Metric Embedding for Kernel Classification Rules

Bharath K. Sriperumbudur

University of California, San Diego

(Joint work with Omer Lang & Gert Lanckriet)

Introduction

- Parzen window methods are popular in density estimation, kernel regression etc.
- We consider these rules for classification.

Set up:

- Binary classification: $\{(X_i, Y_i)\}_{i=1}^n \sim \mathcal{D}$, $X_i \in \mathbb{R}^D$ and $Y_i \in \{0,1\}$. Classify $x \in \mathbb{R}^D$.
- Kernel classification rule: [Devroye et al., 1996]

$$g_n(x) = \begin{cases} 0 & \text{if } \sum_{i=1}^n \mathbb{1}_{\{Y_i=0\}} K\left(\frac{x-X_i}{h}\right) \geq \sum_{i=1}^n \mathbb{1}_{\{Y_i=1\}} K\left(\frac{x-X_i}{h}\right) \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where $K : \mathbb{R}^D \to \mathbb{R}$ is a smoothing kernel function which is usually non-negative (not necessarily positive definite).

Introduction

- Parzen window methods are popular in density estimation, kernel regression etc.
- We consider these rules for classification.

Set up:

- Binary classification: $\{(X_i, Y_i)\}_{i=1}^n \sim \mathcal{D}$, $X_i \in \mathbb{R}^D$ and $Y_i \in \{0,1\}$. Classify $x \in \mathbb{R}^D$.
- Kernel classification rule: [Devroye et al., 1996]

$$g_n(x) = \begin{cases} 0 & \text{if } \sum_{i=1}^n \mathbb{1}_{\{Y_i=0\}} K\left(\frac{x-X_i}{h}\right) \geq \sum_{i=1}^n \mathbb{1}_{\{Y_i=1\}} K\left(\frac{x-X_i}{h}\right) \\ 1 & \text{otherwise,} \end{cases} \quad (1)$$

where $K : \mathbb{R}^D \to \mathbb{R}$ is a smoothing kernel function which is usually non-negative (not necessarily positive definite).

Kernel Classification Rule

Examples:

- Gaussian kernel: $K(x) = e^{-\|x\|_2^2}$ (p.d.)
- Cauchy kernel: $K(x) = (1 + \|x\|_2^{D+1})^{-1}$ (p.d.)
- Naïve kernel: $K(x) = \mathbb{1}_{\{\|x\|_2 \le 1\}}$ (not p.d.)
- Epanechnikov kernel: $K(x) = (1 - \|x\|_2^2)\mathbb{1}_{\{\|x\|_2 \le 1\}}$ (not p.d.)

- Naïve kernel: performs h-ball nearest neighbor (NN) classification.
- K is p.d.: Eq. (1) is similar to the RKHS based kernel rule.
- How to choose h: only asymptotic guarantees for universal consistency are available [Devroye and Krzyżak, 1989].

Kernel Classification Rule

Examples:

- Gaussian kernel: $K(x) = e^{-\|x\|_2^2}$ (p.d.)
- Cauchy kernel: $K(x) = (1 + \|x\|_2^{D+1})^{-1}$ (p.d.)
- Naïve kernel: $K(x) = \mathbb{1}_{\{\|x\|_2 \leq 1\}}$ (not p.d.)
- Epanechnikov kernel: $K(x) = (1 - \|x\|_2^2)\mathbb{1}_{\{\|x\|_2 \leq 1\}}$ (not p.d.)

- Naïve kernel: performs h-ball nearest neighbor (NN) classification.
- K is p.d.: Eq. (1) is similar to the RKHS based kernel rule.
- How to choose h: only asymptotic guarantees for universal consistency are available [Devroye and Krzyżak, 1989].

Metric Learning for k-NN

- Dependence on the metric: Finite-sample risk of the k-NN rule may be reduced by using a weighted Euclidean metric, even though the infinite sample risk is independent of the metric used [Snapp and Venkatesh, 1998].
- Experimentally verified by:
  - [Xing et al., 2003]
  - NCA [Goldberger et al., 2005]
  - MLCC [Globerson and Roweis, 2006]
  - LMNN [Weinberger et al., 2006]
- All these methods learn $\mathbb{L} \in \mathbb{R}^{D \times D}$ so that $x \mapsto \mathbb{L}x$.

Metric Learning for k-NN

- Dependence on the metric: Finite-sample risk of the k-NN rule may be reduced by using a weighted Euclidean metric, even though the infinite sample risk is independent of the metric used [Snapp and Venkatesh, 1998].
- Experimentally verified by:
  - [Xing et al., 2003]
  - NCA [Goldberger et al., 2005]
  - MLCC [Globerson and Roweis, 2006]
  - LMNN [Weinberger et al., 2006]
- All these methods learn $\mathbb{L} \in \mathbb{R}^{D \times D}$ so that $x \mapsto \mathbb{L}x$.

Metric Embedding: Motivation

- Some applications need natural distance measures that reflect the underlying structure of the data.
  - Distance between images : tangent distance
  - Distance between points on a manifold : geodesic distance
- Usually, Euclidean or weighted Euclidean distance is used as a surrogate.
- In the absence of prior knowledge, the data may be used to select the suitable metric.

Questions we address: Find

- $\varphi : (\mathcal{X}, \rho) \to (\mathcal{Y}, \ell_2)$.
- h

Metric Embedding: Motivation

- Some applications need natural distance measures that reflect the underlying structure of the data.
  - Distance between images : tangent distance
  - Distance between points on a manifold : geodesic distance
- Usually, Euclidean or weighted Euclidean distance is used as a surrogate.
- In the absence of prior knowledge, the data may be used to select the suitable metric.

Questions we address: Find

- $\varphi : (\mathcal{X}, \rho) \rightarrow (\mathcal{Y}, \ell_2)$.
- h

Problem Formulation

Multi-class classification:

$$g_n(x) = \arg\max_{l \in [L]} \sum_{i=1}^{n} [\![Y_i = l]\!]\, [\![\rho(x, X_i) \le h]\!] \tag{2}$$

where $[L] := \{1, \ldots, L\}$, $[\![a]\!] = \mathbb{1}_{\{a\}}$ and $\rho(x, X_i) \stackrel{!}{=} \|\varphi(x) - \varphi(X_i)\|_2$.

Goal: To learn φ and h by minimizing the probability of error associated with g_n.

$$(\varphi^*, h^*) = \arg\min_{\varphi, h} \Pr_{(X,Y)\in\mathcal{D}}(g_n(X) \neq Y) \tag{3}$$

Regularized problem:

$$\min_{\varphi, h>0} \sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!] + \lambda\,\Omega[\varphi], \quad \lambda > 0 \tag{4}$$

Problem Formulation

Multi-class classification:

$$g_n(x) = \arg\max_{l \in [L]} \sum_{i=1}^{n} [\![Y_i = l]\!] \, [\![\rho(x, X_i) \le h]\!] \tag{2}$$

where $[L] := \{1, \ldots, L\}$, $[\![a]\!] = \mathbb{1}_{\{a\}}$ and $\rho(x, X_i) \stackrel{!}{=} \|\varphi(x) - \varphi(X_i)\|_2$.

Goal: To learn φ and h by minimizing the probability of error associated with g_n.

$$(\varphi^*, h^*) = \arg\min_{\varphi, h} \Pr_{(X,Y)\in\mathcal{D}}(g_n(X) \neq Y) \tag{3}$$

Regularized problem:

$$\min_{\varphi, h>0} \sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!] + \lambda\,\Omega[\varphi], \quad \lambda > 0 \tag{4}$$

Problem Formulation

Multi-class classification:

$$g_n(x) = \arg\max_{l \in [L]} \sum_{i=1}^{n} [\![Y_i = l]\!]\, [\![\rho(x, X_i) \le h]\!] \tag{2}$$

where $[L] := \{1, \ldots, L\}$, $[\![a]\!] = \mathbb{1}_{\{a\}}$ and $\rho(x, X_i) \stackrel{!}{=} \|\varphi(x) - \varphi(X_i)\|_2$.

Goal: To learn φ and h by minimizing the probability of error associated with g_n.

$$(\varphi^*, h^*) = \arg\min_{\varphi, h} \Pr_{(X,Y) \in \mathcal{D}}(g_n(X) \neq Y) \tag{3}$$

Regularized problem:

$$\min_{\varphi, h > 0} \sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!] + \lambda\, \Omega[\varphi], \quad \lambda > 0 \tag{4}$$

Problem Formulation

Multi-class classification:

$$g_n(x) = \arg\max_{l \in [L]} \sum_{i=1}^{n} [\![Y_i = l]\!] \, [\![\rho(x, X_i) \le h]\!] \tag{2}$$

where $[L] := \{1, \ldots, L\}$, $[\![a]\!] = \mathbb{1}_{\{a\}}$ and $\rho(x, X_i) \stackrel{!}{=} \|\varphi(x) - \varphi(X_i)\|_2$.

Goal: To learn φ and h by minimizing the probability of error associated with g_n.

$$(\varphi^*, h^*) = \arg\min_{\varphi, h} \Pr_{(X,Y)\in\mathcal{D}}(g_n(X) \neq Y) \tag{3}$$

Regularized problem:

$$\min_{\varphi, h>0} \sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!] + \lambda \, \Omega[\varphi], \quad \lambda > 0 \tag{4}$$

Problem Formulation

Minimize an upper bound on $\sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!]$, followed with hinge-relaxation of $[\![.]\!]$, we get

$$min_{\varphi,\tilde{h}} \sum_{i=1}^{n} \left[2 - n_i^+ + \sum_{j=1}^{n} \left[1 + \tau_{ij} \|\varphi(X_i) - \varphi(X_j)\|_2^2 - \tau_{ij}\tilde{h} \right]_+ \right]_+ + \lambda\,\Omega[\varphi] \quad (5)$$

where $\tilde{h} = h^2$, $\tau_{ij} = 2\delta_{Y_i,Y_j} - 1$ and $n_i^+ = \sum_{j=1}^{n} [\![\tau_{ij} = 1]\!]$.

Choice of φ:

- Suppose φ is a Mercer kernel map : $\langle \varphi(x), \varphi(y) \rangle_{\ell_2} = \mathfrak{K}(x, y)$.
- $\|\varphi(X_i) - \varphi(X_j)\|_2^2$ is a function of $\mathfrak{K}$ alone.
- $\Omega[\varphi]$ is usually chosen as $\mathrm{tr}(\mathbf{K})$, $\|\mathbf{K}\|_F^2$ etc.
- This choice does not provide an out-of-sample extension.

Problem Formulation

Minimize an upper bound on $\sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!]$, followed with hinge-relaxation of $[\![.]\!]$, we get

$$min_{\varphi,\tilde{h}} \sum_{i=1}^{n} \left[2 - n_i^+ + \sum_{j=1}^{n} \left[1 + \tau_{ij} \|\varphi(X_i) - \varphi(X_j)\|_2^2 - \tau_{ij}\tilde{h} \right]_+ \right]_+ + \lambda\,\Omega[\varphi] \quad (5)$$

where $\tilde{h} = h^2$, $\tau_{ij} = 2\delta_{Y_i,Y_j} - 1$ and $n_i^+ = \sum_{j=1}^{n} [\![\tau_{ij} = 1]\!]$.

Choice of φ:

- Suppose φ is a Mercer kernel map : $\langle \varphi(x), \varphi(y) \rangle_{\ell_2} = \mathfrak{K}(x,y)$.
- $\|\varphi(X_i) - \varphi(X_j)\|_2^2$ is a function of $\mathfrak{K}$ alone.
- $\Omega[\varphi]$ is usually chosen as $\mathrm{tr}(\mathbf{K})$, $\|\mathbf{K}\|_F^2$ etc.
- This choice does not provide an out-of-sample extension.

Problem Formulation

Minimize an upper bound on $\sum_{i=1}^{n} [\![g_n(X_i) \neq Y_i]\!]$, followed with hinge-relaxation of $[\![.]\!]$, we get

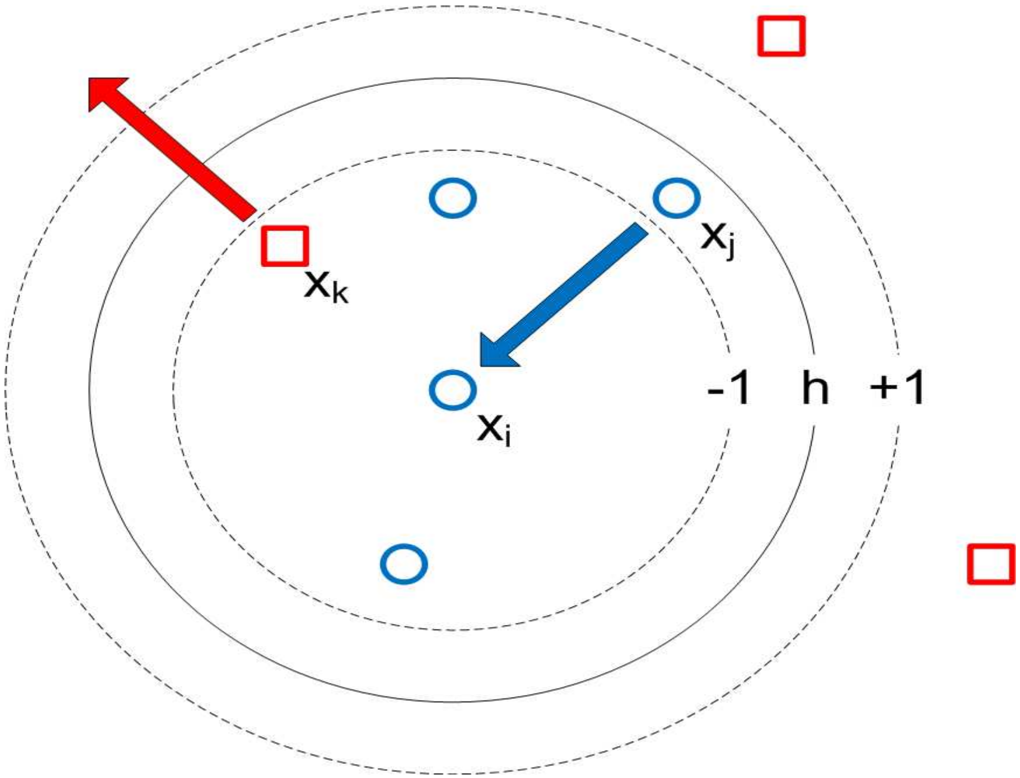
$$\min_{\varphi,\tilde{h}} \sum_{i=1}^{n} \left[2 - n_i^{+} + \sum_{j=1}^{n} \left[1 + \tau_{ij} \|\varphi(X_i) - \varphi(X_j)\|_2^2 - \tau_{ij}\tilde{h} \right]_{+} \right]_{+} + \lambda\,\Omega[\varphi] \quad (5)$$

where $\tilde{h} = h^2$, $\tau_{ij} = 2\delta_{Y_i,Y_j} - 1$ and $n_i^{+} = \sum_{j=1}^{n} [\![\tau_{ij} = 1]\!]$.

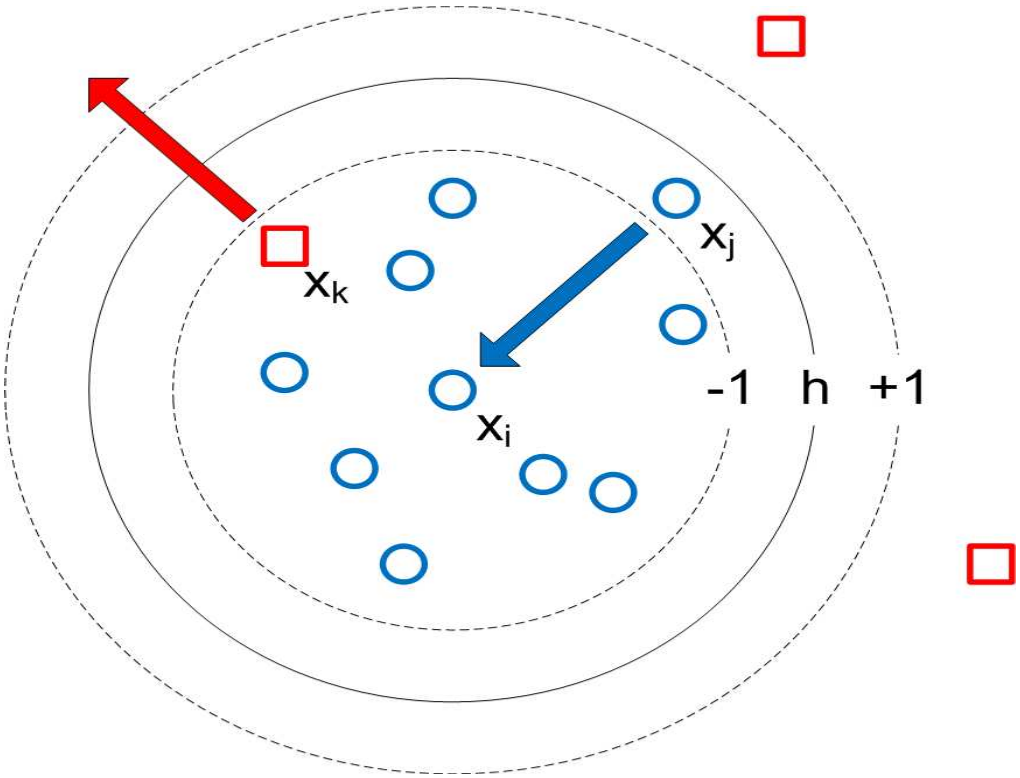
Choice of φ:

- Suppose φ is a Mercer kernel map : $\langle \varphi(x), \varphi(y) \rangle_{\ell_2} = \mathfrak{K}(x, y)$.
- $\|\varphi(X_i) - \varphi(X_j)\|_2^2$ is a function of $\mathfrak{K}$ alone.
- $\Omega[\varphi]$ is usually chosen as $\mathrm{tr}(\mathbf{K})$, $\|\mathbf{K}\|_F^2$ etc.
- This choice does not provide an out-of-sample extension.

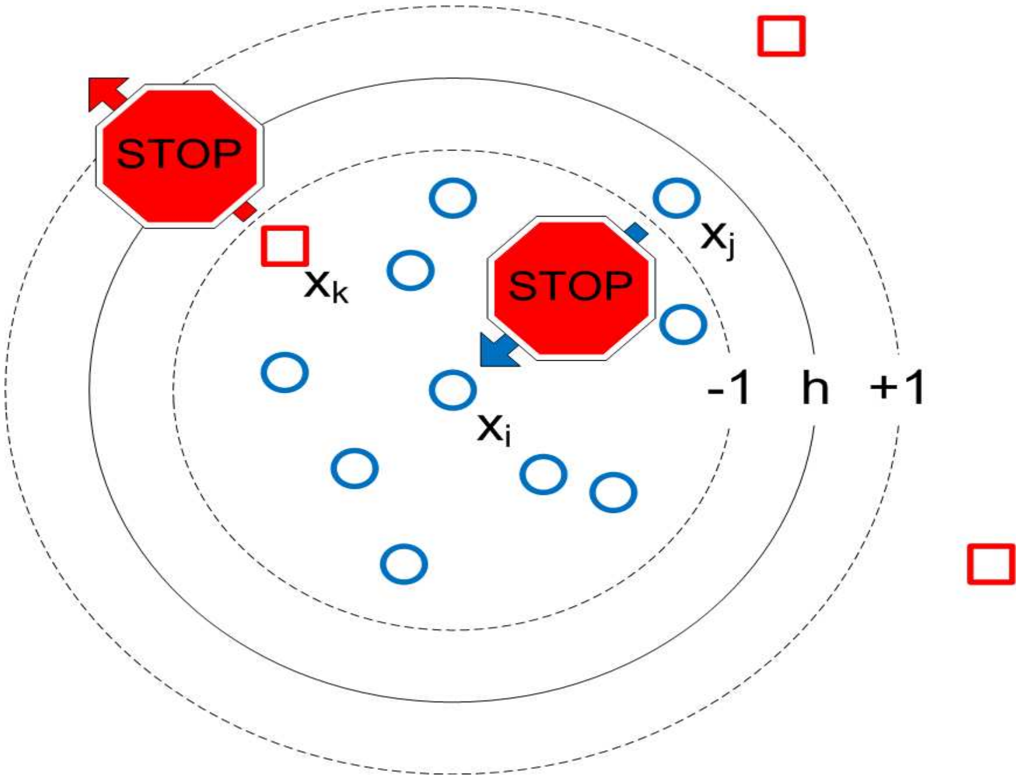
Problem Formulation



Problem Formulation



Problem Formulation



φ in an RKHS

Theorem (Multi-output regularization)

Suppose

- $\varphi = (\varphi_1, \ldots, \varphi_d)$, $\varphi_i : \mathcal{X} \to \mathbb{R}$.
- $\varphi_i \in (\mathcal{H}_i, \mathfrak{K}_i)$.

Then

- *Minimizer of Eq. (5) with* $\Omega[\varphi] = \sum_{i=1}^{d} \|\varphi_i\|^2_{\mathcal{H}_i}$ *is of the form*

$$\varphi_j = \sum_{i=1}^{n} c_{ij} \mathfrak{K}_j(., X_i), \; \forall\, j \in [d], \tag{6}$$

where $c_{ij} \in \mathbb{R}$ *and* $\sum_{i=1}^{n} c_{ij} = 0$, $\forall\, i \in [n]$, $\forall\, j \in [d]$.

φ in an RKHS

Corollary

Suppose

- $\mathfrak{K}_1 = \ldots = \mathfrak{K}_d = \mathfrak{K}$.

Then, $\|\varphi(x) - \varphi(y)\|_2^2$ is the Mahalanobis distance between the empirical kernel maps at x and y.

Corollary (Linear kernel)

Let

- $\mathcal{X} = \mathbb{R}^D$.
- $\mathfrak{K}(z, w) = \langle z, w \rangle_2 = z^T w$.

Then $\|\varphi(z) - \varphi(w)\|_2^2$ is the Mahalanobis distance between z and w.

φ in an RKHS

Corollary

Suppose

- $\mathfrak{K}_1 = \ldots = \mathfrak{K}_d = \mathfrak{K}$.

Then, $\|\varphi(x) - \varphi(y)\|_2^2$ is the Mahalanobis distance between the empirical kernel maps at x and y.

Corollary (Linear kernel)

Let

- $\mathcal{X} = \mathbb{R}^D$.
- $\mathfrak{K}(z, w) = \langle z, w \rangle_2 = z^T w$.

Then $\|\varphi(z) - \varphi(w)\|_2^2$ is the Mahalanobis distance between z and w.

Semidefinite Relaxation

- Non-convex (d.c. program):

$$\begin{aligned} \min_{C,\tilde{h}} \quad & \sum_{i=1}^{n}\left[2-n_i^{+}+\sum_{j=1}^{n}\left[1+\tau_{ij}\mathrm{tr}(CM_{ij}C^T)-\tau_{ij}\tilde{h}\right]_+\right]_+ + \lambda\,\mathrm{tr}(C\mathbf{K}C^T) \\ \text{s.t.} \quad & C\in\mathbb{R}^{d\times n},\ C1=0,\ \tilde{h}>0, \end{aligned} \tag{7}$$

where $M_{ij} := (k^{X_i}-k^{X_j})(k^{X_i}-k^{X_j})^T$ and $k^{X_i} = [\mathfrak{K}(X_1,X_i),\ldots,\mathfrak{K}(X_n,X_i)]^T$.

- Semidefinite relaxation:

$$\begin{aligned} \min_{\Sigma,\tilde{h}} \quad & \sum_{i=1}^{n}\left[2-n_i^{+}+\sum_{j=1}^{n}\left[1+\tau_{ij}\mathrm{tr}(M_{ij}\Sigma)-\tau_{ij}\tilde{h}\right]_+\right]_+ + \lambda\,\mathrm{tr}(\mathbf{K}\Sigma) \\ \text{s.t.} \quad & \Sigma\succeq 0,\ 1^T\Sigma 1=0,\ \tilde{h}>0. \end{aligned} \tag{8}$$

Semidefinite Relaxation

- Non-convex (d.c. program):

$$\begin{aligned} \min_{C,\tilde{h}} \quad & \sum_{i=1}^{n} \left[2 - n_i^+ + \sum_{j=1}^{n} \left[1 + \tau_{ij} \mathrm{tr}(C M_{ij} C^T) - \tau_{ij} \tilde{h} \right]_+ \right]_+ + \lambda \, \mathrm{tr}(C \mathbf{K} C^T) \\ \text{s.t.} \quad & C \in \mathbb{R}^{d \times n}, \; C1 = 0, \; \tilde{h} > 0, \end{aligned} \tag{7}$$

where $M_{ij} := (k^{X_i} - k^{X_j})(k^{X_i} - k^{X_j})^T$ and $k^{X_i} = [\mathfrak{K}(X_1, X_i), \ldots, \mathfrak{K}(X_n, X_i)]^T$.

- Semidefinite relaxation:

$$\begin{aligned} \min_{\Sigma,\tilde{h}} \quad & \sum_{i=1}^{n} \left[2 - n_i^+ + \sum_{j=1}^{n} \left[1 + \tau_{ij} \mathrm{tr}(M_{ij} \Sigma) - \tau_{ij} \tilde{h} \right]_+ \right]_+ + \lambda \, \mathrm{tr}(\mathbf{K} \Sigma) \\ \text{s.t.} \quad & \Sigma \succeq 0, \; 1^T \Sigma 1 = 0, \; \tilde{h} > 0. \end{aligned} \tag{8}$$

Algorithm

- Active sets ($\mathcal{A}$): Find (i, j) for which the hinge functions are active.
- The program reduces to the form,

$$\begin{aligned} \min_{\Sigma, \tilde{h}} \quad & \mathrm{tr}(A\Sigma) + r\tilde{h} \\ \text{s.t.} \quad & \Sigma \succeq 0,\ 1^T\Sigma 1 = 0,\ \tilde{h} > 0. \end{aligned} \tag{9}$$

where $A = \lambda \mathbf{K} + \sum_{(i,j)\in\mathcal{A}} \tau_{ij} M_{ij}$ and $r = -\sum_{(i,j)\in\mathcal{A}} \tau_{ij}$.

- Alternatively solve for Σ and $\tilde{h}$ by gradient descent and projecting onto the convex constraint set.

Algorithm

- Active sets ($\mathcal{A}$): Find (i, j) for which the hinge functions are active.
- The program reduces to the form,

$$\begin{aligned} \min_{\Sigma,\tilde{h}} \quad & \mathrm{tr}(A\Sigma) + r\tilde{h} \\ \text{s.t.} \quad & \Sigma \succeq 0,\ 1^T\Sigma 1 = 0,\ \tilde{h} > 0. \end{aligned} \tag{9}$$

where $A = \lambda\mathbf{K} + \sum_{(i,j)\in\mathcal{A}} \tau_{ij} M_{ij}$ and $r = -\sum_{(i,j)\in\mathcal{A}} \tau_{ij}$.

- Alternatively solve for Σ and $\tilde{h}$ by gradient descent and projecting onto the convex constraint set.

Algorithm

- Active sets ($\mathcal{A}$): Find (i,j) for which the hinge functions are active.
- The program reduces to the form,

$$\begin{aligned} \min_{\Sigma,\tilde{h}} \quad & \operatorname{tr}(A\Sigma) + r\tilde{h} \\ \text{s.t.} \quad & \Sigma \succeq 0,\ 1^T\Sigma 1 = 0,\ \tilde{h} > 0. \end{aligned} \tag{9}$$

where $A = \lambda\mathbf{K} + \sum_{(i,j)\in\mathcal{A}} \tau_{ij} M_{ij}$ and $r = -\sum_{(i,j)\in\mathcal{A}} \tau_{ij}$.

- Alternatively solve for Σ and $\tilde{h}$ by gradient descent and projecting onto the convex constraint set.

Algorithm

Require: $\{M_{ij}\}_{i,j=1}^{n}$, $\mathbf{K}$, $\{\tau_{ij}\}_{i,j=1}^{n}$, $\{n_i^+\}_{i=1}^{n}$, $\lambda > 0$, $\epsilon > 0$ and $\{\alpha_i, \beta_i\} > 0$

Set $t = 0$. Choose $\Sigma_0 \in \mathcal{A}$ and $\tilde{h}_0 > 0$.

repeat

$\quad A_t = \{i : \sum_{j=1}^{n} \left[1 + \tau_{ij}\mathrm{tr}(M_{ij}\Sigma_t) - \tau_{ij}\tilde{h}_t\right]_+ + 2 \leq n_i^+\} \times \{j : j \in [n]\}$

$\quad B_t = \{(i,j) : 1 + \tau_{ij}\mathrm{tr}(M_{ij}\Sigma_t) > \tau_{ij}\tilde{h}_t\}$

$\quad N_t = B_t \backslash A_t$

$\quad \Sigma_{t+1} = P_{\mathcal{N}}(\Sigma_t - \alpha_t \sum_{(i,j)\in N_t} \tau_{ij} M_{ij} - \alpha_t \lambda \mathbf{K})$

$\quad \tilde{h}_{t+1} = \max(\epsilon, \tilde{h}_t + \beta_t \sum_{(i,j)\in N_t} \tau_{ij})$

$\quad t = t + 1$

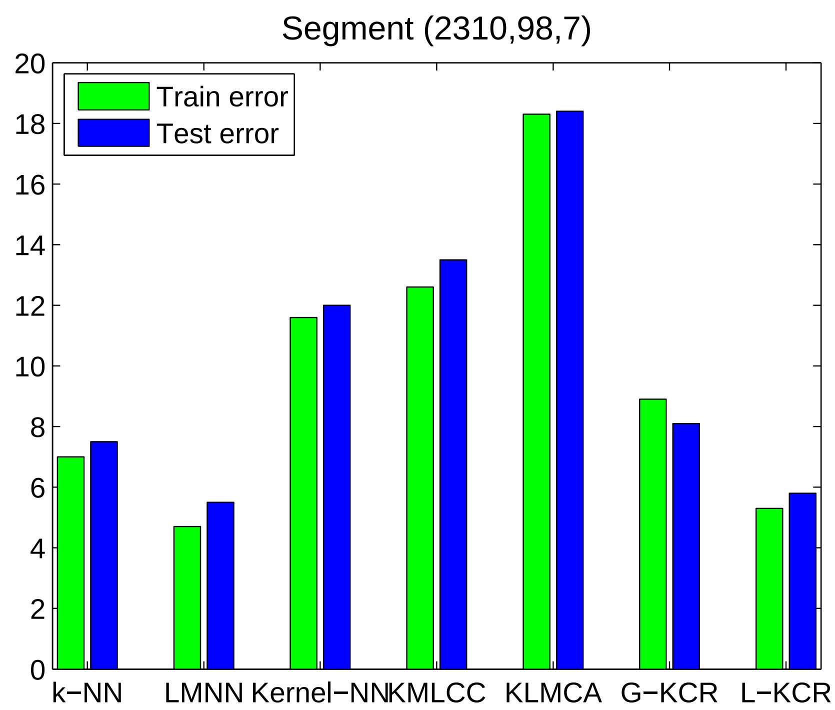
until convergence

return Σ_t, $\tilde{h}_t$

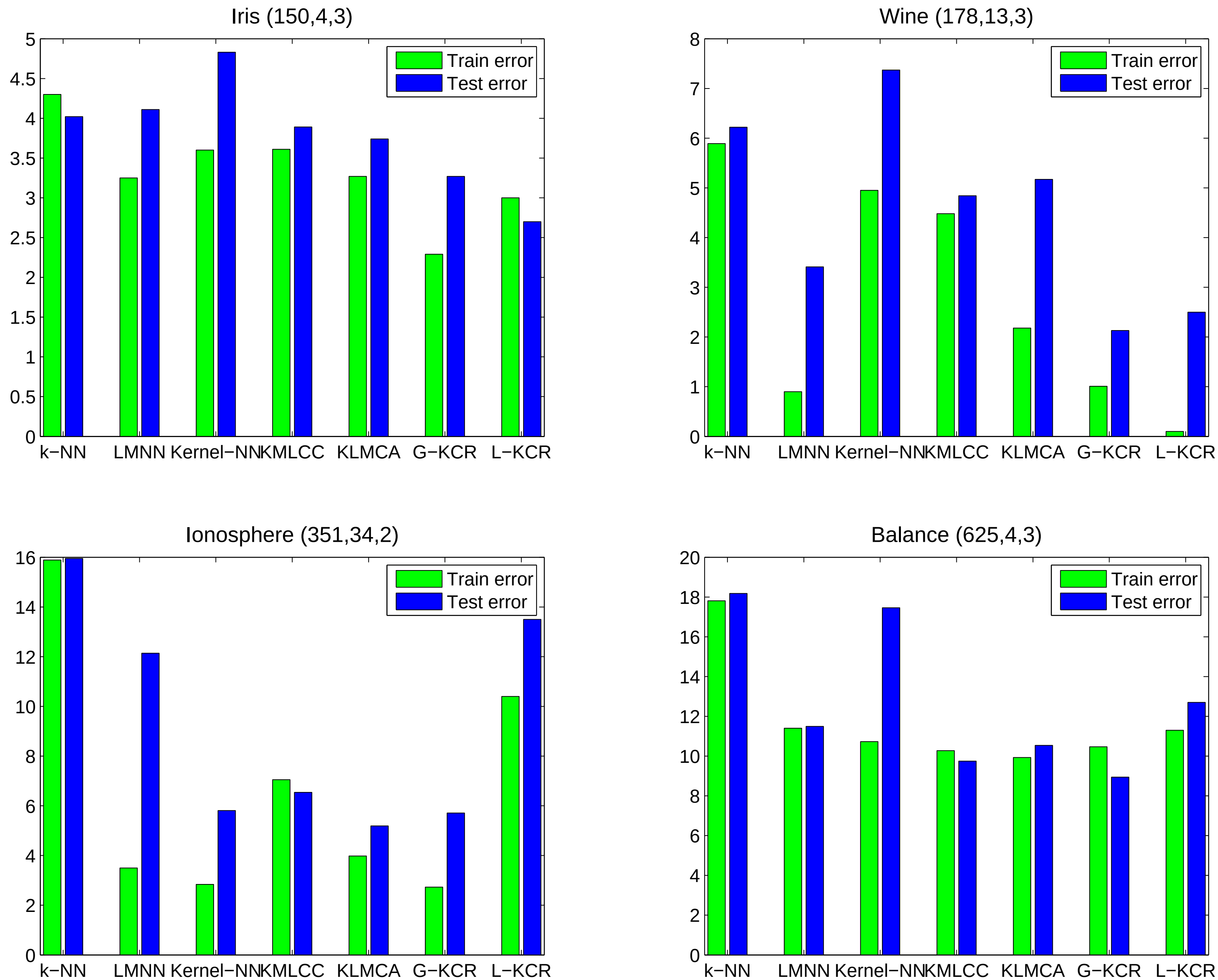
Experiments & Results

Set up:

- 5 UCI datasets
- Methods: k-NN, LMNN, Kernel-NN, KMLCC, KLMCA and KCR (proposed).
- Average error (training/testing) over 20 different splits.



Experiments & Results



Discussion & Summary

- Proposed a method to embed $(\mathcal{X}, \rho)$ into an ℓ_2 space for kernel classification rules.
- Learned the bandwidth of the Parzen window.
- LMNN requires target neighbors to be defined a priori whereas KCR does not require any such neighbors to be defined.
- Compared to LMNN and KLMNN, our method involves fewer tuning parameters.
- KCR provides a unified and formal treatment for solving metric learning problems while other methods have been built on heuristics.
- Issues: Computationally intensive $\sim O(n^3)$.

Discussion & Summary

- Proposed a method to embed $(\mathcal{X}, \rho)$ into an ℓ_2 space for kernel classification rules.
- Learned the bandwidth of the Parzen window.
- LMNN requires target neighbors to be defined a priori whereas KCR does not require any such neighbors to be defined.
- Compared to LMNN and KLMNN, our method involves fewer tuning parameters.
- KCR provides a unified and formal treatment for solving metric learning problems while other methods have been built on heuristics.
- Issues: Computationally intensive $\sim O(n^3)$.

Discussion & Summary

- Proposed a method to embed $(\mathcal{X}, \rho)$ into an ℓ_2 space for kernel classification rules.
- Learned the bandwidth of the Parzen window.
- LMNN requires target neighbors to be defined a priori whereas KCR does not require any such neighbors to be defined.
- Compared to LMNN and KLMNN, our method involves fewer tuning parameters.
- KCR provides a unified and formal treatment for solving metric learning problems while other methods have been built on heuristics.
- Issues: Computationally intensive $\sim O(n^3)$.

Discussion & Summary

- Proposed a method to embed $(\mathcal{X}, \rho)$ into an ℓ_2 space for kernel classification rules.
- Learned the bandwidth of the Parzen window.
- LMNN requires target neighbors to be defined a priori whereas KCR does not require any such neighbors to be defined.
- Compared to LMNN and KLMNN, our method involves fewer tuning parameters.
- KCR provides a unified and formal treatment for solving metric learning problems while other methods have been built on heuristics.
- Issues: Computationally intensive $\sim O(n^3)$.

Discussion & Summary

- Proposed a method to embed $(\mathcal{X}, \rho)$ into an ℓ_2 space for kernel classification rules.
- Learned the bandwidth of the Parzen window.
- LMNN requires target neighbors to be defined a priori whereas KCR does not require any such neighbors to be defined.
- Compared to LMNN and KLMNN, our method involves fewer tuning parameters.
- KCR provides a unified and formal treatment for solving metric learning problems while other methods have been built on heuristics.
- Issues: Computationally intensive $\sim O(n^3)$.

References

Devroye, L., Gyorfi, L., and Lugosi, G. (1996).
A Probabilistic Theory of Pattern Recognition.
Springer-Verlag, New York.

Devroye, L. and Krzyżak, A. (1989).
An equivalence theorem for L_1 convergence of the kernel regression estimate.
Journal of Statistical Planning and Inference, 23:71–82.

Globerson, A. and Roweis, S. (2006).
Metric learning by collapsing classes.
In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 451–458, Cambridge, MA. MIT Press.

Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2005).
Neighbourhood components analysis.
In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 513–520, Cambridge, MA. MIT Press.

Snapp, R. R. and Venkatesh, S. S. (1998).
Asymptotic expansions of the k nearest neighbor risk.
The Annals of Statistics, 26(3):850–878.

Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2006).
Distance metric learning for large margin nearest neighbor classification.
In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480, Cambridge, MA. MIT Press.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2003).
Distance metric learning with application to clustering with side-information.
In S. Becker, S. T. and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA. MIT Press.

Questions

Thank You