Learning Kernels - Tutorial

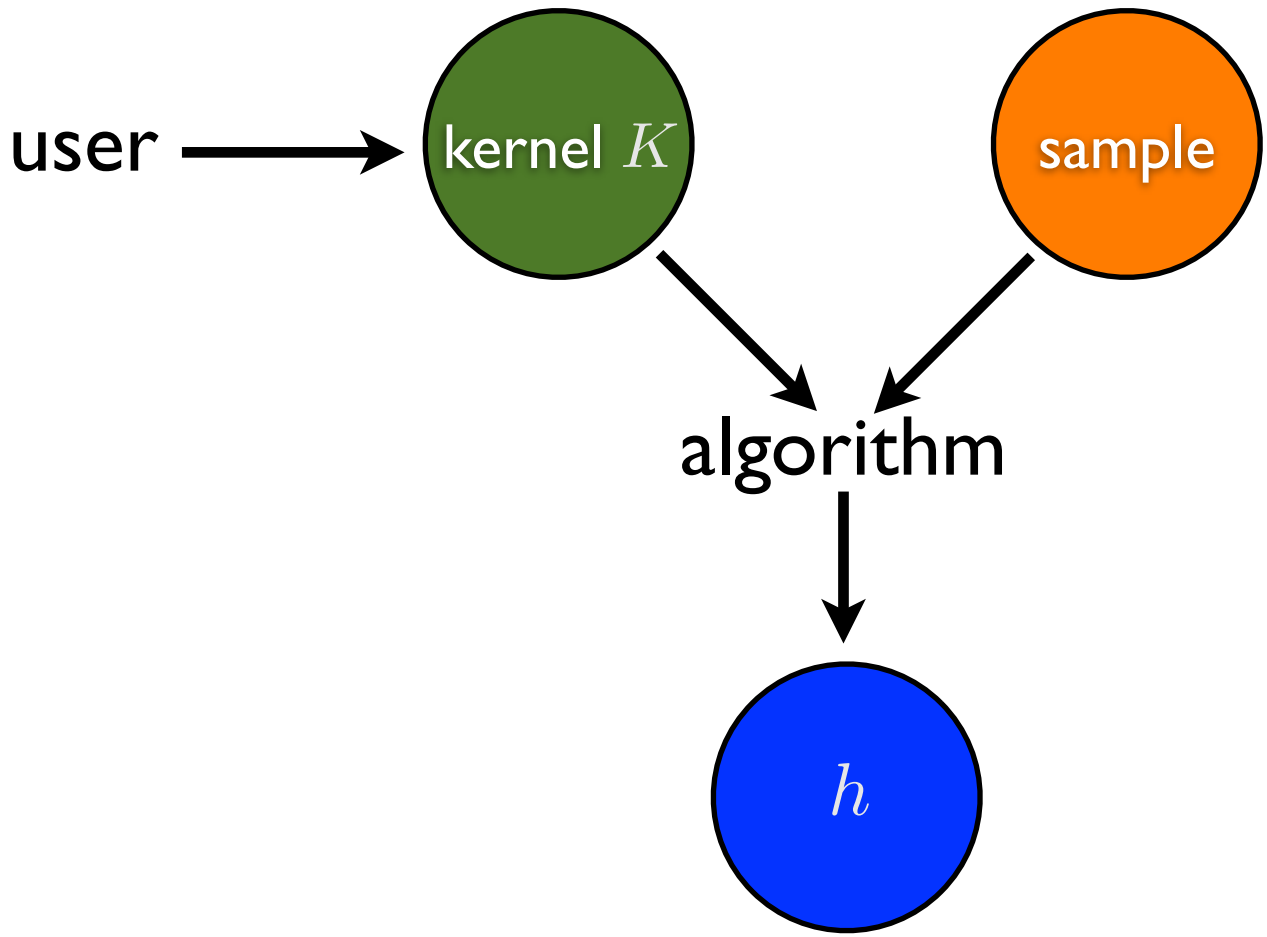
Part II: Learning Kernel Algorithms.

Corinna Cortes
Google Research
corinna@google.com

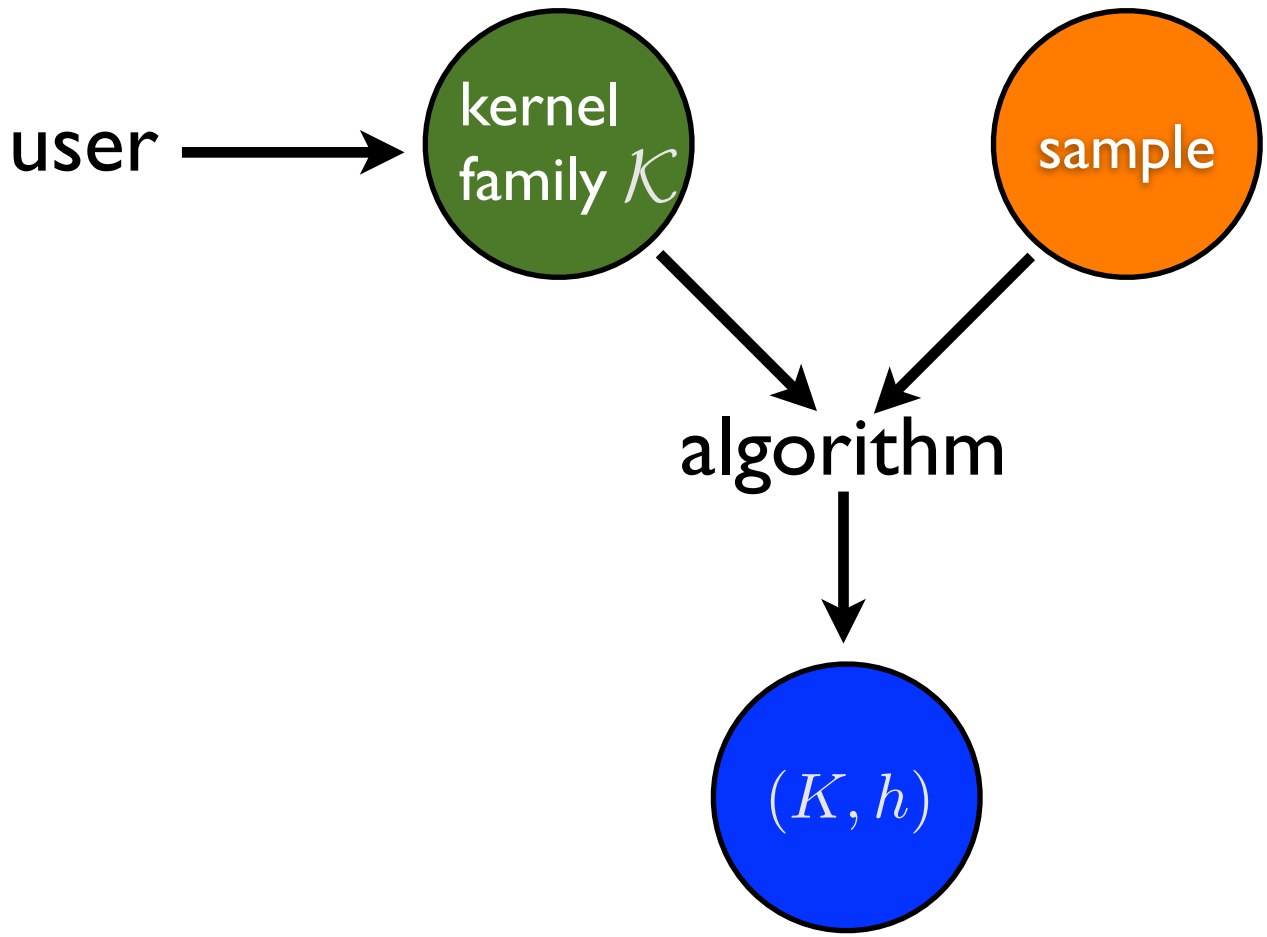
Mehryar Mohri
Courant Institute &
Google Research
mohri@cims.nyu.edu

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

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:

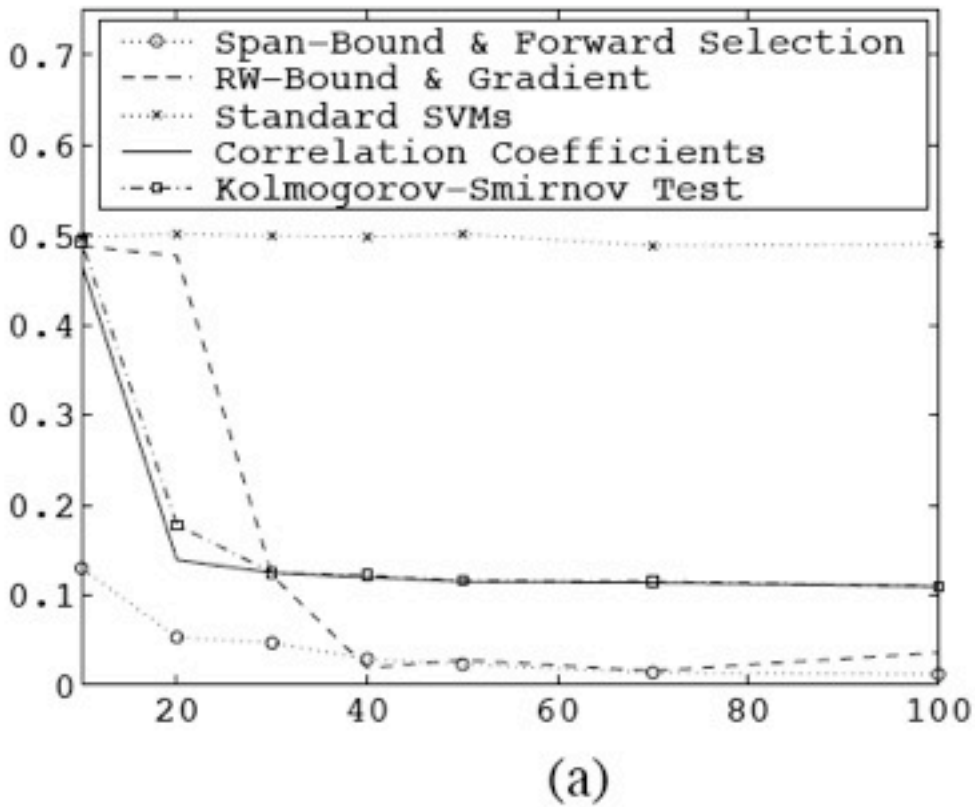
$$K(x_i, x_j) = \left(1 + \sum_{k=1}^{p} \mu_k x_i^k x_j^k\right)^d, \quad \mu_k \geq 0, \quad \sum_{k=1}^{p} (\mu_k)^q \leq \Lambda$$

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

Feature Selection: Reality Check

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

- Comparison with existing methods:



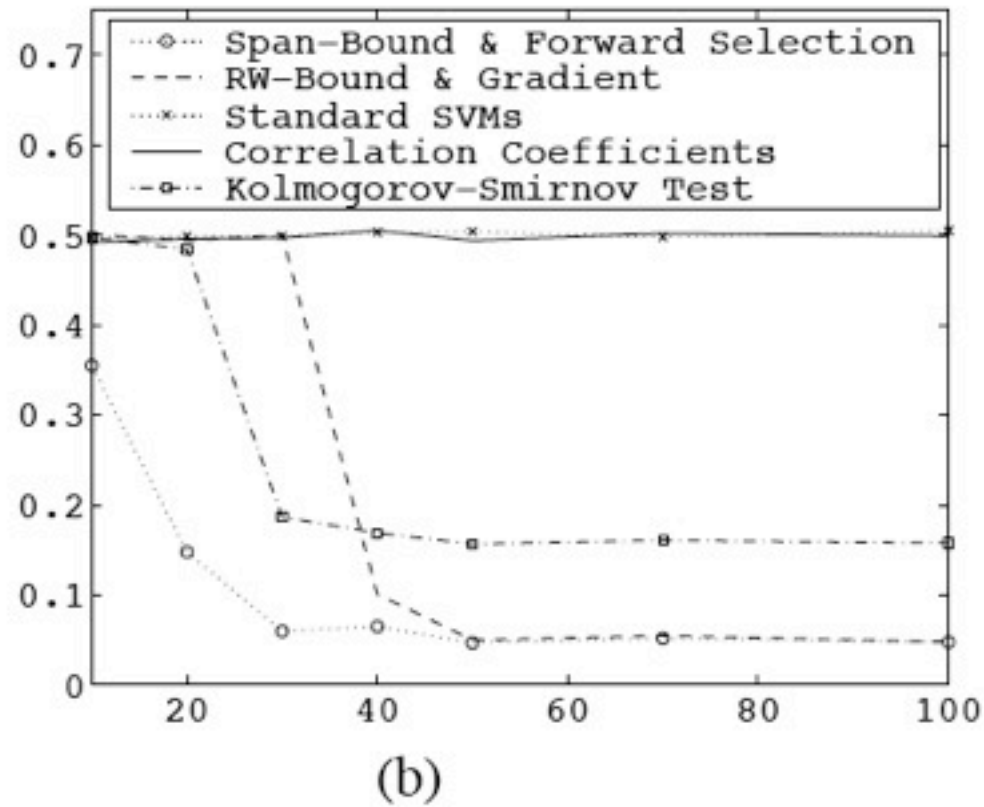


Figure 1: A comparison of feature selection methods on (a) a linear problem and (b) a nonlinear problem both with many irrelevant features. The x-axis is the number of training points, and the y-axis the test error as a fraction of test points.

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:

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

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

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

General LK Formulation - SVMs

- Consider the maximization problem:

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

- The corresponding Lagrange function is

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

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

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

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

General LK Formulation - SVMs

- The problem can now be rewritten as

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

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

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

General LK Formulation - SVMs

- Optimization problem:

$$\min_{\mathbf{K}\in\overline{\mathcal{K}},t,\boldsymbol{\beta},\boldsymbol{\gamma},\delta} \quad t$$

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

$$(\boldsymbol{\beta}\geq\mathbf{0})\wedge(\boldsymbol{\gamma}\geq\mathbf{0}).$$

- the minimization over $t, \boldsymbol{\beta}, \boldsymbol{\gamma}, \delta$ is a semi-definite program (SDP).
- if $\overline{\mathcal{K}} = \{\mathbf{K}\colon (\mathbf{K}\succeq\mathbf{0})\wedge \mathrm{Tr}[\mathbf{K}] = 1\}$ 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 $(\mathbf{K} + \tau \mathbf{I})$ invertible for $\tau \geq 0$ which in general does not hold. In particular, for $\tau = 0$, $\mathbf{K}$ 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, $\mu \mapsto \mathbf{K}_{\mu}$ concave function.
  - $\mathbf{Y} \in \mathbb{R}^{m \times m}$ diagonal matrix with $\mathbf{Y}_{ii} = \mathbf{y}_i$.
- Optimization problem:

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

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

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

Linear Combinations

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

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

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

- 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:

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

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

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

Non-Negative Combinations

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

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

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

- Hypothesis sets:

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

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):

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

Non-Negative Combinations

(Lanckriet et al., 2004)

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

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

$$\text{subject to: } \forall k \in [1, p], t \geq \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \boldsymbol{\alpha};$$

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

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

N-Neg. Comb. - Primal Formulation

- Optimization problem: equivalent primal.

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

Rank-One Base Kernels

- Optimization problem: reduces to simple QP.

$$\max_{\boldsymbol{\alpha}} \; 2\boldsymbol{\alpha}^\top \mathbf{1} - t^2$$
$$\text{subject to:} \; \forall k \in [1,p], -t \leq \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k \leq t;$$
$$\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0.$$

  - $\mathbf{K}_k = \mathbf{X}_k \mathbf{X}_k^\top$.
  - 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?

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

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

  - where $\mathbf{y}, \mathbf{b}, \mathbf{a}_{\boldsymbol{\alpha}} \in \mathbb{R}^m$, $c_{\boldsymbol{\alpha}} \in \mathbb{R}$, and $\boldsymbol{\alpha} \in \mathcal{A}$, 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):

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

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

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

SILP - Formulation

(Sonnenburg et al., 2006)

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

$$\max_{\boldsymbol{\beta},\theta} \theta$$

$$\text{subject to: } \theta \leq \sum_{k=1}^{p} \beta_k (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{1})$$

$$(\sum_{k=1}^{p} \beta_k = 1) \wedge (\boldsymbol{\beta} \geq \mathbf{0})$$

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

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

$$\sum_{k=1}^{p} \beta_k (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_k \mathbf{Y} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{1}) = \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_{\boldsymbol{\beta}} \mathbf{Y} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{1},$$

    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:

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

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

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

- Reduced gradient:

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

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

Reduced Gradient: SimpleMKL

(Rakotomamonjy et al., 2008)

- SimpleMKL algorithm

Algorithm 1 SimpleMKL algorithm

set $d_m = \frac{1}{M}$ for $m = 1, \ldots, M$
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 $m = 1, \ldots, M$ 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}
    $d = d^\dagger, D = D^\dagger$
    $\nu = \operatorname{argmin}_{\{m | D_m < 0\}} -d_m/D_m, \gamma_{\max} = -d_\nu/D_\nu$
    $d^\dagger = d + \gamma_{\max} D, D^\dagger_\mu = D_\mu - D_\nu, D^\dagger_\nu = 0$
    compute $J^\dagger$ by using an SVM solver with $K = \sum_m d^\dagger_m K_m$
  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:

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

- Approximate F:

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

- Solving for μ:

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

Newton's Method: L_q-Norm

(Kloft et al., 2009)

- Optimization problem:

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

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

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

- 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:

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

- 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:

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

- Approximate F:

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

  - strictly convex function $\Omega(\boldsymbol{\mu})$ and

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

    Bregman divergence defined by $\Omega(\boldsymbol{\mu})$.

Mirror Descent

- Solving $\nabla_{\boldsymbol{\mu}} G_t(\boldsymbol{\mu}) = 0$

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

  and the next value of $\boldsymbol{\mu}$ given by

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

coordinate function inversion.

- Examples

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

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

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:

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

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

Mirror Descent: Mixed-Norm MKL

(Nath et al., 2009)

- Update of kernel parameter

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

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

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

- 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:

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

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

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

- Use Mirror Descent algorithm to update w:

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

  - 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:

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

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

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

Projected Gradient, KRR

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

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

$$\text{subject to: } \ \boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_2 \leq \Lambda.$$

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

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

Projected Gradient, KRR - L_2 Reg.

(Cortes et al., 2009)

PROJECTIONBASEDGRADIENTDESCENT$((\mathbf{K}_k)_{k\in[1,p]}, \boldsymbol{\mu}_0)$

$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_0$
$\boldsymbol{\mu}' \leftarrow \infty$
while $\|\boldsymbol{\mu}' - \boldsymbol{\mu}\| > \epsilon$ **do**
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}'$
  $\boldsymbol{\alpha} \leftarrow (\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})^{-1}\mathbf{y}$
  $\boldsymbol{\mu}' \leftarrow \boldsymbol{\mu} + \eta\,(\boldsymbol{\alpha}^\top \mathbf{K}_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top \mathbf{K}_p \boldsymbol{\alpha})^\top$
  for $k \leftarrow 1$ **to** p **do**
    $\boldsymbol{\mu}'_k \leftarrow \max(0, \boldsymbol{\mu}'_k)$
  $\boldsymbol{\mu}' \leftarrow \boldsymbol{\mu}_0 + \Lambda \frac{\boldsymbol{\mu}' - \boldsymbol{\mu}_0}{\|\boldsymbol{\mu}' - \boldsymbol{\mu}_0\|}$
return $\boldsymbol{\mu}'$

Interpolated Step, KRR - L_2 Reg.

(Cortes et al., 2009)

$\textsc{InterpolatedIterativeAlgorithm}((\mathbf{K}_k)_{k\in[1,p]}, \boldsymbol{\mu}_0)$

$\boldsymbol{\alpha} \leftarrow \infty$

$\boldsymbol{\alpha}' \leftarrow (\mathbf{K}_{\boldsymbol{\mu}_0} + \lambda \mathbf{I})^{-1}\mathbf{y}$

while $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| > \epsilon$ **do**

  $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}'$

  $\mathbf{v} \leftarrow (\boldsymbol{\alpha}^\top \mathbf{K}_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top \mathbf{K}_p \boldsymbol{\alpha})^\top$

  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_0 + \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|}$

  $\boldsymbol{\alpha}' \leftarrow \eta\boldsymbol{\alpha} + (1-\eta)(\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})^{-1}\mathbf{y}$

return $\boldsymbol{\alpha}'$

Simple and very efficient: few iterations (less than15).

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

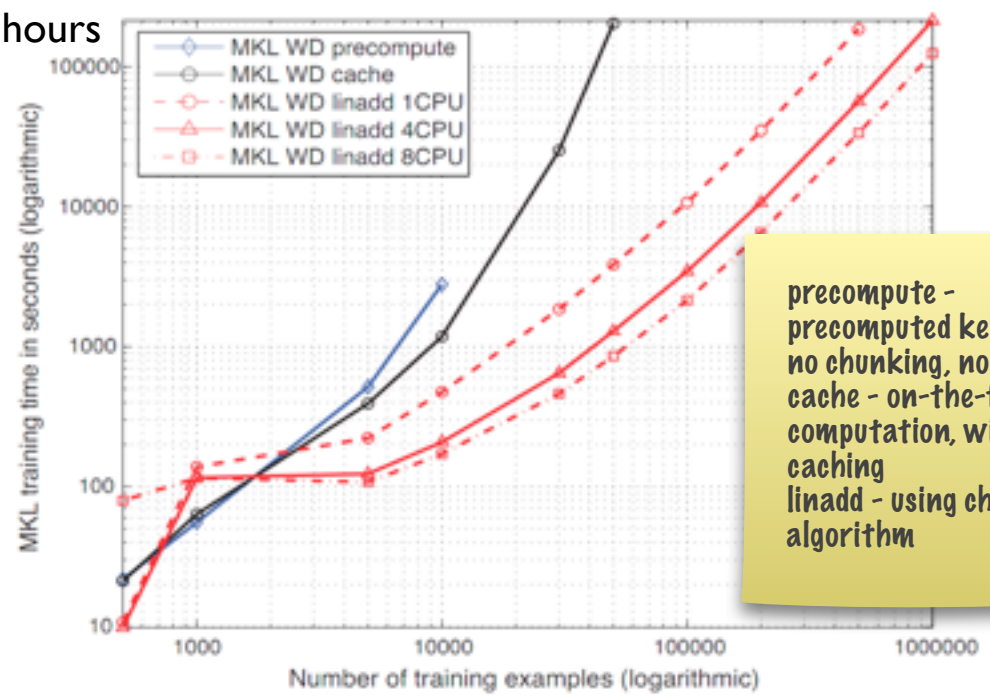
- Solving the same problem.
  - only difference is the norm of the regularization.
- Compare speed for different norms.
- Compare accuracy for different norms.

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.

~28 hours



precompute - precomputed kernels, no chunking, no par cache - on-the-fly computation, with caching
linadd - using chunking algorithm

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 $\ell = 146$ $M = 793$

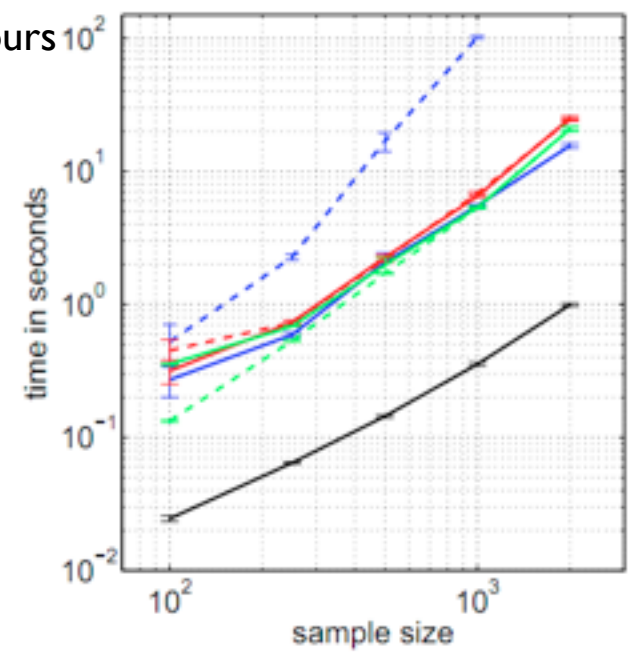
Algorithm	# Kernel	Accuracy	Time (s)	# SVM eval	# Gradient eval
SILP	33.5 ± 3.8	80.5 ± 5.1	2290 ± 864	903 ± 187	903 ± 187
SimpleMKL	36.7 ± 5.1	80.6 ± 5.1	163 ± 93	2770 ± 1560	115 ± 66
Grad. Desc.	35.7 ± 3.9	80.2 ± 4.7	469 ± 90	7630 ± 2600	836 ± 99

Efficient L_p Regularized

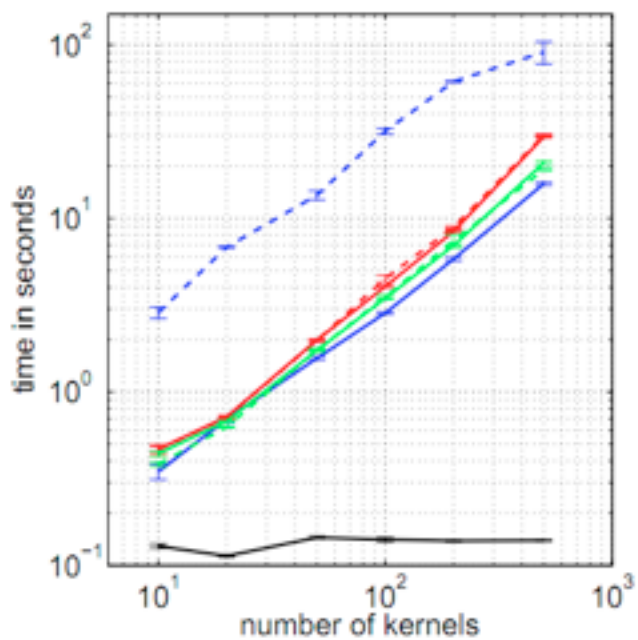
(Kloft et al., 2009)

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

~2.8 hours

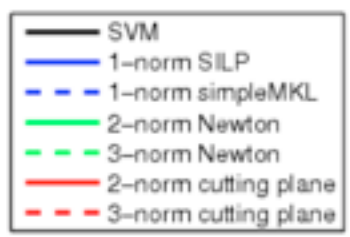


50 kernels



500 samples

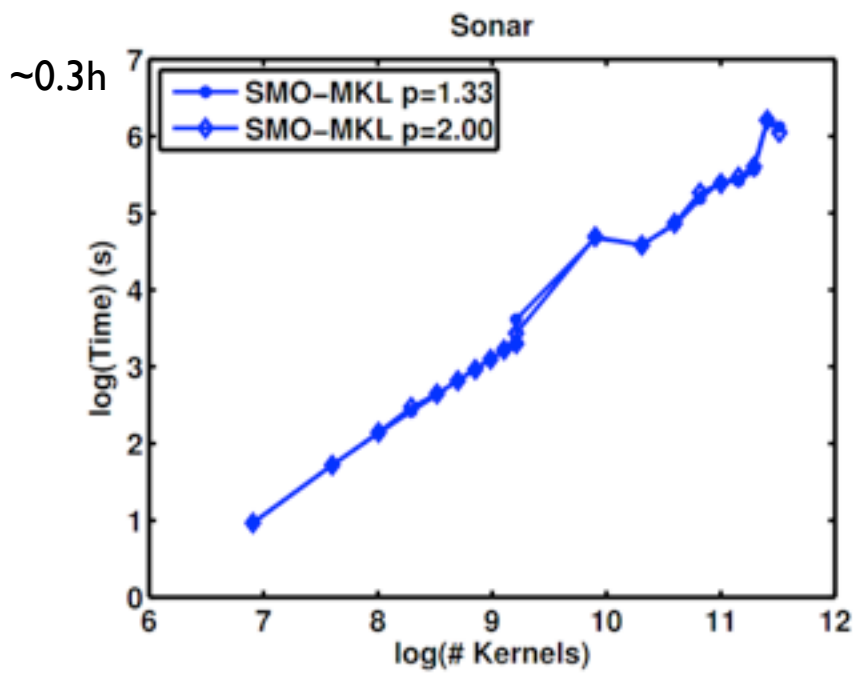
MNIST dataset,
Gaussian kernels



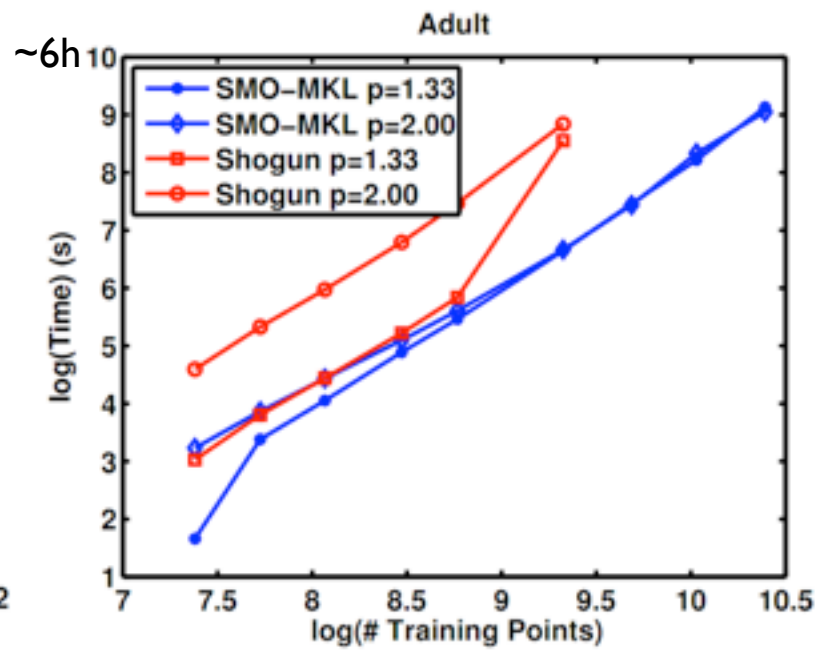
SMO Optimization

(Vishwanathan et al., 2009)

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



166 points

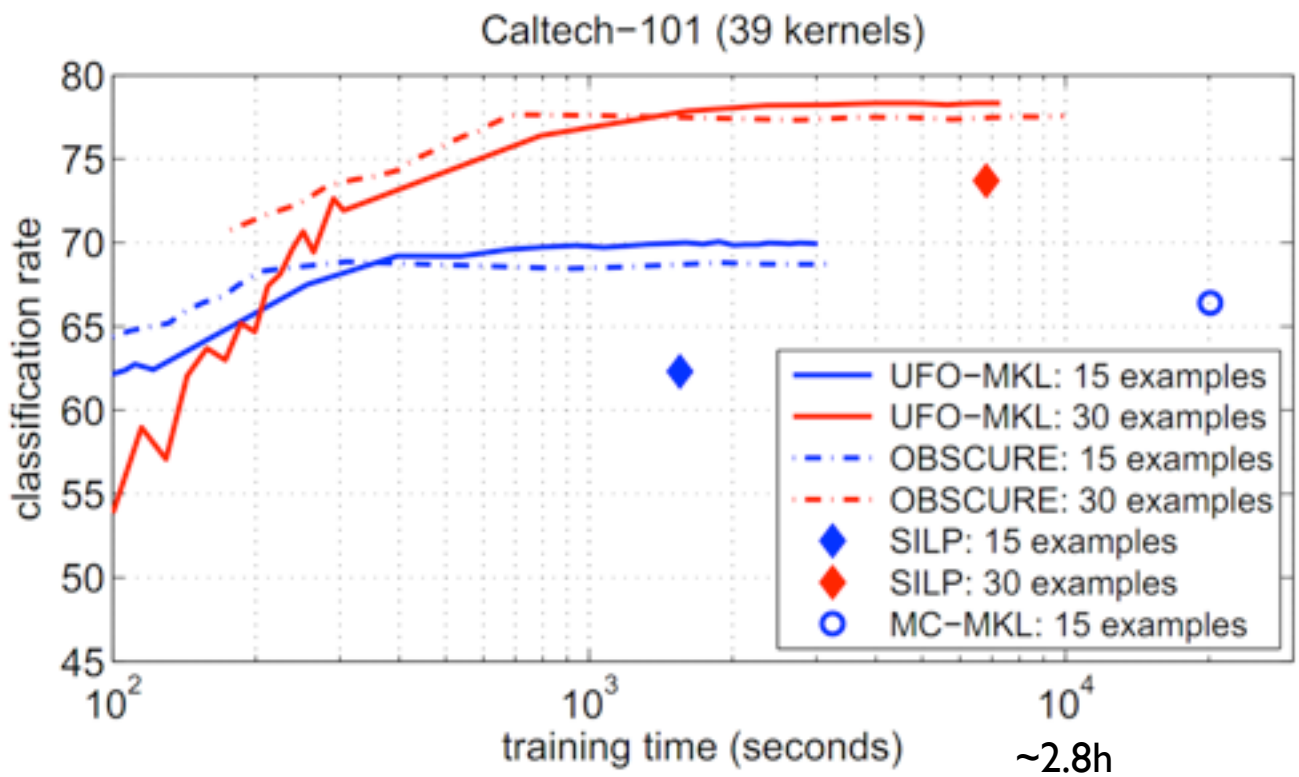


50 kernels

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)

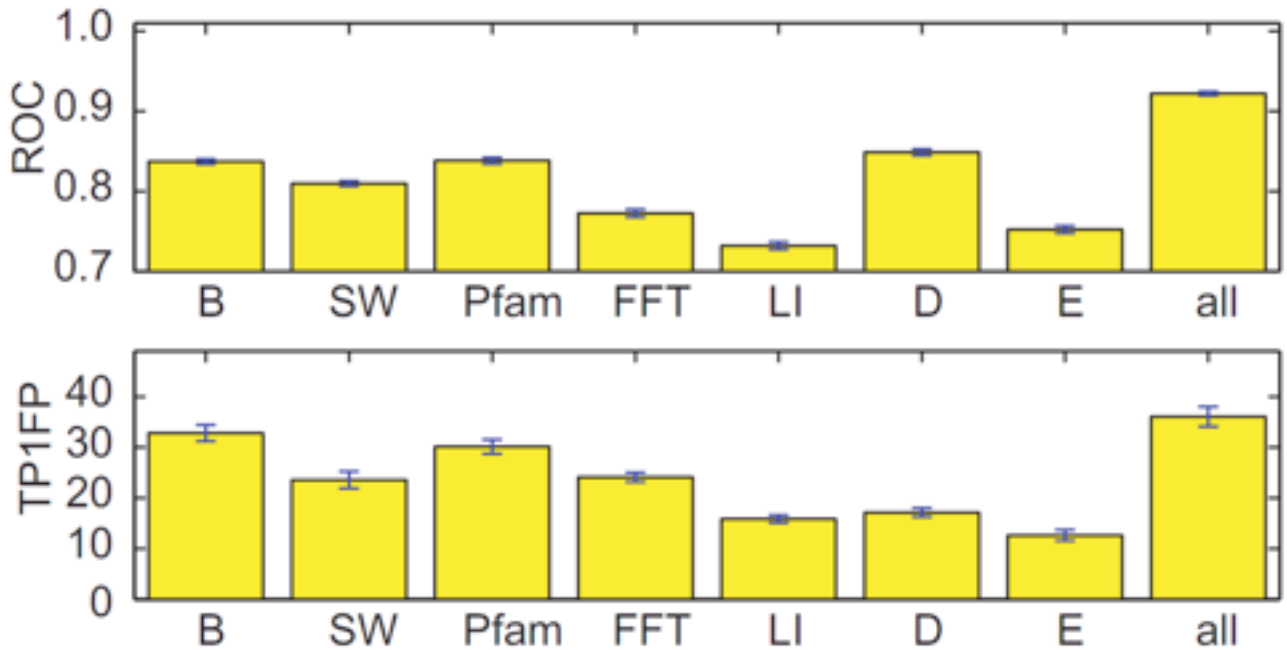
- Learn with sparse linear combinations of kernels.
- Combining kernels can help performance, but do simple uniform combinations suffice?

	$\mu_{1,+}$	$\mu_{2,+}$	$\mu_{3,+}$	$\mu_{4,+}$	$\mu_{5,+}$	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.

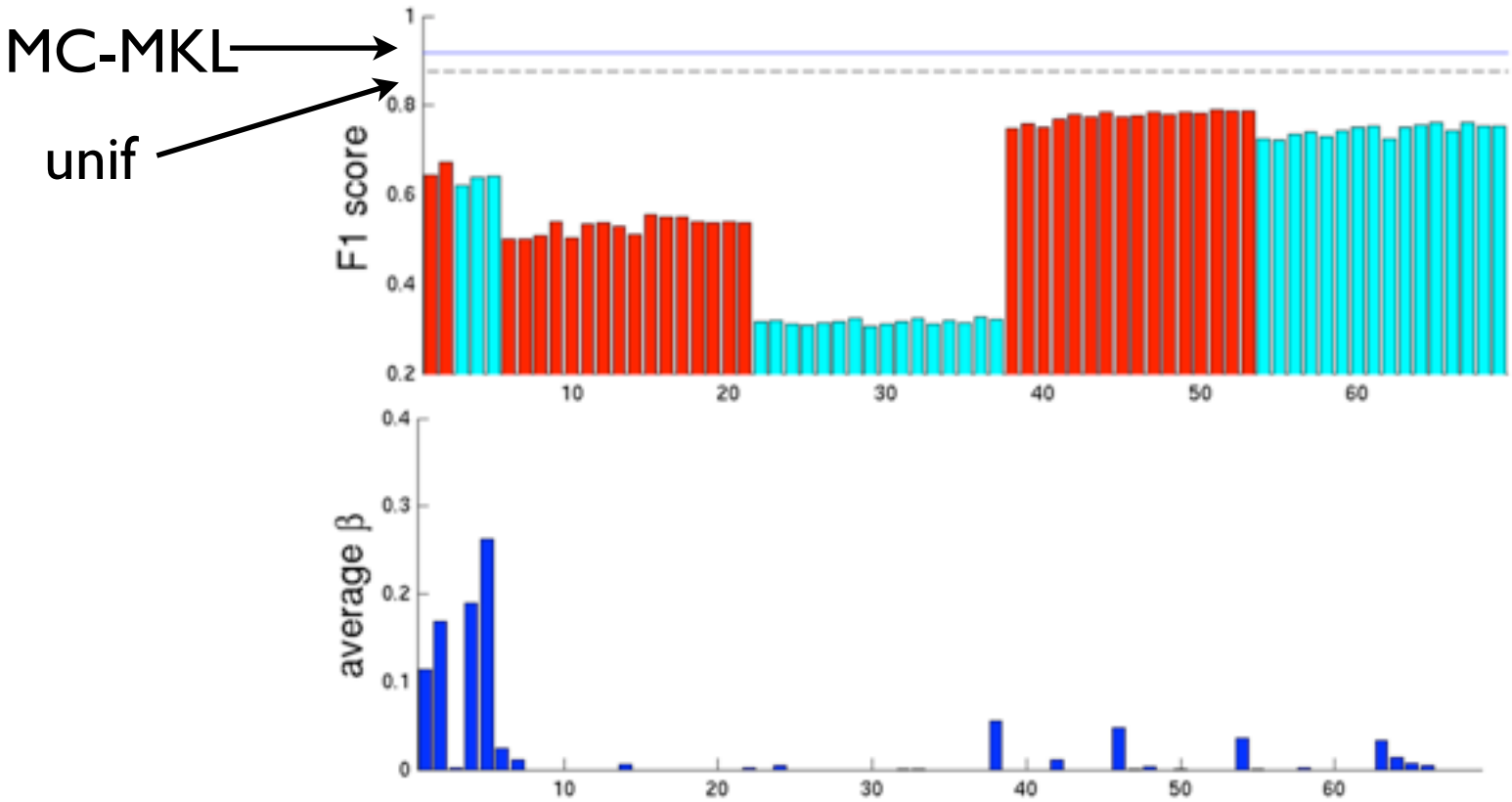


K_B	K_{SW}	K_D	K_E	K_{R1}	K_{R2}	K_{R3}	K_{R4}	TP1FP (%)	ROC
1.81	1.05	0.73	0.42	–	–	–	–	35.71 ± 2.13	0.9196 ± 0.0023
3.30	1.98	1.31	0.79	0.08	0.17	0.21	0.17	34.14 ± 2.09	0.9145 ± 0.0026
1.00	1.00	1.00	1.00	–	–	–	–	33.87 ± 2.20	0.9180 ± 0.0026
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	26.24 ± 1.39	0.8627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

Multi-Class L_1-Regularized

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

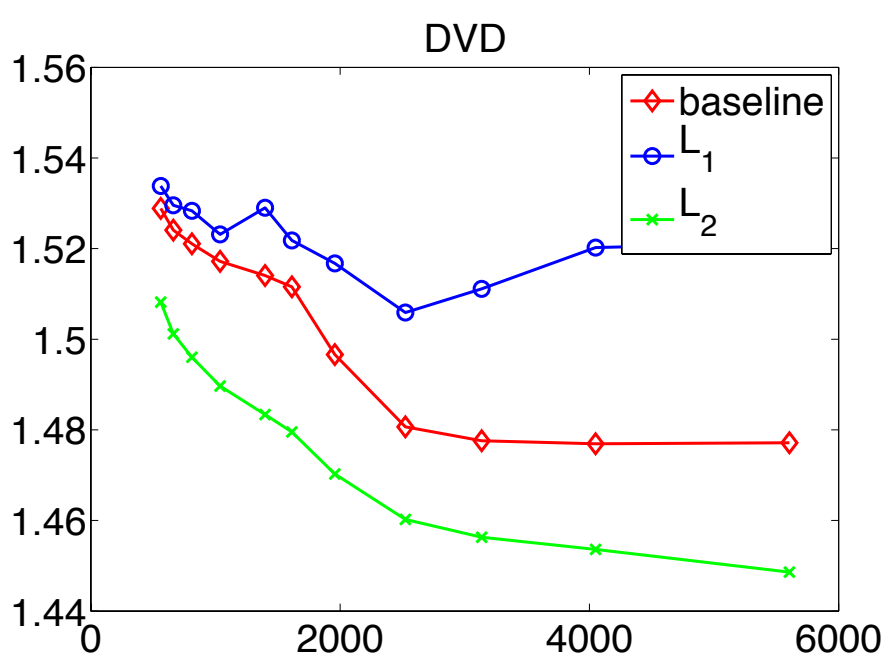
- Predict subcellular localization (TargetP dataset), 5 classes, 69 base kernels.
- Multi-class SVM with L_1-regularization.

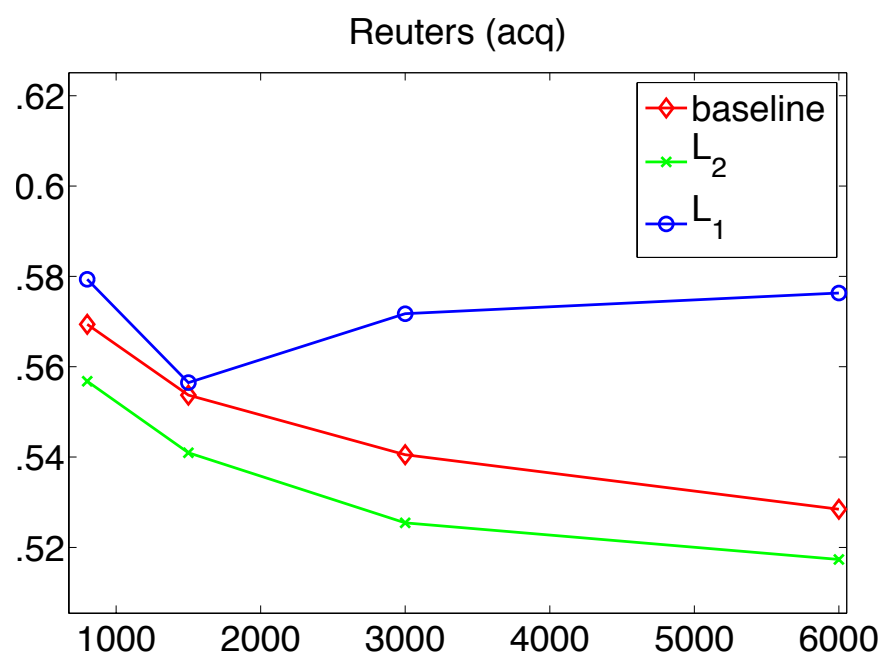


L_2-Regularized Combinations

(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.

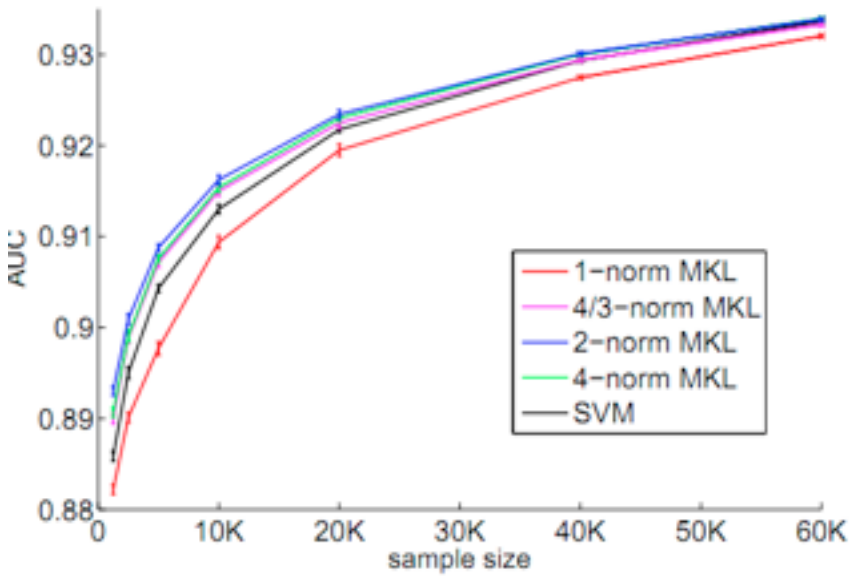


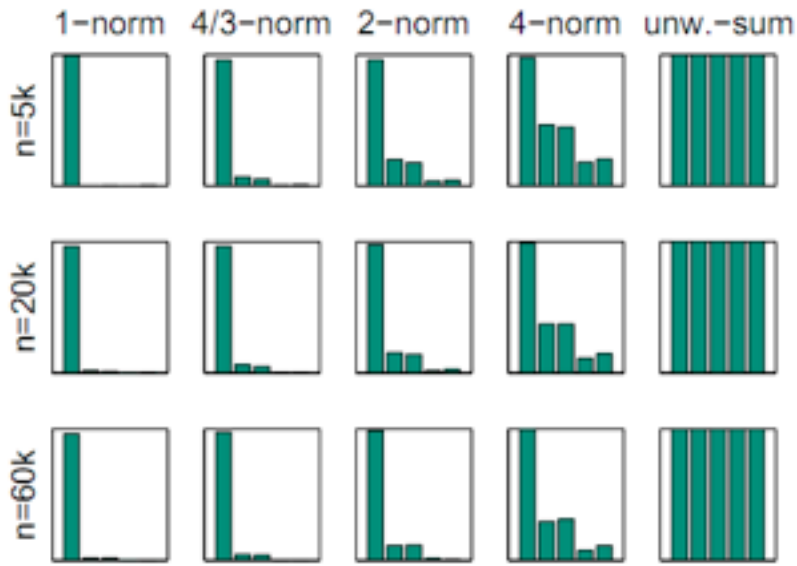


L_p-Regularized Combinations

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

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

- 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 $= 89.9 \pm 2.5$, Uniform GMKL $= 94.6 \pm 2.0$

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	$\mathbf{90.9 \pm 1.9}$
10	–	87.6 ± 2.2	90.6 ± 3.4	90.2 ± 3.5	90.2 ± 2.6	87.8 ± 2.4	$\mathbf{93.7 \pm 2.1}$
15	–	89.1 ± 1.9	93.0 ± 2.1	91.9 ± 2.0	92.6 ± 3.0	87.7 ± 2.2	$\mathbf{94.1 \pm 2.1}$
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	$\mathbf{94.6 \pm 2.0}$	–	–
	75.1 (9.8)	89.2 (25.2)	92.6 (34.0)	92.9 (34.0)	–	88.1 (29.3)	94.4 (16.9)

MKL + l_1-reg: $\mathbf{K}_{\boldsymbol{\mu}}(x_i, x_j) = \sum_{k=1}^{p} \mu_k \exp\left(-\gamma_k (x_{ik} - x_{jk})^2\right)$

GMKL + l_1-reg: $\mathbf{K}_{\boldsymbol{\mu}}(x_i, x_j) = \prod_{k=1}^{p} \exp\left(-\mu_k (x_{ik} - x_{jk})^2\right)$

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**

HKL + l_1-reg: $\mathbf{K}_{\boldsymbol{\mu},4}(x_i, x_j) = \prod_{k=1}^{p} \left(1 + \mu_k x_{ik} x_{jk}\right)^4$

GMKL + l_1-reg: $\mathbf{K}_{\boldsymbol{\mu},2}(x_i, x_j) = \left(1 + \sum_{k=1}^{p} \mu_k x_{ik} x_{jk}\right)^2$

Polynomial Kernels - KRR

(Cortes et al., 2010)

- $p \geq 1$ base PDS kernel functions $K_1, \ldots, K_p$.
- 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 $d=2$.

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

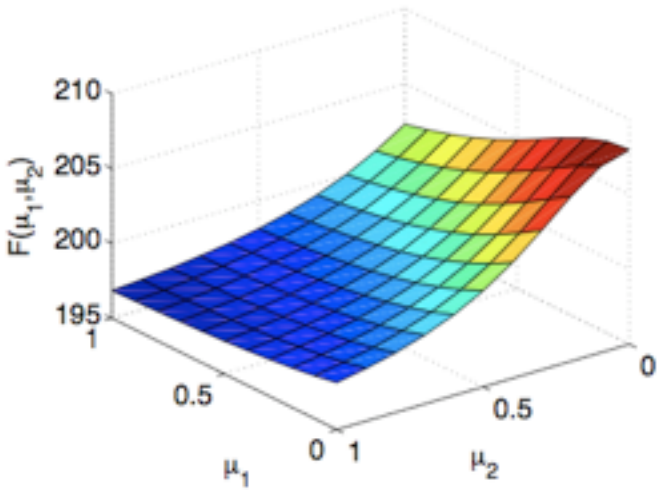
$$\text{subject to: } \boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_q \leq \Lambda.$$

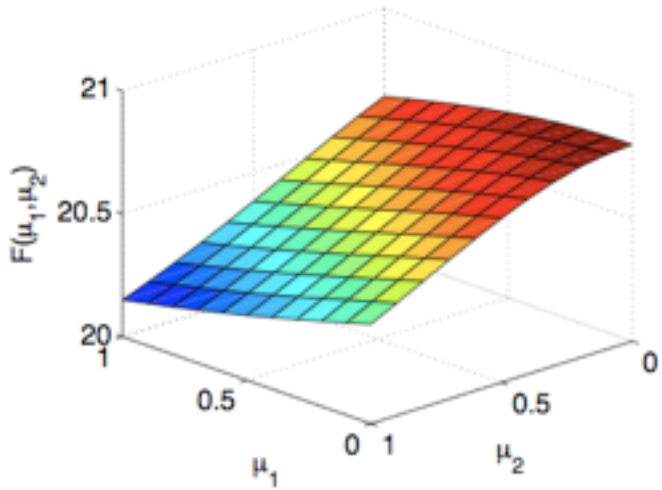
- Closed-form solution $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}} + \lambda \mathbf{I})^{-1} \mathbf{y}$ leads to:

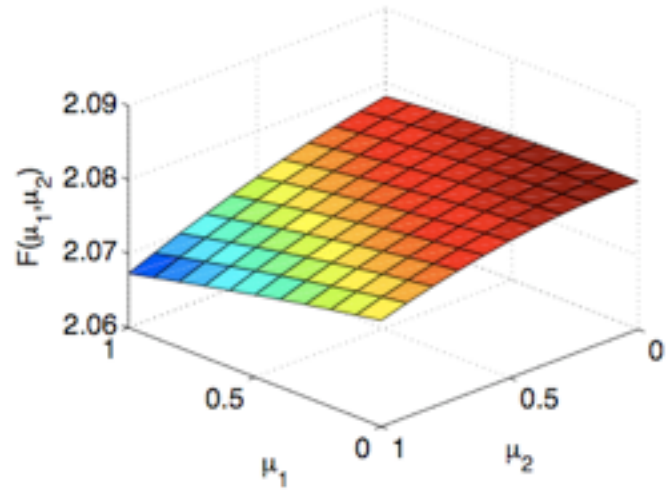
$$\min_{\boldsymbol{\mu}} \; F(\boldsymbol{\mu}) = \mathbf{y}^\top \Big(\sum_{k,l=1}^{p} \mu_k \mu_k \mathbf{K}_k \circ \mathbf{K}_l + \lambda \mathbf{I} \Big)^{-1} \mathbf{y}$$

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

Function Properties







- Two properties:
  - decreasing.
  - no interior stationary points

$$\forall \boldsymbol{\mu}, \nabla F(\boldsymbol{\mu}) \leq \mathbf{0}$$
$$\forall \boldsymbol{\mu} > \mathbf{0}, \nabla F(\boldsymbol{\mu}) \neq \mathbf{0}$$

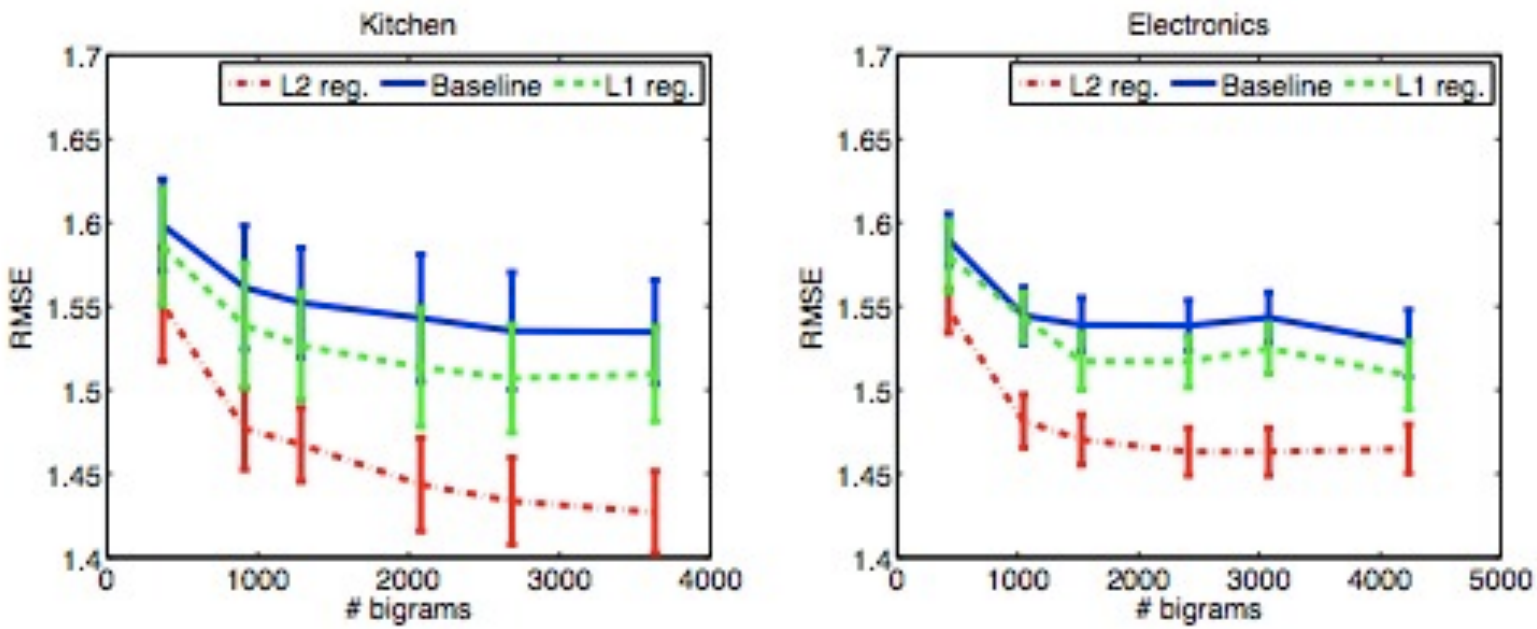
→ optimal solution at the boundary.

- 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:

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

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

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

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: $R^2(K) = \max_{\beta_i} \sum_{i=1}^{m} \beta_i K(x_i, x_i) - \sum_{i,j=1}^{m} \beta_i \beta_j K(x_i, x_j), \text{ s.t. } \beta_i \geq 0, \sum_{i=1}^{m} \beta_i = 1$
- RKL optimization

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

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

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

RKL: Reality Check

(Gai et al, 2010)

Index	1		2		3		4		5		6		7		8	
	Unif		MKL		KL-C		Ours		MKL		KL-C		Ours		Ours	
Constraint			L_1		L_1		L_1		L_2		L_2		L_2		No	
Data set	Acc.	Nk	Acc.	Nk	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	**95.7**(0.9)	3.0	95.7(0.9)	3.0
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	**86.5**(2.4)	2.2	86.3(2.5)	3.2
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	64.1(4.2)	8.0	64.3(4.3)	6.6
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	**100** (0.0)	1.0	100 (0.0)	1.6
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	**84.4**(5.9)	5.4	84.8(5.0)	5.8
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	**73.9**(1.2)	6.0	73.9(1.8)	5.8
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	**93.5**(2.2)	3.8	93.3(2.3)	3.8
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	**97.6**(1.9)	5.8	97.6(1.9)	5.8
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	**76.5**(2.9)	15	76.5(2.9)	15
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	**86.0**(2.6)	2.6	86.0(3.3)	3.0
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	**84.2**(4.2)	5.6	84.2(4.2)	7.6

'KL-C' is (Chapelle et al. 2002).

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

Change of reg. changes relationship to radius.

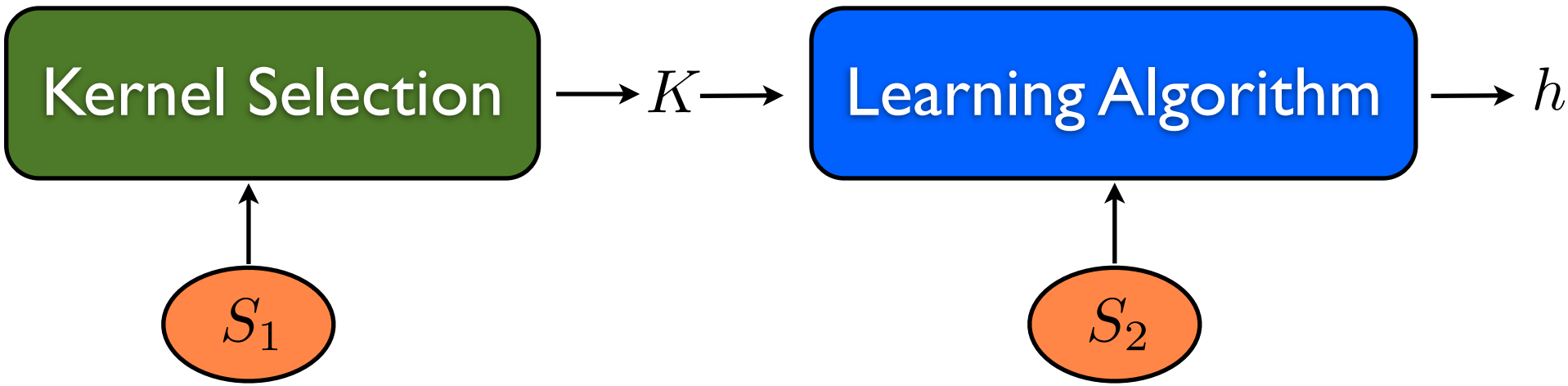
This Part

- 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 (Cristiannini et al., 2001).

Centered Alignment

(Cortes et al., 2010)

- Definition:

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

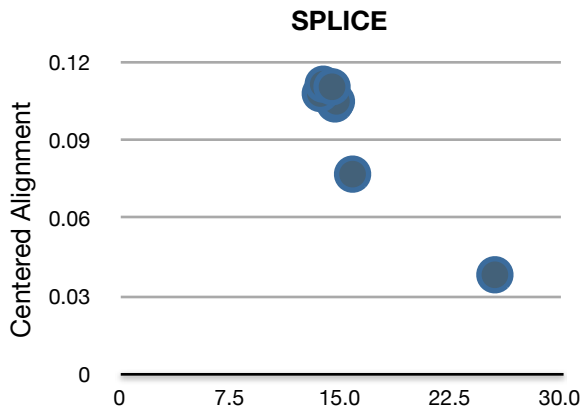
with $K_c(x, x') = (\Phi(x) - \mathbf{E}_x[\Phi])^\top (\Phi(x') - \mathbf{E}_{x'}[\Phi])$.

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

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

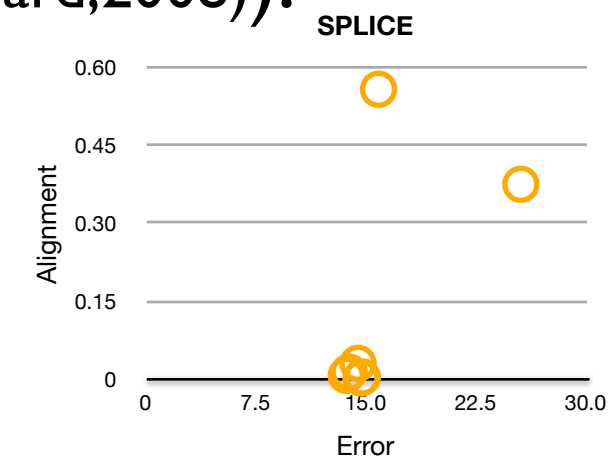
Centering

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

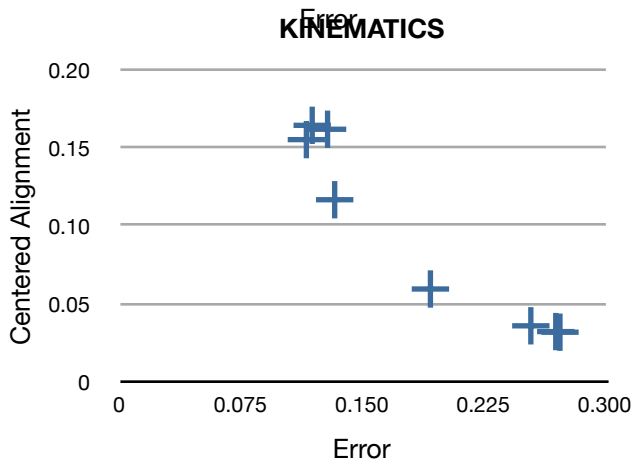


correlation:

-0.95

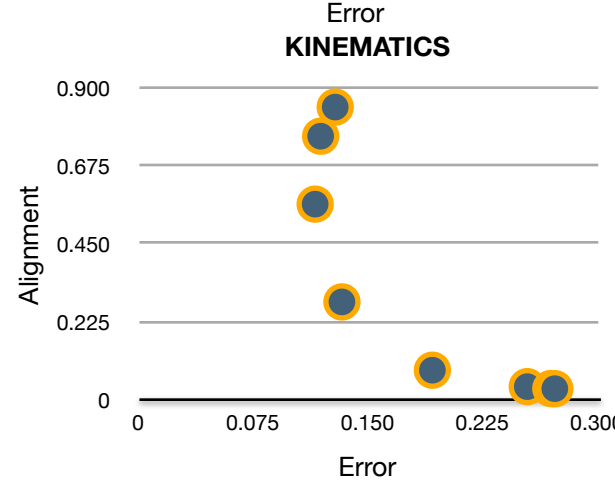


0.45



-0.96

-0.86



Centered

Un-centered

Notes

- (Cortes et al., 2010) comment on (Cristianini, Shawe-Taylor, Elisseef, 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,

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

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,

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

→ 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:

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

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

- Reduces to simple QP: $\boldsymbol{\mu}^* = \dfrac{\mathbf{v}^*}{\|\mathbf{v}^*\|}$,

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

$$\mathbf{a} = \left(\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \ldots, \langle \mathbf{K}_{pc}, \mathbf{y}\mathbf{y}^\top \rangle_F\right)^\top, \quad \mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F.$$

Alternative Algorithm

- Based on independent base kernel alignments:

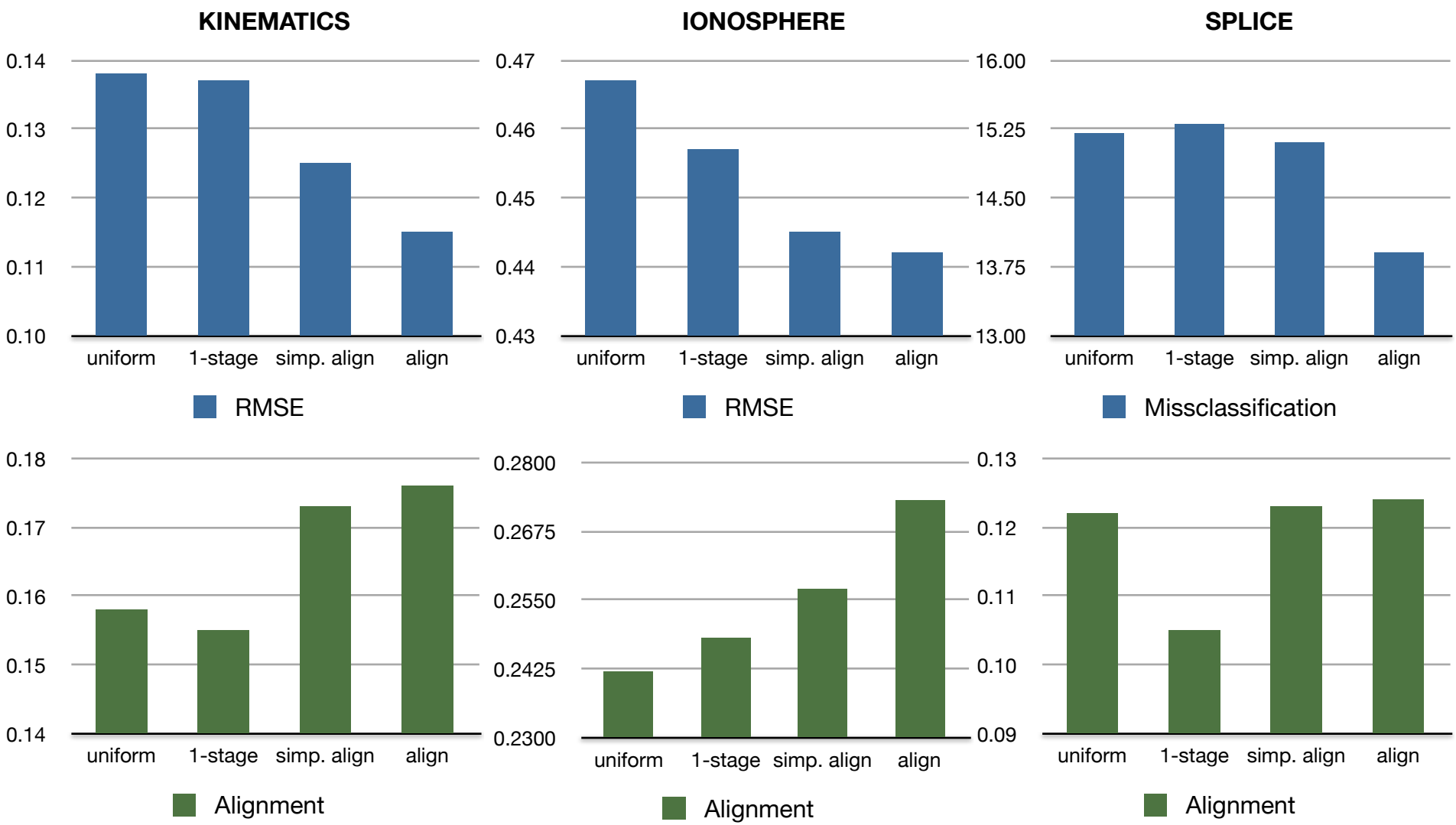
$$\mathbf{K}_{\boldsymbol{\mu}} \propto \sum_{k=1}^{p} \widehat{\rho}(\mathbf{K}_k, \mathbf{K}_Y)\mathbf{K}_k.$$

- Easily scales to very large numbers of kernels.

Centered Alignment: Reality Check

(Cortes et al., 2010)

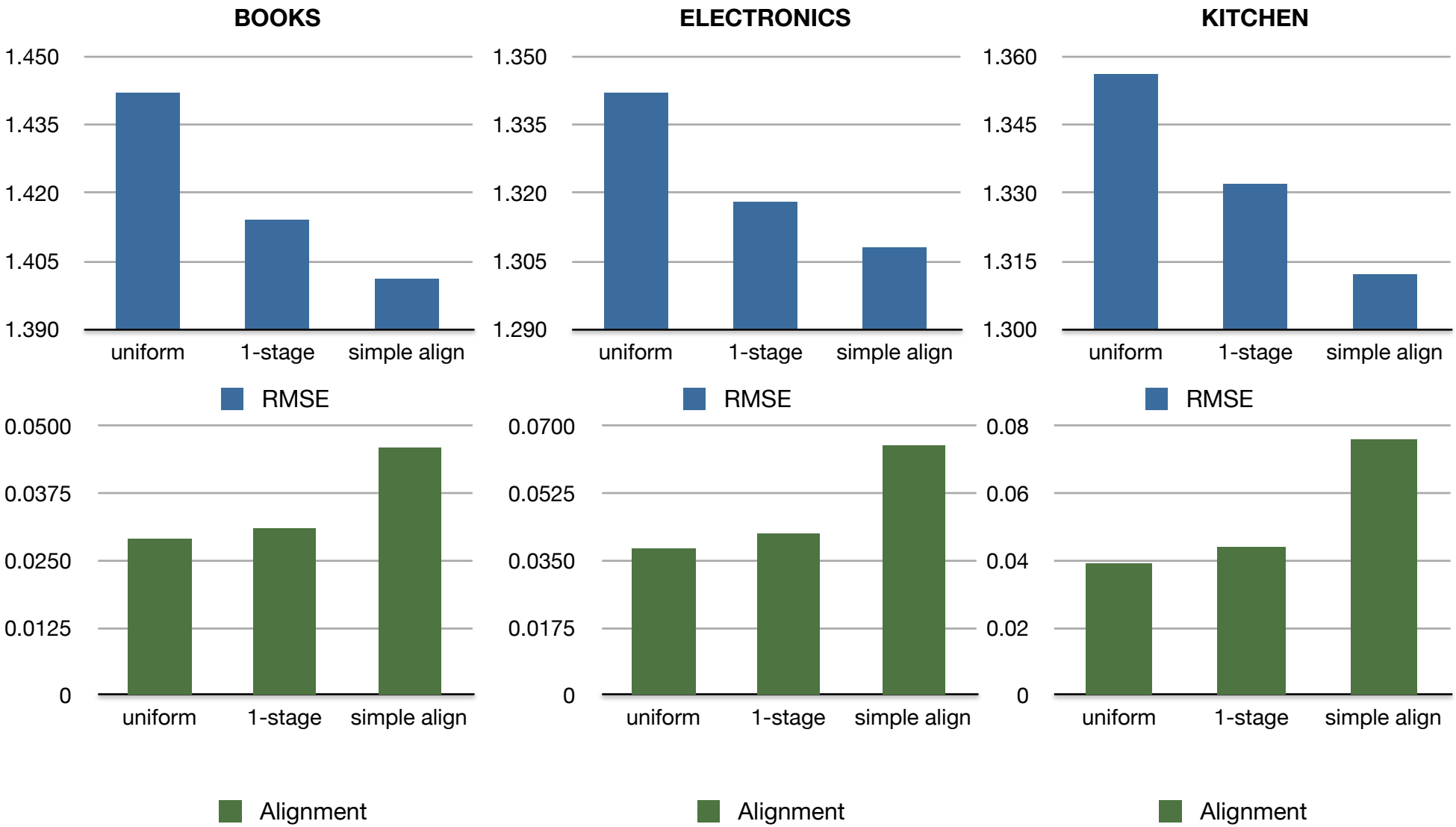
Gaussian base kernels with varying bandwidth.



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?

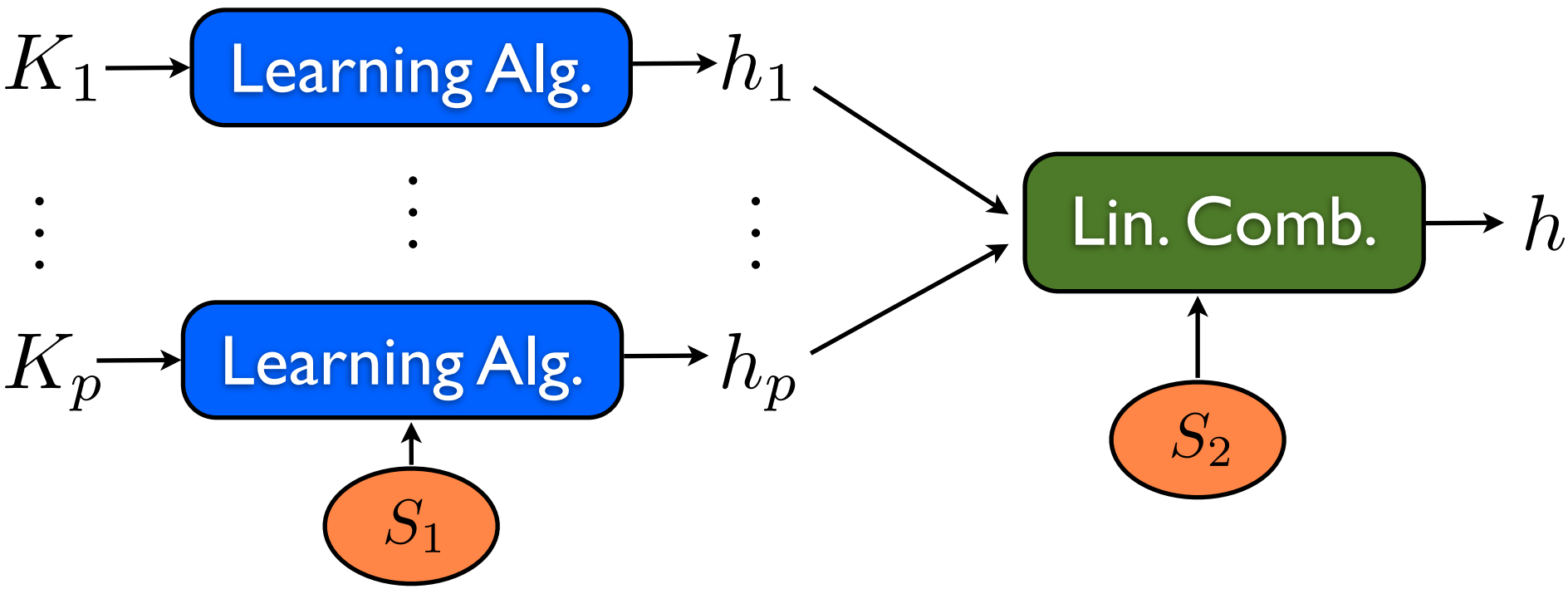
This Part

- 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:

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

- 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!

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

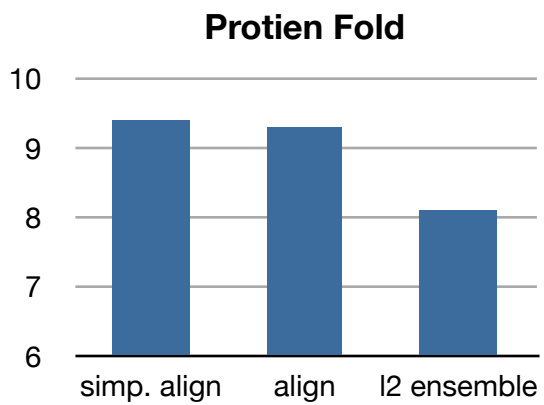
Differs by $p^{1/(2r)}$ factor.

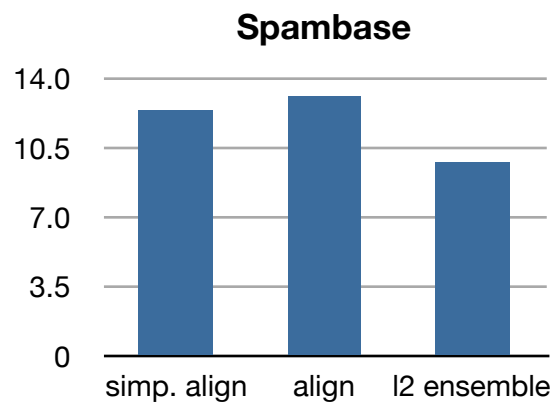
Ensemble Comb.: Reality Check

(Cortes et al., 2011)

Gaussian base kernels

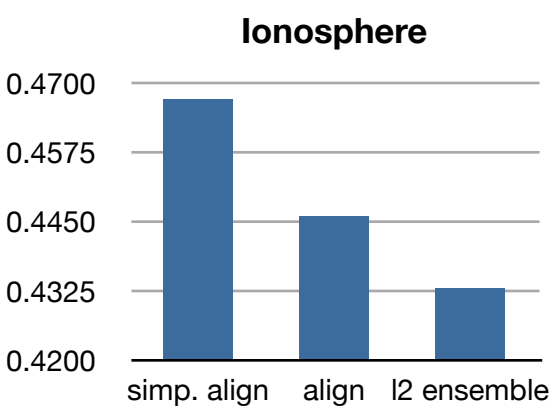
misclassifcation

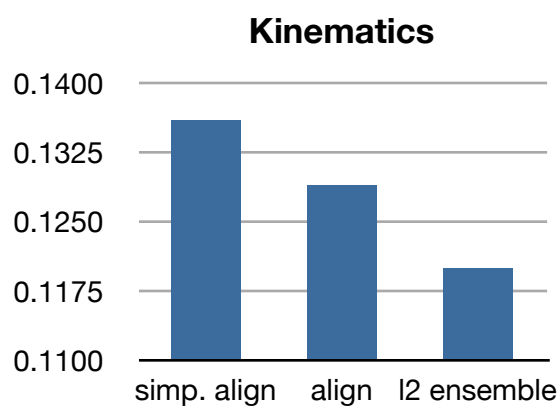




stage 1: SVM
stage 2: L_2-reg SVM

RMSE





stage 1: KRR
stage 2: KRR

One-Stage Ensemble

- Minimize the error of the ensemble hypothesis:

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

- 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:

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

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

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

Multi-Class LPBoost

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

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

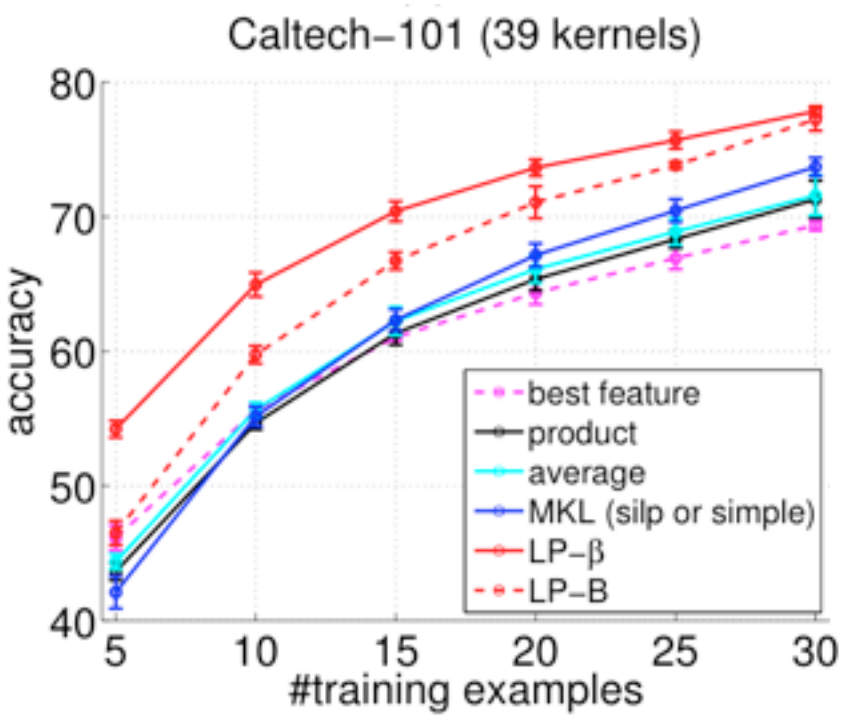
$$\text{s.t.} \sum_{k=1}^{p}\mu_k^{y_i} h_{k,y_i}(x_i) - \operatorname*{argmax}_{y_j \neq y_i} \sum_{k=1}^{p}\mu_k^{y_j} h_{k,y_j}(x_i) + \xi_i \geq \rho$$

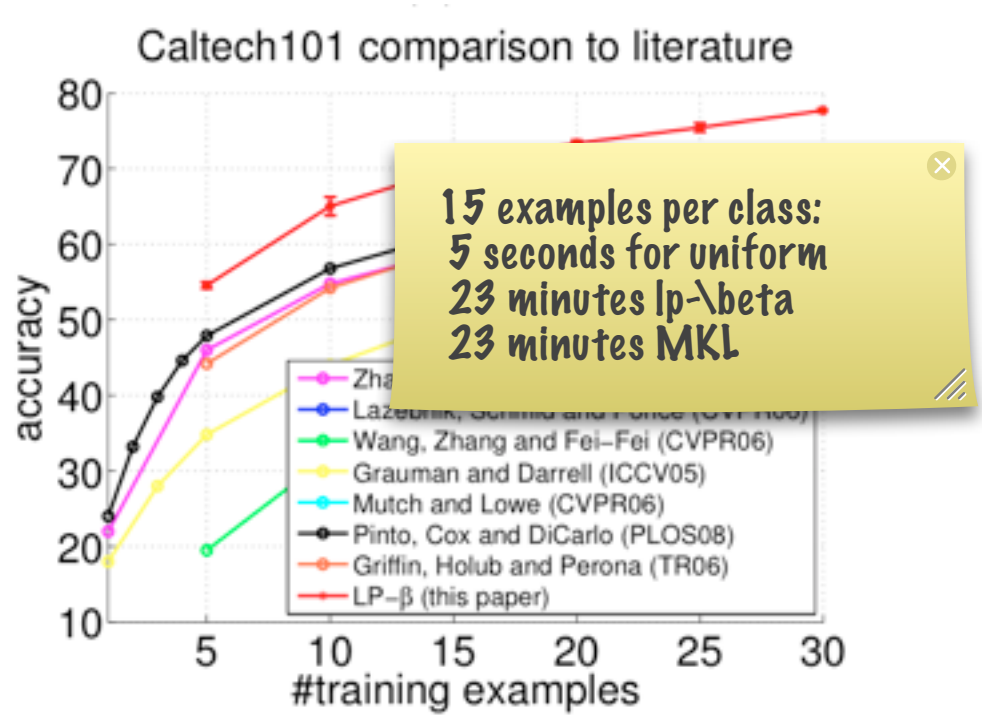
$$\forall c \in [1, C], \quad \sum_{k=1}^{p}\mu_k^c = 1, \mu_k^c \geq 0$$

LP-B 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