Can learning kernels help performance?

Corinna Cortes
Google Research
corinna@google.com

Can learning with kernels help performance?

Corinna Cortes
Google Research
corinna@google.com

Outline

- Learning with kernels, SVM.
- Learning kernels.
- Repeat:

    Discuss new idea

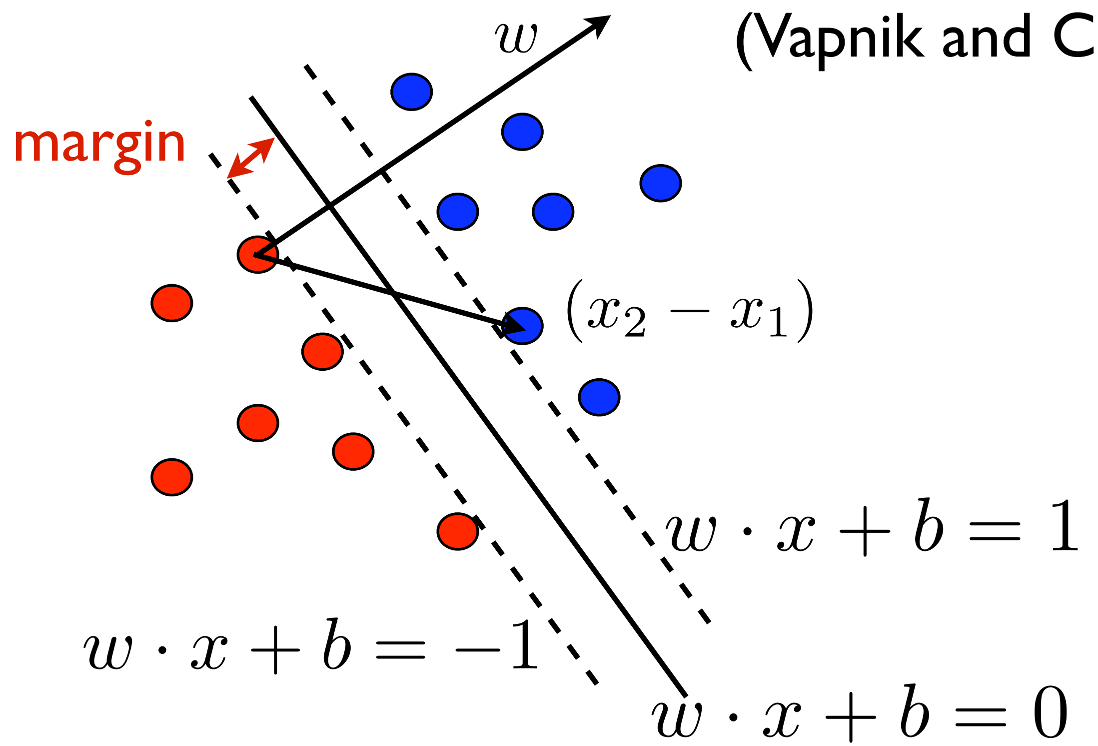
    - convex vs. non-convex optimization,
    - linear vs. non-linear kernel combinations,
    - few vs. many kernels,
    - L_1 vs. L_2 regularization;

    Experimental check;

  Until conclusion.
- Future directions.

Optimal Hyperplane: Max. Margin

(Vapnik and Chervonenkis, 1965)



- Canonical hyperplane: for support vectors,

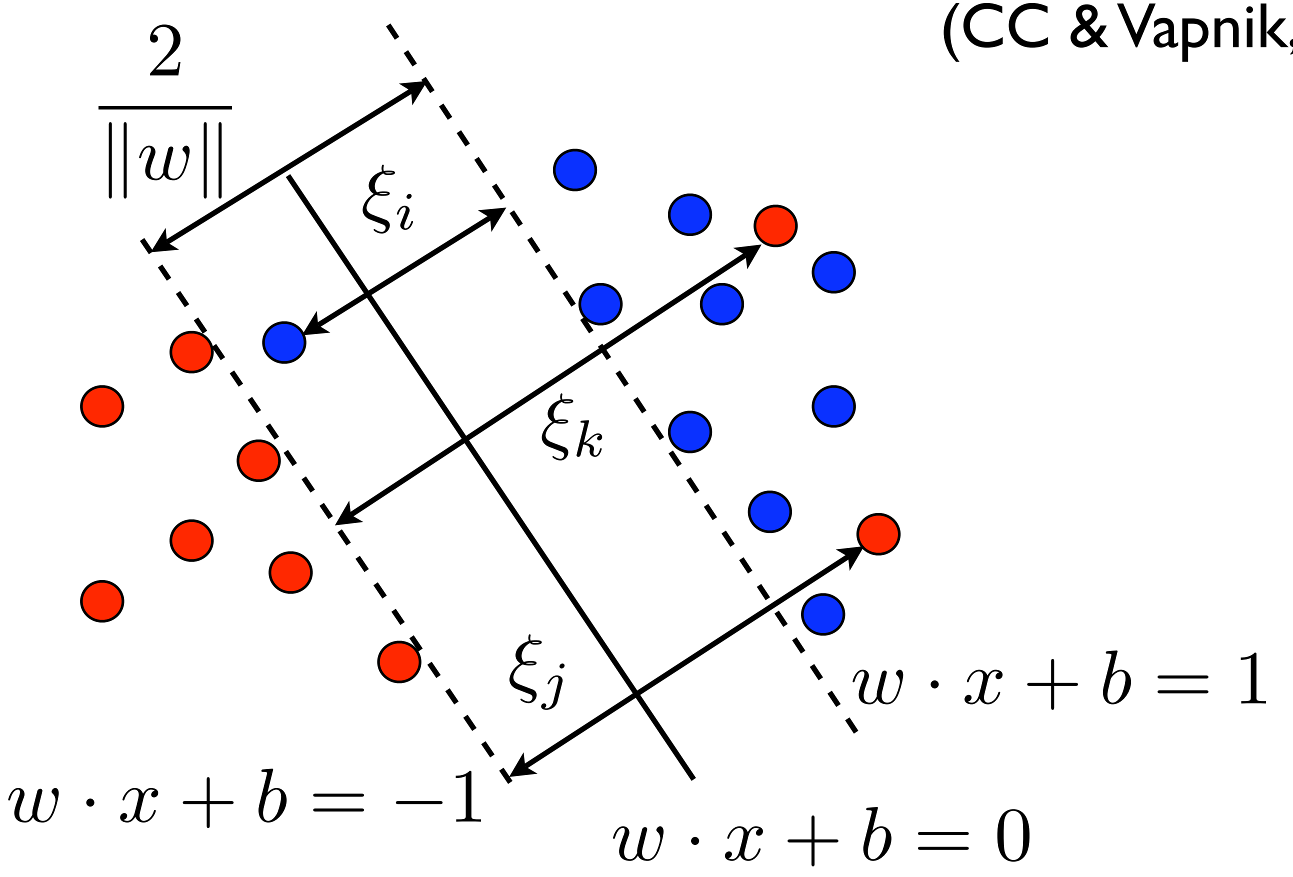
$$w \cdot x + b \in \{-1, +1\}.$$

- Margin: $\rho = 1/||w||$. For points on opposite side,

$$2\rho = \frac{w \cdot (x_2 - x_1)}{||w||} = \frac{2}{||w||}$$

Soft-Margin Hyperplanes

(CC & Vapnik, 1995)



- Support vectors: points along the margin and outliers.

Optimization Problem

- Constrained optimization problem

minimize $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$

subject to $y_i[w \cdot x_i + b] \geq 1 - \xi_i \wedge \xi_i \geq 0, i \in [1, m].$

- Properties
  - C is a non-negative real-valued constant.
  - Convex optimization.
  - Unique solution.

SVMs Equations

- Lagrangian: for all $w, b, \alpha_i \geq 0, \beta_i \geq 0,$

$$L(w, b, \xi, \alpha) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^m \beta_i \xi_i.$$

- KKT conditions:

$$\nabla_w L = w - \sum_{i=1}^m \alpha_i y_i x_i = 0 \iff w = \sum_{i=1}^m \alpha_i y_i x_i.$$

$$\nabla_w b = -\sum_{i=1}^m \alpha_i y_i = 0 \iff \sum_{i=1}^m \alpha_i y_i = 0.$$

$$\nabla_{\xi_i} L = C - \alpha_i - \beta_i = 0 \iff \alpha_i + \beta_i = C.$$

$$\forall i \in [1, m], \ \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i] = 0$$

$$\beta_i \xi_i = 0.$$

Dual Optimization Problem

- Constrained optimization problem

$$\text{maximize} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

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

- Solution

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

$$b = y_i - \sum_{j=1}^{m} \alpha_j y_j (x_j \cdot x_i) \text{ for any SV } x_i \text{ with } \boxed{\alpha_i < C.}$$

SVMs - Kernel Formulation

(Boser, Guyon, and Vapnik, 1992)

- Constrained optimization problem

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

$$\text{subject to } 0 \leq \alpha_i \leq C, i = 1, \ldots, m \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Solution

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

For any support vector such that $0 < \alpha_i < C$,

$$b = y_i - \sum_{j=1}^{m} \alpha_j y_j K(x_i, x_j).$$

Margin Bound

(Bartlett and Shawe-Taylor, 1999)

- Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following holds:

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

fraction of training points with margin less than ρ: $\dfrac{\left|\{x_i : y_i h(x_i) < \rho\}\right|}{m}$.

generalization error.

Kernel Ridge Regression

(Saunders et al., 1998)

- Optimization problem:

$$\max_{\boldsymbol{\alpha}} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \mathbf{y}$$

- Solution:

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

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

Outline

- Learning with kernels, SVM.
- Learning kernels.
- Repeat:

    Discuss new idea

    - convex vs. non-convex optimization,
    - linear vs. non-linear kernel combinations,
    - few vs. many kernels,
    - L_1 vs. L_2 regularization;

    Experimental check;

  Until conclusion.
- Future directions.

Learning the Kernel

- SVM:

$$\max_{\boldsymbol{\alpha}} \quad 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \boldsymbol{\alpha}$$

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

Structural Risk Minimization: select the kernel that minimizes an estimate of the generalization error.

- What estimate should we minimize?

Minimize an Independent Bound

(Chapelle, Vapnik, Bousquet & Mukherjee, 2000)

- Alternate SVM and gradient step algorithm:
  1. maximize the SVM problem over $\boldsymbol{\alpha} \rightarrow \boldsymbol{\alpha}^{\star}$
  2. gradient step over bound on generalization error:
     - margin bound: $T = R^2/\rho^2$
     - span bound: $T = \frac{1}{m}\sum_{i=1}^{m} \Theta(\alpha_i^{\star} S_i^2 - 1)$.

Reality Check

(Chapelle, Vapnik, Bousquet & Mukherjee, 2000)

	Cross-validation	R^2/M^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

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

Kernel Learning & Feature Selection

- Rank-1 kernels

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

- Alternate between solving SVM and gradient step

  - the margin bound: R^2/ρ^2 , (Weston et al., NIPS 2001).

  - the SVM dual: $2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_\mu \mathbf{Y} \boldsymbol{\alpha}$

    (Grandvalet & Canu: NIPS 2002).

Reality Check, Feature Selection

(Chapelle, Vapnik, Bousquet & Mukherjee, 2000)

- Comparison with existing methods:

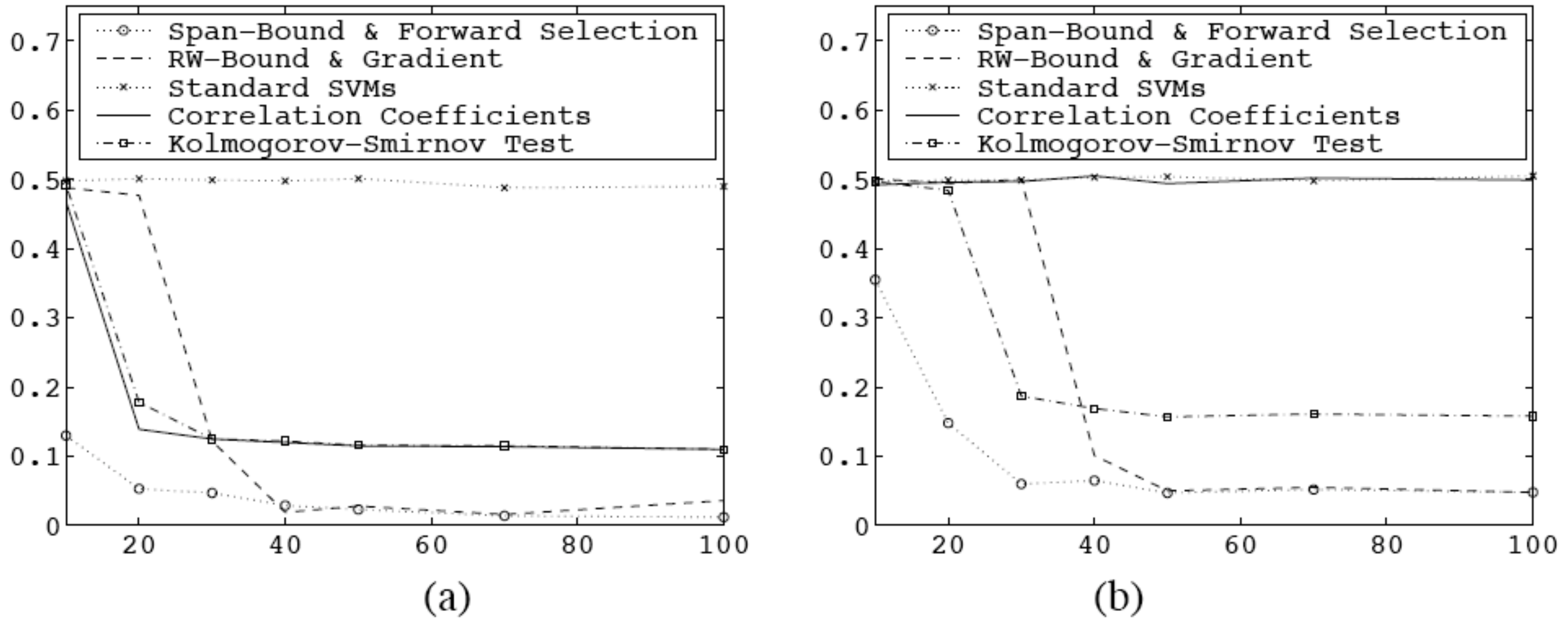


Figure 1: A comparison of feature selection methods on (a) a linear problem and (b) a nonlinear problem both with many irrelevant features. The x-axis is the number of training points, and the y-axis the test error as a fraction of test points.

Kernel Learning Formulation, II

(Lanckriet et al., 2003)

Structural Risk Minimization problem:

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

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

$$\mathcal{K} \succeq 0 \wedge \mathrm{Tr}[\mathbf{K}] \leq \Lambda$$

where $\Lambda > 0$ determines the family of kernels.

SVM - Linear Kernel Expansion

QCQP problem: (Lanckriet et al., 2003)

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \left(\sum_{k=1}^{p} \mu_k \mathbf{K}_k \right) \mathbf{Y} \boldsymbol{\alpha}$$

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

$$\boldsymbol{\mu} \geq \mathbf{0} \wedge \sum_{k=1}^{p} \mu_k \operatorname{Tr}(\mathbf{K}_k) \leq \Lambda.$$

L1 regularization

Computational Complexity

- In general: SDP;
- Non-negative linear combinations: QCQP, SILP (SVM-wrapper solution);
- Rank-1 kernels: QP.

Reality Check

(Lanckriet et al., 2003)

		K_1	K_2	K_3	$\sum_i \mu_i^* K_i$	$\sum_i \mu_{i,+}^* K_i$	best c/v RBF
Heart		$d = 2$	$\sigma = 0.5$				
HM	γ	0.0369	0.1221	-	0.1531	0.1528	
	TSA	**72.9 %**	**59.5 %**	-	**84.8 %**	**84.6 %**	**77.7 %**
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	**-0.09/2.68/0.41**	**0.01/2.60/0.39**	
SM1	ω_{S1}^*	58.169	33.536	74.302	21.361	21.446	
	TSA	**79.3 %**	**59.5 %**	**84.3 %**	**84.8 %**	**84.6 %**	**83.9 %**
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	**-0.09/2.68/0.41**	**0.01/2.60/0.39**	
SM2	ω_{S2}^*	32.726	25.386	45.891	15.988	16.034	
	TSA	**78.1 %**	**59.0 %**	**84.3 %**	**84.8 %**	**84.6 %**	**83.2 %**
	C	1	1	1	1	1	
	$\mu_1/\mu_2/\mu_3$	3/0/0	0/3/0	0/0/3	**-0.08/2.54/0.54**	**0.01/2.47/0.53**	
SM2,C	ω_{S2}^*	19.643	25.153	16.004		15.985	
	TSA	**81.3 %**	**59.6 %**	**84.7 %**		**84.6 %**	**83.2 %**
	C	0.3378	1.18e+7	0.2880		0.4365	
	$\mu_1/\mu_2/\mu_3$	1.04/0/0	0/3.99/0	0/0/0.53		**0.01/0.80/0.53**	

Other Redeeming Properties

- Speed;
- Ranking properties;
- Feature selection, model understanding.

Reality Check

(Lanckriet, De Bie, Cristianini, Jordan, & Noble, 2004)

- Classification performance on the cytoplasmic ribosomal class

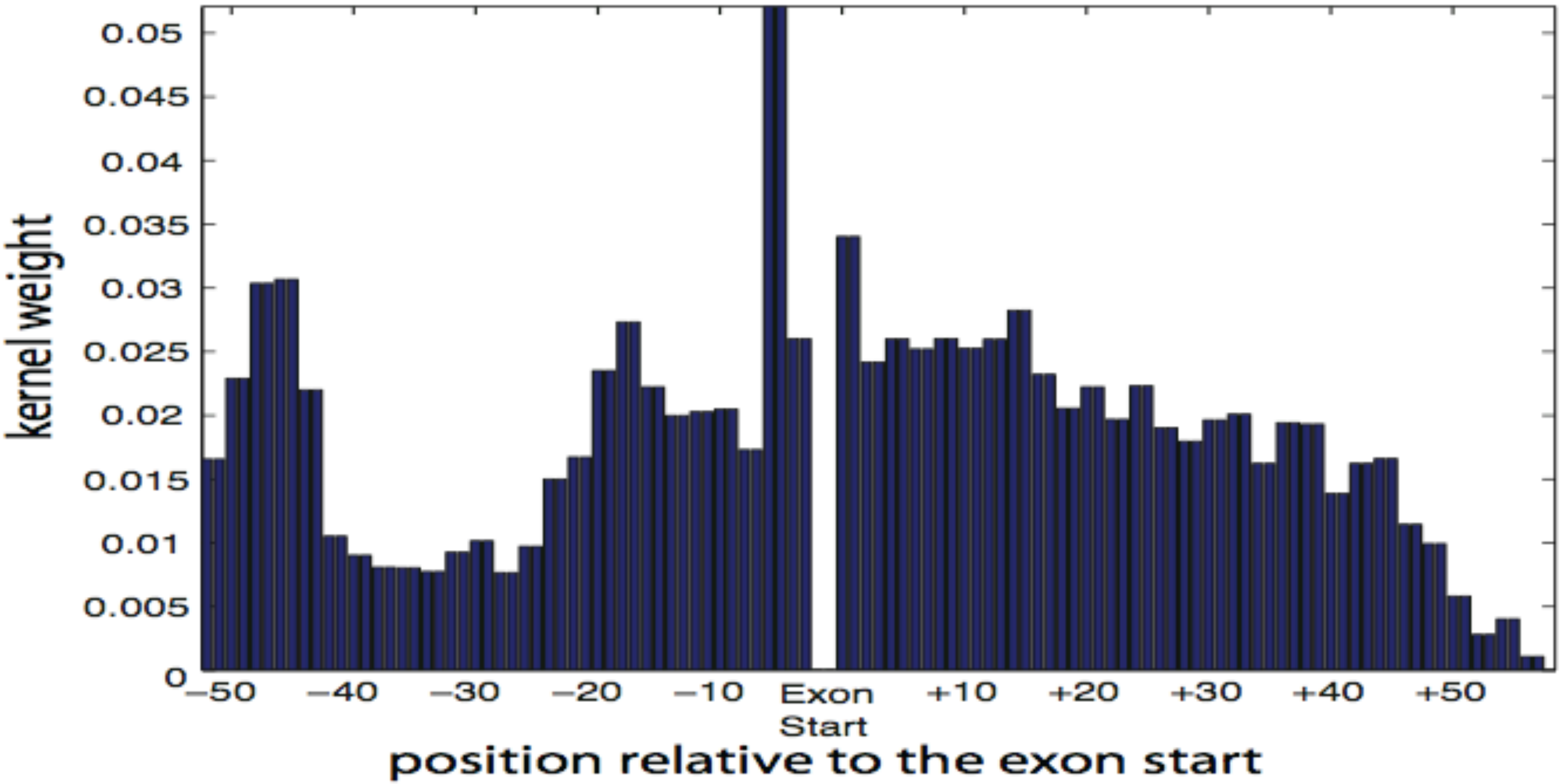
Measuring the performance wrt a ranking criteria

K_{SW}	K_{PF}	K_{LI}	K_B	K_D	$K_{R1...R6}$	$K_{R7...R12}$	TP1FP	ROC
5.08	0.31	0.22	0.39	0.00	–	–	$88.21 \pm 1.73\%$	0.9933 ± 0.0011
5.07	0.31	0.22	0.39	0.00	0.01	–	$88.19 \pm 1.60\%$	0.9932 ± 0.0011
5.06	0.30	0.22	0.38	0.01	0.02	0.01	$88.08 \pm 1.65\%$	0.9932 ± 0.0010
1.00	1.00	1.00	1.00	1.00	–	–	$75.20 \pm 2.38\%$	0.9906 ± 0.0012
1.00	1.00	1.00	1.00	1.00	1.00	–	$59.66 \pm 3.03\%$	0.9791 ± 0.0017
1.00	1.00	1.00	1.00	1.00	1.00	1.00	$42.87 \pm 2.59\%$	0.9620 ± 0.0027

Reality Check

(Sonnenburg et al., 2004)

- Importance weighting in a DNA sequence around a so-called splice site.



Learning Kernels - Theory

(Lanckriet et al., 2003)

- Linear classification, L_1 regularization:

$$R(h) \leq R_\rho(h) + \widetilde{O}\left(p \, \frac{1/\rho^2}{m}\right)$$

$\widetilde{O}$ hides logarithmic factors,

$\widehat{R}_\rho(h)$ fraction of training points with margin $< \rho$.

Learning Kernels - Theory

(Srebro & Ben-David, 2006)

- Linear classification, L_1 regularization:

$$R(h) \leq \widehat{R}_\rho(h) + \widetilde{O}\left(\sqrt{\frac{p + 1/\rho^2}{m}}\right)$$

$\widetilde{O}$ hides logarithmic factors,

$\widehat{R}_\rho(h)$ fraction of training points with margin $< \rho$.

Hyperkernels

(Ong, Smola & Williamson, 2005)

- Kernels of kernels, infinitely many kernels.
- m^2 kernel parameters to optimize over.

$$K(x, x') = \sum_{i,j=1}^{m} \beta_{i,j} \underline{K}((x_i, x_j), (x, x'))$$

$$\forall x, x' \in X, \quad \beta_{i,j} \geq 0$$

- SDP problem.

Reality Check, Hyperkernels

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)

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

Learning Kernels - Theory

(CC et al, 2009)

- Regression, KRR L_2 regularization:

$$R(h) \leq \widehat{R}(h) + O\Big(\sqrt{p/m} + \sqrt{1/m}\Big)$$

  - additive term with number of kernels p.
  - technical condition (orthogonal kernels).
  - suggests using larger number of kernels p.

KRR L2, Problem Formulation

- Optimization problem:

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

with $\mathcal{M} = \{\boldsymbol{\mu} \colon \boldsymbol{\mu} \geq 0 \wedge \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|^2 \leq \Lambda^2\}.$

L2 regularization

Form of the Solution

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \underbrace{\sum_{k=1}^{p}\mu_k\boldsymbol{\alpha}^\top\mathbf{K}_k\boldsymbol{\alpha}}_{\boldsymbol{\mu}^\top\mathbf{v}} + 2\boldsymbol{\alpha}^\top\mathbf{y}$$

$$\max_{\boldsymbol{\alpha}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y} + \min_{\boldsymbol{\mu}\in\mathcal{M}} -\boldsymbol{\mu}^\top\mathbf{v}$$ (von Neumann)

$$\max_{\boldsymbol{\alpha}} \underbrace{-\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y} - \boldsymbol{\mu}_0^\top\mathbf{v}}_{\text{standard KRR with } \boldsymbol{\mu}_0\text{-kernel } \mathbf{K}_0.} - \Lambda\|\mathbf{v}\|$$ (solve min. prob.)

$$\boldsymbol{\alpha} = \left(\sum_{k=1}^{p}\mu_k\mathbf{K}_k + \lambda\mathbf{I}\right)^{-1}\mathbf{y}$$ with $$\begin{cases} \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \Lambda\dfrac{\mathbf{v}}{\|\mathbf{v}\|} \\ v_k = \boldsymbol{\alpha}^\top\mathbf{K}_k\boldsymbol{\alpha} \end{cases}$$

Algorithm

Algorithm 1 Interpolated Iterative Algorithm

Input: $\mathbf{K}_k,\ k \in [1, p]$

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

repeat

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

  $\mathbf{v} \leftarrow (\boldsymbol{\alpha}^\top K_1 \boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top K_p \boldsymbol{\alpha})^\top$

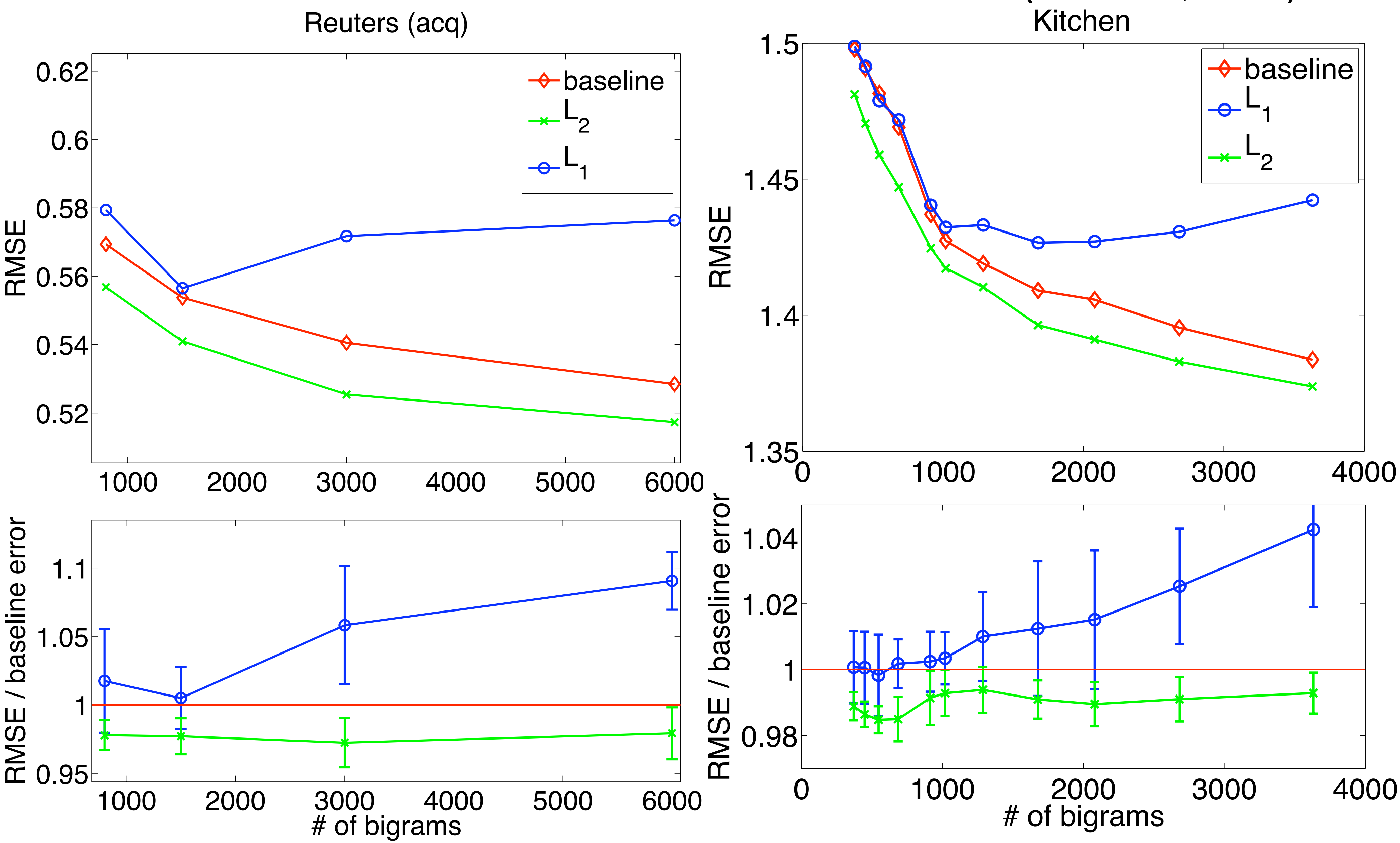
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_0 + \Lambda \frac{\mathbf{v}}{\|\mathbf{v}\|}$

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

until $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| < \epsilon$

Reality Check, KRR, Rank-1 Kernels

(CC et al, 2009)



Hierarchical Kernel Learning

(Bach, 2008)

- Example: polynomial kernels:
  - 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, complexity polynomial in the number of kernels selected, sparsity through L_1 regularization and hierarchical selection criteria.

Reality Check, HKL

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**

Summary

- Does not consistently and significantly outperform unweighted combinations.
    - L_2 regularization may work better than L_1.
    - Large number of kernels helps performance.
- Much faster.
- Great for feature selection.
- What about using non-linear combinations of kernels?

Non-Linear Combinations - Examples

- DC-Programming algorithm (Argyriou et al., 2005)
- Generalized MKL (Varma & Babu, 2009)
- Other non-linear combination studies.
  - Non-convex optimization problems.
  - Theoretical guarantees?
  - Can they improve performance substantially?

DC-Programming Problem

(Argyriou et al., 2005)

- Optimize over a continuously parameterized set of kernels.
- Kernels with bounded norm; Gaussians with the variance restricted to lie in a bounded interval.

$$K_{\boldsymbol{\sigma}}(x, x') = \prod_{i=1}^{d} \exp\left(- \frac{(x_i - x'_i)^2}{\sigma_i^2} \right)$$

- Alternate steps:

  - estimate new Gaussian;

  - fit the data.

Reality Check, DC-Programming

Table 1. Misclassification error percentage for learning one kernel parameter on the MNIST tasks.

Task	DC	standard	finite	SVM	DC	standard	finite	SVM	DC	standard	finite	SVM
	$\sigma \in [75, 25000]$				$\sigma \in [100, 10000]$				$\sigma \in [500, 5000]$			
odd vs. even	6.5	6.6	18.0	11.8	6.5	6.6	10.9	8.6	6.5	6.5	6.7	6.9
3 vs. 8	3.7	3.8	6.9	6.0	3.9	3.8	4.9	5.1	3.6	3.8	3.7	3.8
4 vs. 7	2.7	2.5	4.2	2.8	2.4	2.5	2.7	2.6	2.3	2.5	2.6	2.3

Table 2. Misclassification error percentage of DC algorithm vs. finite grid for 2 parameters on the MNIST tasks.

Task	DC	5×5	10×10	DC	5×5	10×10	DC	5×5	10×10
	$\sigma \in [75, 25000]$			$\sigma \in [100, 10000]$			$\sigma \in [500, 5000]$		
odd vs. even	5.8	15.8	11.2	5.8	10.1	6.2	5.8	6.8	5.8
3 vs. 8	2.7	6.5	5.1	2.5	4.6	2.5	2.6	3.5	2.5
4 vs. 7	1.8	3.9	2.9	1.7	2.7	2.0	1.8	2.0	1.8

Learning the σ (s) in a Gaussian kernel, DC formulation.

Generalized MKL

(Varma & Babu, 2009)

- Product kernel, GMKL:
- Gaussian: $K_{\boldsymbol{\sigma}}(x, x') = \prod_{i=1}^{d} \exp\left(- \frac{(x_i - x'_i)^2}{\sigma_i^2} \right)$
- Polynomial: $K_{\boldsymbol{d}}(x, x') = \left(\sum_{i=1}^{d} 1 + \mu_i x_i x'_i \right)^p, \quad \mu_i \geq 0$
- Non-convex optimization problem, gradient descent algorithm alternating with solving the SVM problem.

Reality Check, GMKL

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)

Future directions

- Get it to work!
- Can theory guide us to how?
- Should we change paradigm?