Maximum Margin based Semi-supervised Spectral Kernel Learning

Zenglin Xu, Jianke Zhu, Michael R. Lyu and Irwin King

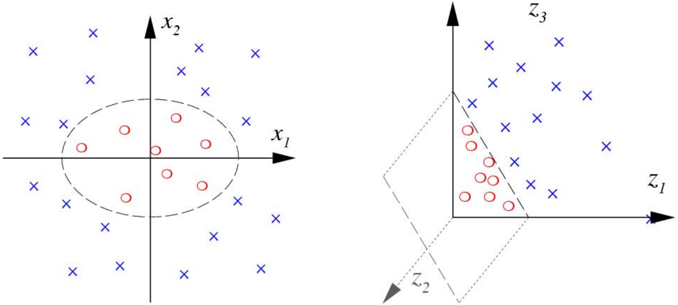
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Internet Joint Conference on Neural Networks 2007

Outline

Let's Start from the Kernel Trick

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



Kernel Learning

- Different kernel functions defines a different implicit mapping (linear kernel, RBF kernel, etc.)
- How to find an appropriate kernel?
- This leads to the kernel learning task.

Definition

Kernel Learning works by embedding data from the input space to a Hilbert space, and then searching for relations among the embedded data points to maximize a performance measure.

Semi-supervised Kernel Learning

We design a kernel using both:

- the label information of labeled data
- the unlabeled data

Spectral Kernel Learning

Given an input kernel K, a spectral kernel is obtained by adjusting the spectra of K

$$\bar{K} = \sum_{i=1}^{n} g(\mu_i)\phi_i\phi_i^T, \quad (1)$$

where $g(\cdot)$ is a transformation function of the spectra of a kernel matrix, $< \mu_i, \phi_i >$ is the i-th eigenvalue and eigenvector.

Typical Approaches in Spectral Kernel Learning

- Diffusion kernels, [Kondor and Lafferty, 02]
- Regularization on graphs, [Smola and Kondor, 03]
- Non-parametric spectral kernel learning, [Zhu et al., 03]
- Fast decay spectral kernel, [Hoi et al., 06]

The Property and Limitation in Previous Approaches

Property

- Distances on the graph can give a useful, more global, sense of similarity between objects

Limitation

- The kernel designing process does not involve the bias or the decision boundary of a kernel-based learning algorithm

Why the Bias is Important?

Different kernel methods try to utilize different prior knowledge in order to derive the separating hyperplane

- SVM maximizes the margin between two classes of data in the kernel induced feature space
- Kernel Fisher Discriminant Analysis (KFDA) maximizes the between-class covariance while minimizes the within-class covariance
- Minimax Probability Machine (MPM) finds a hyperplane in the feature space, which minimizes the maximum Mahalanobis distances to two classes

Our Supplement to Spectral Kernel Methods

This motivates us to design spectral kernel learning algorithms:

- Keep the properties of spectral kernels
- Incorporate the decision boundary of a kernel-based classifier

Our Contributions

- We generalize the previous work in spectral kernel learning to a spectral kernel learning framework
- We incorporate the decision boundary of a classifier into the spectral kernel learning process

An Illustration

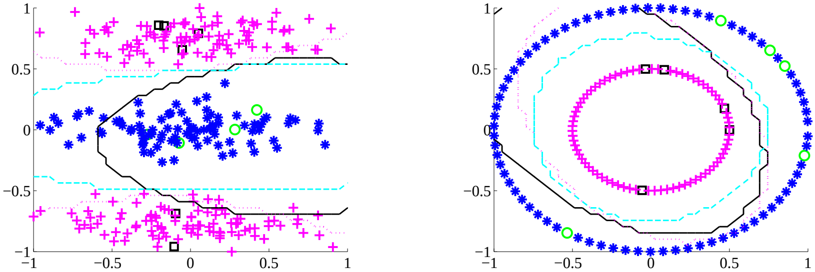


Figure: The decision boundaries on **Relevance** and **Twocircles**.

- The black (dark) line – regular RBF
- The magenta (doted) line – spectral kernel optimizing the kernel target alignment [Hoi et al., 06]
- The cyan (dashed) line – proposed spectral kernel attained by maximizing the margin

The Framework

- Theoretical foundation
- Semi-supervised spectral kernel learning framework
- Maximum-margin based spectral kernel learning

Spectral Kernel Design Rule

We consider the following regularized linear prediction method on the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$:

$$\hat{f} = \arg \inf_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} L(h(\mathbf{x}_i), y_i) + r||h||_{\mathcal{H}}^2, \tag{2}$$

where r is a regularization coefficient, ℓ is the number of labeled data points, and L is a loss function.
Based on Representer Theorem, we have

$$\hat{f} = \arg \inf_{f \in \mathcal{R}^n} \frac{1}{\ell} \sum_{i=1}^{\ell} L(f_i, y_i) + r f^T K^{-1} f. \tag{3}$$

Spectral Kernel Design Rule

- The previous formulation is equivalent to a supervised learning model.
- A way of unsupervised kernel design is to replace the kernel matrix K with $\bar{K}$, i.e.,

$$\bar{K} = \sum_{i=1}^{n} g(\mu_i)\phi_i\phi_i^T. \tag{4}$$

Spectral Kernel Design Rule

Depending on different forms of $g(\cdot)$, different kernel matrices can be learned.

Table: Semi-supervised kernels achieved by different spectral transformation.

$g(\mu)$	**Kernels**
$g(\mu) = \exp(-\frac{\sigma^2}{2}\mu)$	the diffusion kernel
$g(\mu) = \frac{1}{\mu+\epsilon}$	the Gaussian field kernel
$g(\mu) = \mu_i, \mu_i \leq \mu_{i+1}, i = 1, \ldots, n-1$	the order-constrained spectral kernel
$g(\mu) = \mu_i, \mu_i \geq w\mu_{i+1}, i = 1, \ldots, q-1$	the fast-decay spectral kernel

Optimization Criteria

There are several performance measure for kernel learning:

- Kernel Target Alignment
- Soft Margin
- Fisher Discriminant Ratio
- Others

Kernel Target Alignment

The empirical alignment of a kernel κ_1 with a kernel κ_2 with respect to the sample $\mathcal{X}$ is the quantity:

$$\omega_A(\mathcal{X}, \kappa_1, \kappa_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}}, \quad (5)$$

where K_i is the kernel matrix for the sample $\mathcal{X}$ using the kernel function κ_i and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product between two matrices, i.e., $\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{n} \kappa_1(\mathbf{x_1}, \mathbf{x}_2)\kappa_2(\mathbf{x_1}, \mathbf{x}_2)$.

Soft Margin

Given a labeled sample $\mathcal{X}_l$, the hyperplane $(\mathbf{w}_*, b_*)$ that solves the optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{l} \xi_i \\ \text{s. t.} \quad & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b \rangle) \geq 1 - \xi_i,\ i = 1, \ldots, l, \\ & \xi_i \geq 0, \end{aligned} \tag{6}$$

realizes the maximal margin classifier with geometric margin $\gamma = 1/||\mathbf{w}_*||_2$, assuming it exists.

Spectral Kernel Learning Framework

We summarize the spectral kernel learning framework

$$\max_{g(\mu)} \quad \omega(\bar{K}) \tag{7}$$
$$\text{s. t.} \quad \bar{K} = \sum_{i=1}^{n} g(\mu_i)\phi_i\phi_i^T,$$

where $\omega(\bar{K})$ is a generalized performance measure, such as the kernel target alignment, the soft margin, etc.

Spectral Kernel Learning Framework

According to [Hoi et al., 06], a fast spectral decay rate benefits the kernel design. Adjusting the spectral decay rate, we have

$$\begin{aligned} \max_{\mu} \quad & \omega(\bar{K}) \\ \text{s. t.} \quad & \bar{K} = \sum_{i=1}^{q} \mu_i \phi_i \phi_i^T, \\ & trace(\bar{K}) = \delta, \\ & \mu_i \geq 0, \\ & \mu_i \geq w \mu_{i+1}, i = 1, \ldots, q-1, \end{aligned} \tag{8}$$

where $w \geq 1$ specifies the spectral decay rate and q specifies the number of eigen-pairs selected.

Maximum Margin Based Spectral Kernel Learning

By maximizing the margin between two classes, we have the following semi-supervised learning problem:

$$\max_{\mu,\alpha} \quad 2\alpha^T \mathbf{e} - \alpha^T G(\bar{K}^{tr})\alpha \tag{9}$$

$$\text{s. t.} \quad \bar{K} = \sum_{i=1}^{d} \mu_i \phi_i \phi_i^T, \; trace(\bar{K}) = \delta,$$

$$\alpha^T \mathbf{y} = 0, \; 0 \leq \alpha_j \leq C, \; j = 1, \ldots, n,$$

$$\mu_i \geq 0, i = 1, \ldots, q \;\; \mu_i \geq w\mu_{i+1}, i = 1, \ldots, q-1,$$

where $G(\bar{K}^{tr}) = D(\mathbf{y})\bar{K}^{tr}D(\mathbf{y})$, $D(\mathbf{y})$ is the diagonal matrix of the label vector **y**.

Maximum Margin Based Spectral Kernel Learning

We note each rank-one kernel matrix as $\bar{K}_i = \phi_i \phi_i^T$. Following [Lanckriet et al., 04], we have:

$$\begin{aligned} \max_{\alpha,\mu} \quad & 2\alpha^T \mathbf{e} - \delta\rho \\ \text{s. t.} \quad & \delta = \mu^T \mathbf{t},\ \mu_i \geq \mathbf{0},\ i = 1, \ldots, q \\ & \rho \geq \frac{1}{t_i}\alpha^T G(\bar{K}_i^{tr})\alpha,\ 1 \leq i \leq q, \\ & \alpha^T \mathbf{y} = 0,\ 0 \leq \alpha_j \leq C,\ j = 1, \ldots, n, \\ & \mu_i \geq w\mu_{i+1},\ i = 1, \ldots, q-1, \end{aligned} \tag{10}$$

where $\mathbf{t} = \{t_1, t_2, \ldots, t_q\}$ is the trace vector of $\bar{K}_i$, i.e., $trace(\bar{K}_i) = t_i$.

Experiment Setup

- Data sets:
  - Two toy data sets
  - Four UCI data sets
- Comparison methods:
  - Standard linear kernel and RBF kernel
  - Order-constrained spectral kernel (abbreviated as “order”)
  - Fast-decay spectral kernel optimizing the kernel alignment (noted as “KA”)
- Procedure:
  - 20 random trials
  - 10-fold cross-validation
  - Training data size from 10 to 30

Toy Data Sets

Table: Experimental results on two synthetic data sets (%).

Algorithm	Relevance	Twocircles
RBF	81.52±4.63	78.74±5.02
KA	91.27±4.57	84.10±4.44
MM	**93.15**±3.49	**94.98**±3.13

UCI Data Sets

Table: Classification performance of different kernels

Training Size	Standard Kernels: Linear	Standard Kernels: RBF	Semi-supervised Kernels: Order	Semi-supervised Kernels: KA (Linear)	Semi-supervised Kernels: KA (RBF)	Semi-supervised Kernels: MM (Linear)	Semi-supervised Kernels: MM (RBF)
Ionosphere (%)							
10	71.51±2.12	66.56±2.04	62.31±3.92	74.36±2.47	70.24±4.99	**74.45**±2.54	69.56±2.26
20	77.50±1.20	71.37±2.48	63.64±2.71	78.75±1.89	76.62±3.12	**78.83**±1.74	77.55±3.04
30	80.23±0.90	77.82±2.52	63.52±2.44	81.21±1.17	80.51±2.80	81.47±1.08	**82.5**9±0.96
Banana (%)							
10	53.69±1.69	55.63±2.07	50.22±0.94	53.87±1.34	62.68±2.18	53.95±1.54	**64.92**±2.26
20	55.30±1.86	58.73±2.39	50.44±0.93	54.74±1.63	66.18±2.46	55.14±1.76	**69.88**±1.87
30	56.07±2.43	60.48±1.57	50.73±0.93	55.72±1.55	69.33±1.96	56.24±2.07	**74.87**±1.33
Sonar (%)							
10	63.89±2.25	57.52±1.70	49.96±1.16	**64.30**±1.88	60.92±2.22	64.14±1.77	61.95±2.44
20	68.72±1.50	65.73±1.71	49.80±0.62	69.17±1.64	67.91±1.87	68.94±1.49	**69.18**±1.73
30	71.98±1.20	71.20±1.32	49.73±1.09	72.31±1.86	70.90±1.34	**73.22**±1.61	71.32±1.60
Solar-flare (%)							
10	55.92±1.78	56.58±2.53	51.45±1.83	57.75±2.08	57.88±2.23	**58.11**±1.92	57.95±1.93
20	59.73±1.97	60.44±2.27	51.14±1.56	60.64±1.84	60.87±1.96	60.60±1.68	**61.08**±1.77
30	61.77±1.44	61.67±1.53	50.85±2.06	**62.19**±1.01	62.14±1.42	61.95±1.21	61.75±1.11

Summary

- We discuss a semi-supervised spectral kernel learning framework
- To supplement this framework, we incorporate the decision boundary into the kernel learning process

Future Work

- Extend the semi-supervised kernel learning to multi-way classification
- Apply the proposed method to some applications, such as text categorization, where the data sets have a cluster structure

QA

Thanks for your attention!