R}_\rho(h) + 2p^{\frac{1}{2r}} \sqrt{\frac{\frac{23}{22}rR^2/\rho^2}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- mild dependency on p .
- analysis based on Rademacher complexity.

Lower Bound

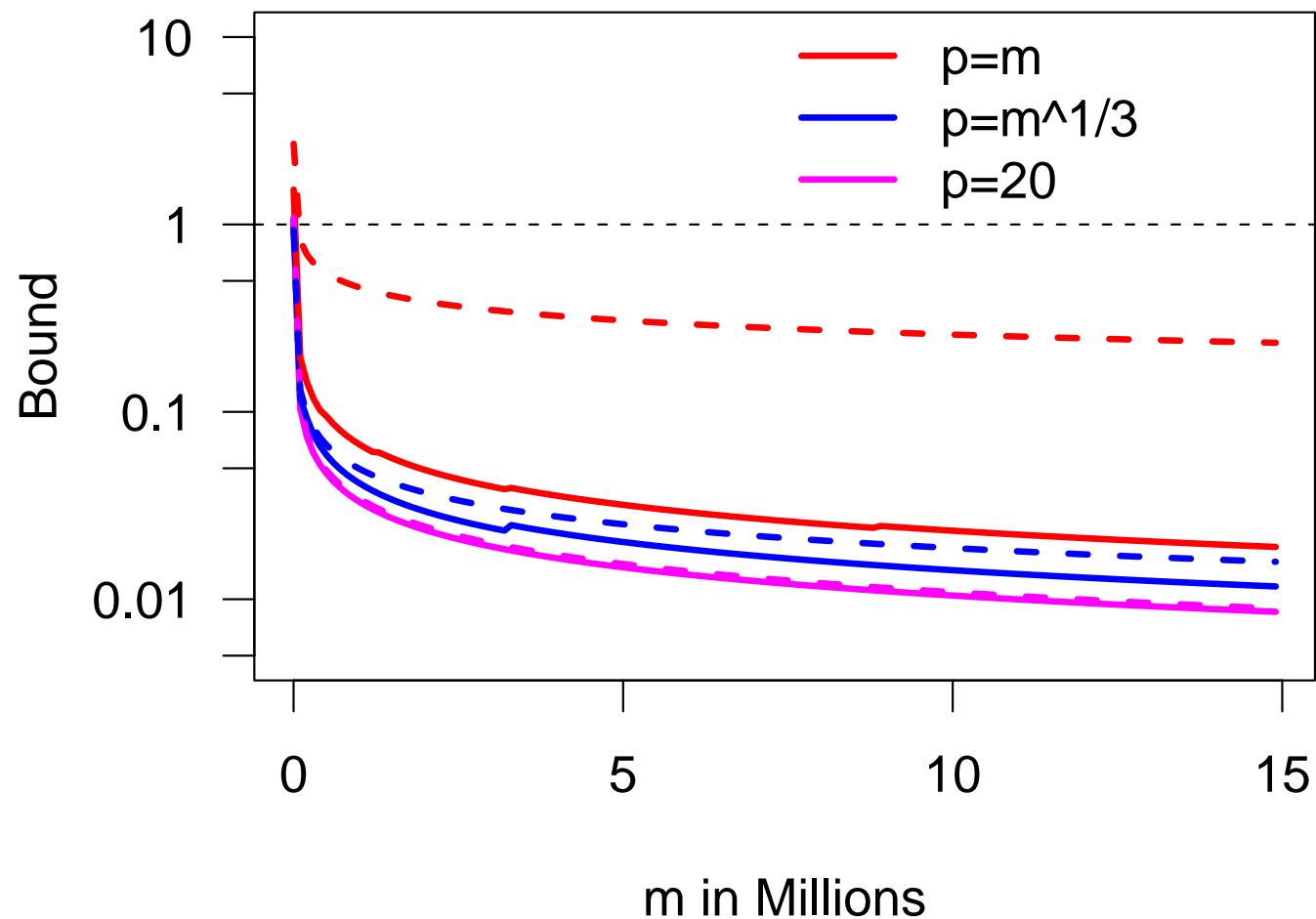
■ Tight bound:

- dependency $p^{\frac{1}{2r}}$ cannot be improved.
- in particular $p^{\frac{1}{4}}$ tight for L_2 regularization.

■ Observations: equal kernels.

- $\sum_{k=1}^p \mu_k K_k = \left(\sum_{k=1}^p \mu_k \right) K_1$.
- thus, $\|h\|_{\mathbb{H}_{K_1}}^2 = (\sum_{k=1}^p \mu_k) \|h\|_{\mathbb{H}_K}^2$ for $\sum_{k=1}^p \mu_k \neq 0$.
- $\sum_{k=1}^p \mu_k \leq p^{\frac{1}{r}} \|\mu\|_q = p^{\frac{1}{r}}$ (Hölder's inequality).
- H_q coincides with $\{h \in \mathbb{H}_{K_1} : \|h\|_{\mathbb{H}_{K_1}} \leq p^{\frac{1}{2r}}\}$.

Comparison L1 vs L2



Conclusion

- **Theory:** tight generalization bounds for learning kernels with L_1 or L_q regularization (p dependency).
 - mild dependency on p .
 - similar proof and analysis for other regularizations.
- **Applications:**
 - results suggest using large number of kernels.
 - recent results show significant improvements (CC, MM, AR, ICML 2010).

Part III

- Non-negative combinations.
- General case.

Kernel Family

■ General case:

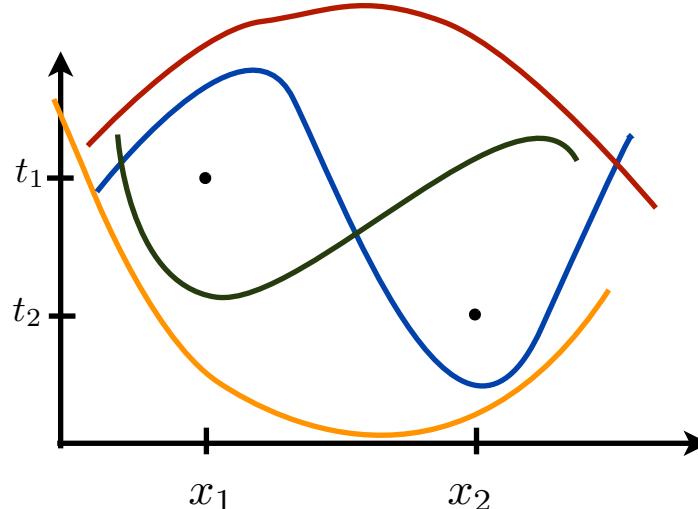
- \mathcal{K} a family of kernels bounded by R .
- finite pseudo-dimension: $\text{Pdim}(\mathcal{K}) < \infty$.
- general hypothesis set:

$$H_{\mathcal{K}} = \left\{ h \in \mathbb{H}_K : K \in \mathcal{K}, \|h\|_{\mathbb{H}_K} \leq 1 \right\}.$$

Shattering

- **Definition:** Let H be a hypothesis set of functions from X to \mathbb{R} . $A = \{x_1, \dots, x_m\}$ is **shattered** by H if there exist $t_1, \dots, t_m \in \mathbb{R}$ such that

$$\left| \left\{ \begin{bmatrix} \operatorname{sgn}(L(h(x_1), f(x_1)) - t_1) \\ \vdots \\ \operatorname{sgn}(L(h(x_m), f(x_m)) - t_m) \end{bmatrix} : h \in H \right\} \right| = 2^m.$$

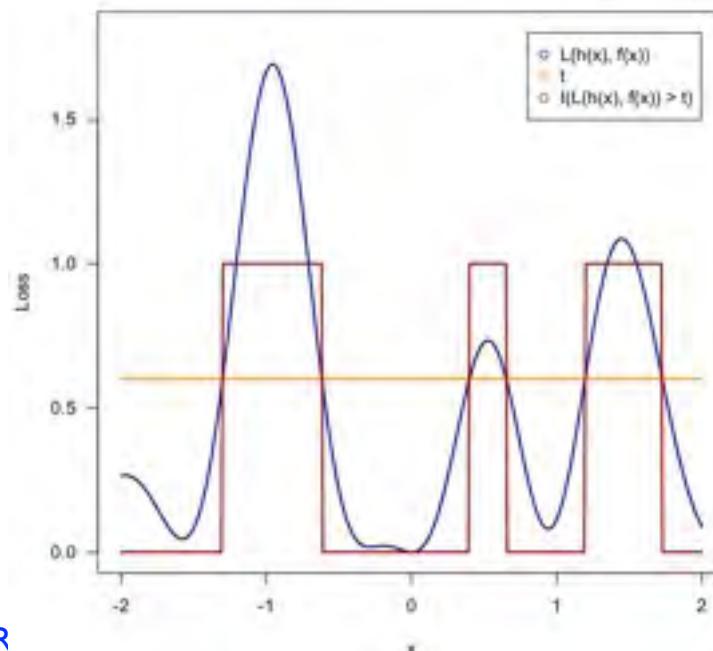


Pseudo-Dimension

(Pollard, 1984)

- **Definition:** Let H be a hypothesis set of functions from X to \mathbb{R} . The pseudo-dimension of H , $\text{Pdim}(H)$, is the size of the largest set shattered by H .
- **Definition (equivalent, see also (Vapnik, 1995)):**

$$\text{Pdim}(H) = \text{VCdim}\left(\{(x, t) \mapsto 1_{(h(x)-t)>0} : h \in H\}\right).$$



Pseudo-Dimension - Properties

- **Theorem:** Pseudo-dimension of hyperplanes.

$$\text{Pdim}(\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b: \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}) = N + 1.$$

- **Theorem:** Pseudo-dimension of a vector space of real-valued functions H .

$$\text{Pdim}(H) = \dim(H).$$

- **Theorem:** Pseudo-dimension of $\phi(H) = \{\phi \circ h: h \in H\}$ where ϕ is a monotone function:

$$\text{Pdim}(\phi(H)) \leq \dim(H).$$

General Pdim Learning Bound

(Srebro and Ben-David, 2006)

- Let \mathcal{K} a family of kernel functions bounded by R .
Let $d = \text{Pdim}(\mathcal{K})$, then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H_{\mathcal{K}}$,

$$R(h) \leq \widehat{R}_{\rho}(h) + \sqrt{8 \frac{2 + d \log \frac{128em^3R^2}{\rho^2d} + 256 \frac{R^2}{\rho^2} \log \frac{\rho em}{8R} \log \frac{128mR^2}{\rho^2} + \log(1/\delta)}{m}}.$$

- bound additive in d (modulo log terms).
- not informative for $d > m$.

Application: Linear Combinations

- Linear and non-negative combination of base kernels (previous section):

$$\mathcal{K}_{\text{lin}} = \left\{ K_{\boldsymbol{\mu}} = \sum_{k=1}^p \mu_k K_k : \left(\sum_{k=1}^p \mu_k = 1 \right) \wedge \left(\mathbf{K}_{\boldsymbol{\mu}} \succeq \mathbf{0} \right) \right\}$$

$$\mathcal{K}_1 = \left\{ K_{\boldsymbol{\mu}} = \sum_{k=1}^p \mu_k K_k : \left(\sum_{k=1}^p \mu_k = 1 \right) \wedge \left(\boldsymbol{\mu} \geq \mathbf{0} \right) \right\}$$

- Since $\mathcal{K}_{\text{lin}} \subseteq \mathcal{K}_1 \subseteq \left\{ \sum_{k=1}^p \mu_k K_k \right\}$,

$$\text{Pdim}(\mathcal{K}_1) \leq \text{Pdim}(\mathcal{K}_{\text{lin}}) \leq \dim \left(\left\{ \sum_{k=1}^p \mu_k K_k \right\} \right) = p.$$

Application: Gaussian Kernels

■ Gaussian kernels with a fixed covariance matrix:

$$\mathcal{K}_{\text{Gaussian}} = \left\{ (\mathbf{x}_1, \mathbf{x}_2) \mapsto \exp(-(\mathbf{x}_2 - \mathbf{x}_1)^\top \mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1)) : \mathbf{A} \in \mathbb{S}_+^N \right\}.$$

- since \exp is monotone and since

$$\begin{aligned} & \left\{ (\mathbf{x}_1, \mathbf{x}_2) \mapsto (\mathbf{x}_2 - \mathbf{x}_1)^\top \mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) : \mathbf{A} \in \mathbb{S}_+^N \right\} \\ &= \left\{ (\mathbf{x}_1, \mathbf{x}_2) \mapsto \sum_{i,j=1}^n \mathbf{A}_{ij} (\mathbf{x}_2 - \mathbf{x}_1)_i (\mathbf{x}_2 - \mathbf{x}_1)_j : \mathbf{A} \in \mathbb{S}_+^N \right\} \\ &\subseteq \text{span} \left\{ (\mathbf{x}_1, \mathbf{x}_2) \mapsto (\mathbf{x}_2 - \mathbf{x}_1)_i (\mathbf{x}_2 - \mathbf{x}_1)_j : 1 \leq i \leq j \leq N \right\}, \end{aligned}$$

- $\text{Pdim}(\mathcal{K}_{\text{Gaussian}}) \leq \frac{N(N-1)}{2}$.
- Similar for \mathbf{A} diagonal, $\text{Pdim}(\mathcal{K}_{\text{Gaussian}}) \leq N$.

References

- Bousquet, Olivier and Herrmann, Daniel J. L. On the complexity of learning the kernel matrix. In NIPS, 2002.
- Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Generalization Bounds for Learning Kernels. In ICML, 2010.
- Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Two-stage learning kernel methods. In ICML, 2010.
- Zakria Hussain, John Shawe-Taylor. Improved Loss Bounds For Multiple Kernel Learning. In AISTATS, 2011.
- Kakade, Sham M., Shalev-Shwartz, Shai, and Tewari, Ambuj. Applications of strong convexity–strong smoothness duality to learning with matrices, 2010. arXiv:0910.0610v1.
- Koltchinskii, V. and Panchenko, D. Empirical margin distributions and bounding the generalization error of combined classifiers. Annals of Statistics, 30, 2002.
- Koltchinskii, Vladimir and Yuan, Ming. Sparse recovery in large ensembles of kernel machines on-line learning and bandits. In COLT, 2008.

References

- Lanckriet, Gert, Cristianini, Nello, Bartlett, Peter, Ghaoui, Laurent El, and Jordan, Michael. Learning the kernel matrix with semidefinite programming. JMLR, 5, 2004.
- Srebro, Nathan and Ben-David, Shai. Learning bounds for support vector machines with learned kernels. In COLT, 2006.
- Ying, Yiming and Campbell, Colin. Generalization bounds for learning the kernel problem. In COLT, 2009.

Learning Kernels -Tutorial

Part IV: Software Solutions

Corinna Cortes
Google Research
corinna@google.com

Mehryar Mohri
Courant Institute &
Google Research
mohri@cims.nyu.edu

Afshin Rostami
UC Berkeley
arostami@eecs.berkeley.edu

This Part

- Software Solutions
- Demo

Open-Source Software

- Provide easy-to-use and efficient implementations of useful learning kernel algorithms.
- Allows end-users to combine standard as well as specialized domain-specific kernels.
- Allow researchers to easily compare against established learning kernel algorithms.
- Allow developers to makes use of and extend existing algorithms.

Open-Source Software

■ Libraries and single algorithms - a starting point:

- SHOGUN <http://www.shogun-toolbox.org>
- OpenKernel.org <http://www.openkernel.org>
- DOGMA (online alg: UFO) <http://dogma.sourceforge.net/index.html>
- MKL-SMO, HKL <http://www.di.ens.fr/~fbach/index.htm#software>
- SMO q-norm, GMKL <http://research.microsoft.com/~manik/>
- SimpleMKL <http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html>
- Mixed-Norm <http://www.cse.iitb.ac.in/saketh/research.html>
- DC-program <http://www.cs.ucl.ac.uk/staff/A.Argyriou/code/dc/>
- LP-B, LP- β <http://www.vision.ee.ethz.ch/~pgehler/>
- MC-MKL <http://http://www.fml.tuebingen.mpg.de/raetsch/>

SHOGUN

- www.shogun-toolbox.org
S. Sonnenburg, G. Raetsch, S. Henschel
- Large scale kernel methods, focusing on SVM
- MATLAB, R, Octave and Python interfaces.
- Choice of LibSVM, Liblinear, SVMLight for internal solver.
- Standard kernels (e.g. Gaussian) as well as some string kernels (e.g. Locality Improved, Fischer).
- LI-combinations, SILP implementation [Sonnenburg et al., 2006]
- Lq-combinations ($q > 1$), specialized interleaved optimization or Newton step wrapper method [Kloft et al., NIPS 2009]



OpenKernel

- www.openkernel.org
Cyril Allauzen, Mehryar Mohri, Afshin Rostami
- Supports standard kernels, general rational kernels (string kernels) and custom pre-computed kernels.
- Interfaces to LibSVM, includes Kernel Ridge Regression implementation.
- L1-regularized positive linear combinations
- L2-regularized positive linear combinations [Cortes et al., UAI 2009]
- L2-positive quadratic combinations [Cortes et al., NIPS 2009]
- Two-stage alignment based methods [Cortes et al., ICML 2010]

OpenKernel

- Command-line programs:
- *klcombinekernels*: combine pre-computed kernels
 - LibSVM or binary format.
- *klcombinefeatures*: combine explicit feature mappings
 - Supports sparse mappings, millions of features.
- *klweightfeatures*: weights individual features, i.e. rank-1 kernel combinations
 - Feature weighting/selection.

OpenKernel

[DEMO]