Infinite Kernel Learning

Peter Gehler and Sebastian Nowozin

NIPS workshop on Automatic selection of optimal kernels
13.12.2008

December 13, 2008

MAX-PLANCK-GESELLSCHAFT

BIOLOGISCHE KYBERNETIK

Main Results

1. Multiple Kernel Learning (MKL) can be extended to an infinite number of kernels (Argyriou et.al [1]).
2. A new Infinite Kernel Learning (IKL) algorithm (also a MKL algorithm).
3. No performance gain by linearly combining kernels on many standard benchmark datasets.
4. Using IKL with a much enriched kernel class (Gaussians with arbitrary covariance) can improve results considerably.

Main Results

1. Multiple Kernel Learning (MKL) can be extended to an infinite number of kernels (Argyriou et.al [1]).
2. A new Infinite Kernel Learning (IKL) algorithm (also a MKL algorithm).
3. No performance gain by linearly combining kernels on many standard benchmark datasets.
4. Using IKL with a much enriched kernel class (Gaussians with arbitrary covariance) can improve results considerably.

Main Results

1. Multiple Kernel Learning (MKL) can be extended to an infinite number of kernels (Argyriou et.al [1]).
2. A new Infinite Kernel Learning (IKL) algorithm (also a MKL algorithm).
3. No performance gain by linearly combining kernels on many standard benchmark datasets.
4. Using IKL with a much enriched kernel class (Gaussians with arbitrary covariance) can improve results considerably.

Main Results

1. Multiple Kernel Learning (MKL) can be extended to an infinite number of kernels (Argyriou et.al [1]).
2. A new Infinite Kernel Learning (IKL) algorithm (also a MKL algorithm).
3. No performance gain by linearly combining kernels on many standard benchmark datasets.
4. Using IKL with a much enriched kernel class (Gaussians with arbitrary covariance) can improve results considerably.

Notations

- kernel $k(x, x'; \theta)$ with θ specifies the parameters of the kernel *and* its type.
- Θ_f finite set
- Θ arbitrary set
- $k(x, x') = \sum_{\theta \in \Theta_f} d_\theta k(x, x'; \theta)$

SVM → MKL → IKL

Needed

- Regularization Parameter C, kernel $k(\cdot,\cdot;\theta)$
- Finite set of kernels $\Theta_f = \{\theta_1, \theta_2, \dots, \theta_K\}$
- Kernel parameters Θ

$$\min_{d,v,\xi,b} \quad \sum_{\theta\in\Theta_f} \frac{1}{d_\theta}\|v_\theta\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{sb.t.} \quad y_i\left(\sum_{\theta\in\Theta_f}\langle v_\theta, \phi_\theta(x_i)\rangle + b\right) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

$$\sum_{\theta\in\Theta_f} d_\theta = 1, \qquad d_\theta \geq 0.$$

SVM → MKL → IKL

Needed

- Regularization Parameter C, kernel $k(\cdot,\cdot;\theta)$
- Finite set of kernels $\Theta_f = \{\theta_1, \theta_2, \dots, \theta_K\}$
- Kernel parameters Θ

$$\min_{d,v,\xi,b} \quad \sum_{\theta\in\Theta_f} \frac{1}{d_\theta}\|v_\theta\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{sb.t.} \quad y_i\left(\sum_{\theta\in\Theta_f}\langle v_\theta, \phi_\theta(x_i)\rangle + b\right) \geq 1-\xi_i$$

$$\xi_i \geq 0$$

$$\sum_{\theta\in\Theta_f} d_\theta = 1, \qquad d_\theta \geq 0.$$

SVM → MKL → IKL

Needed

- Regularization Parameter C, kernel $k(\cdot,\cdot;\theta)$
- Finite set of kernels $\Theta_f = \{\theta_1, \theta_2, \ldots, \theta_K\}$
- Kernel parameters Θ

$$\min_{\Theta_f \subset \Theta} \quad \min_{d,v,\xi,b} \quad \sum_{\theta \in \Theta_f} \frac{1}{d_\theta} \|v_\theta\|^2 + C \sum_{i=1}^{N} \xi_i$$

$$\text{sb.t.} \quad y_i \left(\sum_{\theta \in \Theta_f} \langle v_\theta, \phi_\theta(x_i) \rangle + b \right) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

$$\sum_{\theta \in \Theta_f} d_\theta = 1, \qquad d_\theta \geq 0.$$

The Dual Program

$$\begin{aligned}
\max_{\alpha,\lambda} \quad & \sum_{i=1}^{N} \alpha_i - \lambda \\
\text{sb.t.} \quad & \alpha \in \mathbb{R}^N, \ \lambda \in \mathbb{R} \\
& 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \\
& \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j; \theta) \leq \lambda \quad \forall \theta \in \Theta_f
\end{aligned}$$

- λ is the Lagrange multiplier for $\int_\Theta d_\theta \mathrm{d}\theta = 1$
- Finite number of variables, finite number of constraints

The Dual Program

$$
\begin{aligned}
\max_{\alpha,\lambda} \quad & \sum_{i=1}^{N} \alpha_i - \lambda \\
\text{sb.t.} \quad & \alpha \in \mathbb{R}^N, \; \lambda \in \mathbb{R} \\
& 0 \le \alpha_i \le C, \quad i = 1, \dots, N \\
& \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j; \theta) \le \lambda \quad \forall \theta \in \Theta
\end{aligned}
$$

- λ is the Lagrange multiplier for $\int_\Theta d_\theta \mathrm{d}\theta = 1$
- Finite number of variables, infinite number of constraints

The Dual Program

$$\max_{\alpha,\lambda} \quad \sum_{i=1}^{N} \alpha_i - \lambda$$

$$\text{sb.t.} \quad \alpha \in \mathbb{R}^N, \; \lambda \in \mathbb{R}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N$$

$$\underbrace{\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j; \theta)}_{T(\theta;\alpha)} \leq \lambda \quad \forall \theta \in \Theta$$

- λ is the Lagrange multiplier for $\int_\Theta d_\theta \mathrm{d}\theta = 1$
- Finite number of variables, infinite number of constraints

IKL algorithm

- Delayed constraint generation algorithm.
- Iterate between
  1. restricted master problem: $(\alpha, b, d_\theta, \lambda) \leftarrow$ MKL solution with Θ_f
  2. Subproblem: $\theta_v \leftarrow \arg\max_{\theta \in \Theta} T(\theta; \alpha)$
  3. if $T(\theta_v; \alpha) \geq \lambda$ include θ_v, otherwise stop

IKL algorithm

- Delayed constraint generation algorithm.
- Iterate between
  1. restricted master problem: $(\alpha, b, d_\theta, \lambda) \leftarrow$ MKL solution with Θ_f
  2. Subproblem: $\theta_v \leftarrow \arg\max_{\theta \in \Theta} T(\theta; \alpha)$
  3. if $T(\theta_v; \alpha) \geq \lambda$ include θ_v, otherwise stop

IKL algorithm

- Delayed constraint generation algorithm.
- Iterate between
  1. restricted master problem: $(\alpha, b, d_\theta, \lambda) \leftarrow$ MKL solution with Θ_f
  2. Subproblem: $\theta_v \leftarrow \arg\max_{\theta \in \Theta} T(\theta; \alpha)$
  3. if $T(\theta_v; \alpha) \geq \lambda$ include θ_v, otherwise stop

IKL algorithm

- Delayed constraint generation algorithm.
- Iterate between
  1. restricted master problem: $(\alpha, b, d_\theta, \lambda) \leftarrow$ MKL solution with Θ_f
  2. Subproblem: $\theta_v \leftarrow \arg\max_{\theta \in \Theta} T(\theta; \alpha)$
  3. if $T(\theta_v; \alpha) \geq \lambda$ include θ_v, otherwise stop

The Subproblem

Problem

Given the parameters $0 \leq \alpha_i \leq C$ *and training points* $\{x_i, y_i\}$, $i = 1, \ldots, N$, *solve*

$$\theta_v = \arg\max_{\theta \in \Theta} T(\theta; \alpha) = \arg\max_{\theta \in \Theta} \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j; \theta).$$

- T is not convex
- Subproblem is a weighted, unnormalized version of *Kernel Target Alignment* [2]

The Subproblem

Problem

Given the parameters $0 \leq \alpha_i \leq C$ *and training points* $\{x_i, y_i\}$, $i = 1, \ldots, N$, *solve*

$$\theta_v = \arg\max_{\theta \in \Theta} T(\theta; \alpha) = \arg\max_{\theta \in \Theta} \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j; \theta).$$

- T is not convex
- Subproblem is a weighted, unnormalized version of *Kernel Target Alignment* [2]

Theoretical guarantees

Results from Hettich & Kortanek,[5]

Theorem

If for all $\theta \in \Theta$ and for all $\alpha \in [0, C]^N$ we have $T(\theta; \alpha) < \infty$, then there exists a finite set $\Theta_f \subset \Theta$ for which the Dual Program achieves its optimum.

Theorem

If the subproblem T can be solved, the IKL-Algorithm either stops after a finite number of iterations or has at least one point of accumulation and each one of these points solve the IKL program.

Solving the subproblem

- [1] devise a DC algorithm to solve optimally - only for low dimensional problems
- Give up on global optimality.
- We solve via gradient ascent for differentiable $k(\cdot, \cdot; \theta)$ w.r.t. θ using many different starting points

A Teaser

- We learn with all kernels of the form ($\theta = \{\gamma_1, \gamma_2, \ldots\}, \gamma_i \geq 0$)

$$k(x, x') = \sum d_\theta \exp(-\sum_{k=1}^{20} \gamma_k (x_k - x'_k)^2)$$

- $d_{\theta_1} = 0.98, \theta_1 = (1.1, 3.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0)$
- $d_{\theta_2} = 0.02, \theta_2 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 2.1, 0, 0.2, 0, \ldots, 0)$

A Teaser

- We learn with all kernels of the form $(\theta = \{\gamma_1, \gamma_2, \dots\}, \gamma_i \geq 0)$

$$k(x, x') = \sum d_\theta \exp(-\sum_{k=1}^{20} \gamma_k (x_k - x'_k)^2)$$

- $d_{\theta_1} = 0.98, \theta_1 = (1.1, 3.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \dots, 0)$
- $d_{\theta_2} = 0.02, \theta_2 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 2.1, 0, 0.2, 0, \dots, 0)$

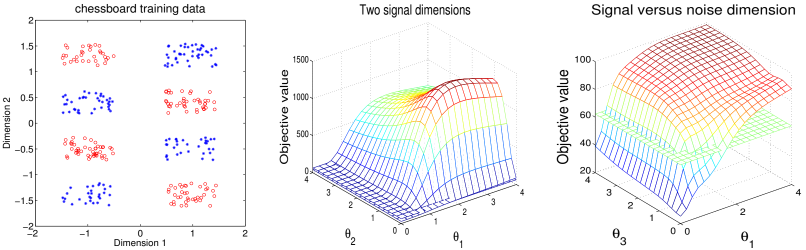
A Teaser

- We learn with all kernels of the form $(\theta = \{\gamma_1, \gamma_2, \ldots\}, \gamma_i \geq 0)$

$$k(x, x') = \sum d_\theta \exp(-\sum_{k=1}^{20} \gamma_k (x_k - x'_k)^2)$$

- $d_{\theta_1} = 0.98, \theta_1 = (1.1, 3.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \ldots, 0)$
- $d_{\theta_2} = 0.02, \theta_2 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 2.1, 0, 0.2, 0, \ldots, 0)$

IKL step by step



Kernel classes

1. (single) Gaussian with 1 bandwidth

$$k(x, x'; \theta) = \exp\left(-\theta \|x - x'\|^2\right)$$

2. (separate) As (single) + kernels for each dimension seperately

$$k(x, x'; \theta) = \exp\left(-\theta_k (x_k - x'_k)^2\right)$$

3. (products) Gaussian kernels with arbitrary non-negative bandwidths $\theta \in [0, 30]^K$

$$k(x, x'; \theta) = \exp\left(-\sum_{k=1}^{K} \theta_k (x_k - x'_k)^2\right)$$

Kernel classes

1. (single) Gaussian with 1 bandwidth

$$k(x, x'; \theta) = \exp\left(-\theta \|x - x'\|^2\right)$$

2. (separate) As (single) + kernels for each dimension seperately

$$k(x, x'; \theta) = \exp\left(-\theta_k (x_k - x'_k)^2\right)$$

3. (products) Gaussian kernels with arbitrary non-negative bandwidths $\theta \in [0, 30]^K$

$$k(x, x'; \theta) = \exp\left(-\sum_{k=1}^{K} \theta_k (x_k - x'_k)^2\right)$$

Kernel classes

1. (single) Gaussian with 1 bandwidth

$$k(x, x'; \theta) = \exp\left(-\theta \|x - x'\|^2\right)$$

2. (separate) As (single) + kernels for each dimension seperately

$$k(x, x'; \theta) = \exp\left(-\theta_k (x_k - x'_k)^2\right)$$

3. (products) Gaussian kernels with arbitrary non-negative bandwidths $\theta \in [0, 30]^K$

$$k(x, x'; \theta) = \exp\left(-\sum_{k=1}^{K} \theta_k (x_k - x'_k)^2\right)$$

Models

We compare three different models

- SVM: one kernel via CV
- MKL: with (single) + (separate)
- IKL: with (single) + (products)

Regularization parameter C estimated by CV.

Benchmark Datasets : class (single)

- Averaged over 100 runs

			(single)				
			SVM	MKL		IKL	
Dataset	#dim	#tr / #te	err	err	#k	err	#k
Banana	2	400/4900	10.5 ± 0.5	10.5 ± 0.5	1.0	10.6 ± 0.5	2.3
Breast-cancer	9	200/77	25.9 ± 4.3	27.9 ± 4.0	2.3	26.9 ± 4.7	2.9
Diabetis	8	468/300	23.2 ± 1.6	24.2 ± 1.9	2.8	23.8 ± 1.7	3.4
Flare-Solar	9	666/400	32.4 ± 1.7	35.1 ± 1.7	1.9	35.0 ± 1.8	2.2
German	20	700/300	23.7 ± 2.1	25.3 ± 2.3	2.0	25.3 ± 2.5	3.4
Heart	13	170/100	15.2 ± 3.1	16.4 ± 3.3	1.0	16.9 ± 3.2	2.5
Image	18	130/1010	3.0 ± 0.6	3.3 ± 0.7	1.0	3.4 ± 0.6	5.3
Ringnorm	20	400/7000	1.6 ± 0.1	1.6 ± 0.1	1.0	1.6 ± 0.1	1.2
Splice	60	1000/2175	10.6 ± 0.7	11.1 ± 0.7	2.0	12.6 ± 0.9	2.0
Thyroid	5	140/75	4.0 ± 2.2	4.7 ± 2.1	1.0	3.6 ± 2.1	3.2
Titanic	3	150/2051	22.9 ± 1.2	22.4 ± 1.0	1.1	22.5 ± 1.1	2.2
Twonorm	20	400/7000	2.5 ± 0.1	2.5 ± 0.1	2.0	2.6 ± 0.2	2.0
Waveform	21	400/4600	10.1 ± 0.5	9.9 ± 0.4	2.9	9.9 ± 0.4	2.5

Benchmark Datasets: classes (separate),(products)

				(separate)		(products)	
			SVM	MKL		IKL	
Dataset	#dim	#tr / #te	err	err	#k	err	#k
Banana	2	400/4900	10.5 ± 0.5	10.5 ± 0.5	1.0	10.7 ± 0.5	3.7
Breast-cancer	9	200/77	25.9 ± 4.3	26.7 ± 4.2	4.5	25.7 ± 4.1	16.1
Diabetis	8	468/300	23.2 ± 1.6	24.5 ± 1.6	4.0	24.3 ± 1.8	22.3
Flare-Solar	9	666/400	32.4 ± 1.7	34.3 ± 2.1	2.9	32.8 ± 1.9	2.6
German	20	700/300	23.7 ± 2.1	25.1 ± 2.2	8.3	24.6 ± 2.4	46.1
Heart	13	170/100	15.2 ± 3.1	16.7 ± 4.1	9.0	*20.1 ± 3.6*	28.2
Image	18	130/1010	3.0 ± 0.6	3.0 ± 0.6	1.6	**1.4** ± **0.3**	27.1
Ringnorm	20	400/7000	1.6 ± 0.1	1.7 ± 0.1	2.6	*2.1 ± 0.2*	16.3
Splice	60	1000/2175	10.6 ± 0.7	**6.0** ± **0.4**	24.1	**3.1** ± **0.3**	72.8
Thyroid	5	140/75	4.0 ± 2.2	4.7 ± 2.1	1.0	4.1 ± 2.0	12.7
Titanic	3	150/2051	22.9 ± 1.2	22.4 ± 1.0	1.9	22.4 ± 1.1	5.2
Twonorm	20	400/7000	2.5 ± 0.1	2.5 ± 0.1	3.8	*3.8 ± 0.4*	36.2
Waveform	21	400/4600	10.1 ± 0.5	10.2 ± 0.4	9.7	11.4 ± 0.6	33.7

Benchmark Datasets: classes (separate),(products)

				(separate)		**(products)**	
			SVM	MKL		IKL	
Dataset	#dim	#tr / #te	err	err	#k	err	#k
Banana	2	400/4900	10.5 ± 0.5	10.5 ± 0.5	1.0	10.7 ± 0.5	3.7
Breast-cancer	9	200/77	25.9 ± 4.3	26.7 ± 4.2	4.5	25.7 ± 4.1	16.1
Diabetis	8	468/300	23.2 ± 1.6	24.5 ± 1.6	4.0	24.3 ± 1.8	22.3
Flare-Solar	9	666/400	32.4 ± 1.7	34.3 ± 2.1	2.9	32.8 ± 1.9	2.6
German	20	700/300	23.7 ± 2.1	25.1 ± 2.2	8.3	24.6 ± 2.4	46.1
Heart	13	170/100	15.2 ± 3.1	16.7 ± 4.1	9.0	*20.1 ± 3.6*	28.2
Image	18	130/1010	3.0 ± 0.6	3.0 ± 0.6	1.6	**1.4** ± **0.3**	27.1
Ringnorm	20	400/7000	1.6 ± 0.1	1.7 ± 0.1	2.6	*2.1 ± 0.2*	16.3
Splice	60	1000/2175	10.6 ± 0.7	**6.0** ± **0.4**	24.1	**3.1** ± **0.3**	72.8
Thyroid	5	140/75	4.0 ± 2.2	4.7 ± 2.1	1.0	4.1 ± 2.0	12.7
Titanic	3	150/2051	22.9 ± 1.2	22.4 ± 1.0	1.9	22.4 ± 1.1	5.2
Twonorm	20	400/7000	2.5 ± 0.1	2.5 ± 0.1	3.8	*3.8 ± 0.4*	36.2
Waveform	21	400/4600	10.1 ± 0.5	10.2 ± 0.4	9.7	11.4 ± 0.6	33.7

Multiclass Datasets [3]

- One-Versus-Rest, averaged over 20 predefined splits

	(single)					**(separate)**		**(products)**	
	SVM	MKL		IKL		MKL		IKL	
Dataset	err	err	#k	err	#k	err	#k	err	#k
WAV	15.6 ± 1.2	15.5 ± 0.6	2.7	15.8 ± 0.7	2.1	16.4 ± 1.7	13.6	*18.0* ± *1.0*	35.1
SEG	6.5 ± 1.0	6.8 ± 0.9	2.8	6.9 ± 0.9	3.7	**5.0** ± **0.7**	8.4	**3.0** ± **0.5**	18.0
ABE	1.1 ± 0.3	0.8 ± 0.3	2.5	0.8 ± 0.3	3.0	0.7 ± 0.3	11.3	0.7 ± 0.2	33.8
SAT	10.4 ± 0.4	10.2 ± 0.3	3.6	10.1 ± 0.4	4.0	n/a		n/a	
DNA	7.7 ± 0.7	7.8 ± 0.7	1.4	7.7 ± 0.8	2.0	n/a		n/a	

Dataset	#dim	#tr / #te	#cl
WAV	21	300/4700	3
SEG	17	500/1810	7
ABE	16	560/1763	3
SAT	36	1500/4935	6
DNA	181	500/2686	3

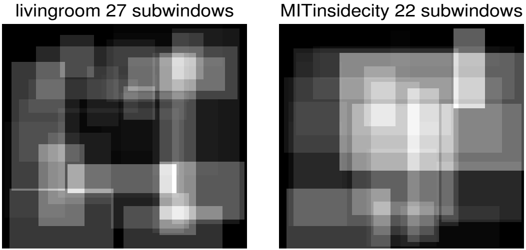
Main Results

1. Multiple Kernel Learning (MKL) can be extended to an infinite number of kernels [1]
2. An Infinite Kernel Learning (IKL) algorithm (which is also a MKL algorithm)
3. No performance gain by linearly combining kernels on many standard benchmark datasets
4. Using IKL with a much enriched kernel class (Gaussians with arbitrary covariance) may improve results tremendously

Technical Report available [4].

Future work

- How much is lost by approximately solving $T(\theta; \alpha)$?
- Efficient ways to solve $T(\theta; \alpha)$.
- Application to structured kernels (under submission)
  - learning the spatial layout of a pyramid match kernel
  - Dictionary learning as Kernel learning



References

A. Argyriou, R. Hauser, C. A. Micchelli, and M. Pontil.
A dc-programming algorithm for kernel selection.
In *ICML '06*, 2006.

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola.
On kernel-target alignment.
In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

K. Duan and S. Keerthi.
Which is the best multiclass svm method? an empirical study.
In *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 278–285, 2005.

P. V. Gehler and S. Nowozin.
Infinite kernel learning.
Technical Report 178, Max Planck Institute for Biological Cybernetics, 2008.

R. Hettich and K. O. Kortanek.
Semi-infinite programming: theory, methods, and applications.
SIAM Rev., 35(3):380–429, 1993.

IKL algorithm

Input: Training set X, Regularizer C, Kernel Class Θ.
Output: The classification function $f(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}$.

Select any $\theta_v \in \Theta$ and set $\Theta_0 = \{\theta_v\}$
$t \leftarrow 0$
loop
    $(\alpha, b, d_\theta, \lambda) \leftarrow$ MKL solution with Θ_t ▷ Solve MKL
    $\theta_v \leftarrow \arg\max_{\theta \in \Theta} T(\theta; \alpha)$ ▷ Solve subproblem
    if $T(\theta_v; \alpha) > \lambda$ **then**
        $\Theta_{t+1} = \Theta_t \cup \{\theta_v\}$
    else
        break
    end if
    $t \leftarrow t + 1$
end loop