# Learning in Indefiniteness

Purushottam Kar

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

August 2, 2010

1. A brief introduction to learning

1. A brief introduction to learning
2. Kernels - Definite and Indefinite

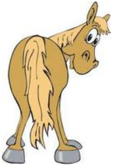
1. A brief introduction to learning
2. Kernels - Definite and Indefinite
3. Using kernels as measures of distance
   - Landmarking based approaches
   - Approximate embeddings into Pseudo Euclidean spaces
   - Exact embeddings into Banach spaces

1. A brief introduction to learning
2. Kernels - Definite and Indefinite
3. Using kernels as measures of distance
   - Landmarking based approaches
   - Approximate embeddings into Pseudo Euclidean spaces
   - Exact embeddings into Banach spaces
4. Using kernels as measures of similarity
   - Approximate embeddings into Pseudo Euclidean spaces
   - Exact embeddings into Kreĭn spaces
   - Landmarking based approaches

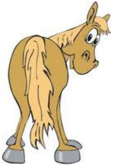
1. A brief introduction to learning
2. Kernels - Definite and Indefinite
3. Using kernels as measures of distance
   - Landmarking based approaches
   - Approximate embeddings into Pseudo Euclidean spaces
   - Exact embeddings into Banach spaces
4. Using kernels as measures of similarity
   - Approximate embeddings into Pseudo Euclidean spaces
   - Exact embeddings into Kreĭn spaces
   - Landmarking based approaches
5. Conclusion

# A Quiz

# A Quiz



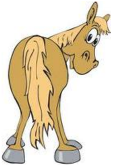
# A Quiz



# A Quiz








# Learning 100

# Learning as pattern recognition

- Binary classification

# Learning as pattern recognition

- Binary classification
- Multi-class classification

# Learning as pattern recognition

- Binary classification
- Multi-class classification
- Multi-label classification

# Learning as pattern recognition

- Binary classification
- Multi-class classification
- Multi-label classification
- Regression

# Learning as pattern recognition

- Binary classification
- Multi-class classification
- Multi-label classification
- Regression
- Clustering

# Learning as pattern recognition

- Binary classification
- Multi-class classification
- Multi-label classification
- Regression
- Clustering
- Ranking

# Learning as pattern recognition

- Binary classification
- Multi-class classification
- Multi-label classification
- Regression
- Clustering
- Ranking
- ...

# Learning as pattern recognition

- Binary classification ✓
- Multi-class classification
- Multi-label classification
- Regression
- Clustering
- Ranking
- ...

# Binary classification

- Learning Dichotomies from examples

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :
  - Generative (Bayesian classification)

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :
  - Generative (Bayesian classification)
  - Predictive

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :
  - Generative (Bayesian classification)
  - Predictive
    - Feature Based

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :
  - Generative (Bayesian classification)
  - Predictive
    - Feature Based
    - Kernel Based

# Binary classification

- Learning Dichotomies from examples
- Learning the distinction between a bird and a non-bird
- Main approaches :
  - Generative (Bayesian classification)
  - Predictive
    - ⋆ Feature Based
    - ⋆ Kernel Based ✓
- This talk : Kernel Based predictive approaches to binary classification

# Probably Approximately Correct learning [Kearns and Vazirani, 1997]

## Definition

A class of boolean functions $\mathcal{F}$ defined on a domain $\mathcal{X}$ is said to be PAC-learnable if there exists a class of boolean functions $\mathcal{H}$ defined on $\mathcal{X}$, an algorithm $\mathcal{A}$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all distributions $\mu$ defined on $\mathcal{X}$, all $t \in F$, all $\epsilon, \delta > 0$ : $\mathcal{A}$, when given $(x_i, f(x_i))_{i=1}^n, x_i \in_R \mu$ where $n = S(1/\epsilon, 1/\delta)$, returns with probability (taken over the choice of $x_1, \ldots, x_n$) greater than $1 - \delta$, a function $h \in \mathcal{H}$ such that

$$\Pr_{x \in_R \mu}[h(x) \neq t(x)] \leq \epsilon.$$

- $t$ is the Target function, $\mathcal{F}$ the Concept Class

# Probably Approximately Correct learning [Kearns and Vazirani, 1997]

## Definition

A class of boolean functions $\mathcal{F}$ defined on a domain $\mathcal{X}$ is said to be PAC-learnable if there exists a class of boolean functions $\mathcal{H}$ defined on $\mathcal{X}$, an algorithm $\mathcal{A}$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all distributions $\mu$ defined on $\mathcal{X}$, all $t \in F$, all $\epsilon, \delta > 0$ : $\mathcal{A}$, when given $(x_i, f(x_i))_{i=1}^n, x_i \in_R \mu$ where $n = S(1/\epsilon, 1/\delta)$, returns with probability (taken over the choice of $x_1, \ldots, x_n$) greater than $1 - \delta$, a function $h \in \mathcal{H}$ such that

$$\Pr_{x \in_R \mu}[h(x) \neq t(x)] \leq \epsilon.$$

- $t$ is the Target function, $\mathcal{F}$ the Concept Class
- $h$ is the Hypothesis, $\mathcal{H}$ the Hypothesis Class

# Probably Approximately Correct learning [Kearns and Vazirani, 1997]

## Definition

A class of boolean functions $\mathcal{F}$ defined on a domain $\mathcal{X}$ is said to be PAC-learnable if there exists a class of boolean functions $\mathcal{H}$ defined on $\mathcal{X}$, an algorithm $\mathcal{A}$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all distributions $\mu$ defined on $\mathcal{X}$, all $t \in F$, all $\epsilon, \delta > 0$ : $\mathcal{A}$, when given $(x_i, f(x_i))_{i=1}^n, x_i \in_R \mu$ where $n = S(1/\epsilon, 1/\delta)$, returns with probability (taken over the choice of $x_1, \ldots, x_n$) greater than $1 - \delta$, a function $h \in \mathcal{H}$ such that

$$\Pr_{x \in_R \mu} [h(x) \neq t(x)] \leq \epsilon.$$

- $t$ is the Target function, $\mathcal{F}$ the Concept Class
- $h$ is the Hypothesis, $\mathcal{H}$ the Hypothesis Class
- $S$ is the Sample Complexity of the algorithm $\mathcal{A}$

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning
  - Weak-PAC learning - require only that $\epsilon$ be bounded away from $\frac{1}{2}$

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning
  - Weak-PAC learning - require only that $\epsilon$ be bounded away from $\frac{1}{2}$
  - Restrict oneself to benign distributions (uniform, mixture of Gaussians)

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning
  - Weak-PAC learning - require only that $\epsilon$ be bounded away from $\frac{1}{2}$
  - Restrict oneself to benign distributions (uniform, mixture of Gaussians)
  - Restrict oneself to benign learning scenarios (target function-distribution pairs that are benign)

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning
  - Weak-PAC learning - require only that $\epsilon$ be bounded away from $\frac{1}{2}$
  - Restrict oneself to benign distributions (uniform, mixture of Gaussians)
  - Restrict oneself to benign learning scenarios (target function-distribution pairs that are benign)
  - Vaguely defined in literature

# Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities eg. Regular Languages
- Adversarial combinations of target functions and distributions can make learning impossible
- Weaker notions of learning
  - Weak-PAC learning - require only that $\epsilon$ be bounded away from $\frac{1}{2}$
  - Restrict oneself to benign distributions (uniform, mixture of Gaussians)
  - Restrict oneself to benign learning scenarios (target function-distribution pairs that are benign) ✓
  - Vaguely defined in literature

# Weak$^*$-Probably Approximately Correct learning

## Definition

A class of boolean functions $\mathcal{F}$ defined on a domain $\mathcal{X}$ is said to be weak$^*$-PAC-learnable if for every $t \in F$ and distribution $\mu$ defined on $\mathcal{X}$, there exists a class of boolean functions $\mathcal{H}$ defined on $\mathcal{X}$, an algorithm $\mathcal{A}$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all $\epsilon, \delta > 0$ : $\mathcal{A}$, when given $(x_i, f(x_i))_{i=1}^n, x_i \in_R \mu$ where $n = S(1/\epsilon, 1/\delta)$, returns with probability (taken over the choice of $x_1, \ldots, x_n$) greater than $1 - \delta$, a function $h \in \mathcal{H}$ such that

$$\Pr_{x \in_R \mu}[h(x) \neq t(x)] \leq \epsilon.$$

# Kernels

# Kernels

## Definition

Given a non-empty set $\mathcal{X}$, a symmetric real-valued (resp. Hermitian complex valued) function $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (resp $f : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$) is called a kernel.

- All notions of (symmetric) distances, similarities are kernels

# Kernels

## Definition

Given a non-empty set $\mathcal{X}$, a symmetric real-valued (resp. Hermitian complex valued) function $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (resp $f : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$) is called a kernel.

- All notions of (symmetric) distances, similarities are kernels
- Alternatively kernels can be thought of as measures of similarity or distance

# Definiteness

## Definition

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} \neq \mathbf{0}$, $\mathbf{c}^\top A\mathbf{c} > 0$.

# Definiteness

## Definition

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} \neq \mathbf{0}$, $\mathbf{c}^\top A\mathbf{c} > 0$.

## Definition

A kernel $K$ defined on a domain $\mathcal{X}$ is said to be positive definite if $\forall n \in \mathbb{N}$, $\forall x_1, \ldots x_n \in \mathcal{X}$, the matrix $G = (G_{ij}) = (K(x_i, x_j))$ is positive definite. Alternatively, for every $g \in L_2(\mathcal{X})$, $\iint_{\mathcal{X}} g(x)g(x')K(x, x') \geq 0$.

# Definiteness

### Definition

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} \neq \mathbf{0}$, $\mathbf{c}^\top A\mathbf{c} > 0$.

### Definition

A kernel $K$ defined on a domain $\mathcal{X}$ is said to be positive definite if $\forall n \in \mathbb{N}$, $\forall x_1, \ldots x_n \in \mathcal{X}$, the matrix $G = (G_{ij}) = (K(x_i, x_j))$ is positive definite. Alternatively, for every $g \in L_2(\mathcal{X})$, $\iint_{\mathcal{X}} g(x)g(x')K(x, x') \geq 0$.

### Definition

A kernel $K$ is said to be indefinite if it is neither positive definite nor negative definite.

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay
- ...

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay
- ...
- Testing the Mercer condition difficult

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay
- ...
- Testing the Mercer condition difficult
- Indefinite kernels known to give good performance

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay
- ...
- Testing the Mercer condition difficult
- Indefinite kernels known to give good performance
- Ability to use indefinite kernels increases the scope of learning-the-kernel algorithms

# The Kernel Trick

- All PD Kernels turn out to be inner products in some Hilbert space
- Thus, any algorithm that only takes as input pairwise inner products can be made to implicitly work in such spaces
- Results known as Representer Theorems keep any Curses of dimensionality at bay
- ...
- Testing the Mercer condition difficult
- Indefinite kernels known to give good performance
- Ability to use indefinite kernels increases the scope of learning-the-kernel algorithms
- Learning paradigm somewhere between PAC and weak$^*$-PAC

# Kernels as distances

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^{n}, x_i \in X, y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^{n}, x_i \in X, y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$
- Classify a new point $x$ as $+$ if $d(x, T^+) < d(x, T^-)$ otherwise as $-$

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^n, x_i \in X, y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$
- Classify a new point $x$ as $+$ if $d(x, T^+) < d(x, T^-)$ otherwise as $-$
- When will this work ?

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^n, x_i \in X, y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$
- Classify a new point $x$ as $+$ if $d(x, T^+) < d(x, T^-)$ otherwise as $-$
- When will this work ?
  - Intuitively when a large fraction of domain points are closer (according to $d$) to points of the same label than points of the different label

# Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^n, x_i \in X, y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$
- Classify a new point $x$ as $+$ if $d(x, T^+) < d(x, T^-)$ otherwise as $-$
- When will this work ?
  - Intuitively when a large fraction of domain points are closer (according to $d$) to points of the same label than points of the different label
  - $\Pr\limits_{x \in_R \mu}\left[d(x, \mathcal{X}^{t(x)}) < d(x, \mathcal{X}^{\overline{t(x)}})\right] \geq 1 - \epsilon$

# What is a *good* distance function

## Definition

A distance function $d$ is said to be strongly $(\epsilon, \gamma)$-good for a learning problem, if at least $1 - \epsilon$ probability mass of examples $x \in \mu$ satisfy

$$\Pr_{x, x'' \in_R \mu} \left[ d(x, x') < d(x, x'') | x' \in \mathcal{X}^{t(x)}, x'' \in \mathcal{X}^{\overline{t(x)}} \right] \geq \frac{1}{2} + \gamma.$$

- A smoothed version of the earlier intuitive notion of good distance function

# What is a *good* distance function

## Definition

A distance function $d$ is said to be strongly $(\epsilon, \gamma)$-good for a learning problem, if at least $1 - \epsilon$ probability mass of examples $x \in \mu$ satisfy

$$\Pr_{x,x'' \in_R \mu}\left[d(x,x') < d(x,x'') | x' \in \mathcal{X}^{t(x)}, x'' \in \mathcal{X}^{\overline{t(x)}}\right] \geq \frac{1}{2} + \gamma.$$

- A smoothed version of the earlier intuitive notion of good distance function
- Correspondingly the algorithm is also a smoothed version of the classical NN algorithm

# Learning with a good distance function

## Theorem ([Wang et al., 2007])

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{1}{\gamma^2} lg\left(\frac{1}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n, a_i \in_R \mu^+, b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} sgn\left[d(x, b_i) - d(x, a_i)\right]$$

- What about the NN algorithm - any guarantees for that ?

# Learning with a good distance function

## Theorem ([Wang et al., 2007])

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{1}{\gamma^2} lg\left(\frac{1}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in_R \mu^+$, $b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} sgn\left[d(x, b_i) - d(x, a_i)\right]$$

- What about the NN algorithm - any guarantees for that ?
- For metric distances - in a few slides

# Learning with a good distance function

## Theorem ([Wang et al., 2007])

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{1}{\gamma^2} lg\left(\frac{1}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n, a_i \in_R \mu^+, b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} sgn\left[d(x, b_i) - d(x, a_i)\right]$$

- What about the NN algorithm - any guarantees for that ?
- For metric distances - in a few slides
- Note that this is an instance of weak$^*$-PAC learning

# Learning with a good distance function

## Theorem ([Wang et al., 2007])

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier $h$, for any $\epsilon, \delta > 0$, when given $n = \frac{1}{\gamma^2} lg\left(\frac{1}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in_R \mu^+, b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} sgn\left[d(x, b_i) - d(x, a_i)\right]$$

- What about the NN algorithm - any guarantees for that ?
- For metric distances - in a few slides
- Note that this is an instance of weak$^*$-PAC learning
- Guarantees for NN on non-metric distances ?

# Other landmarking approaches

- [Weinshall et al., 1998], [Jacobs et al., 2000] investigate algorithms where a (set of) representative(s) is chosen for each label: eg the centroid of all training points with that label

# Other landmarking approaches

- [Weinshall et al., 1998], [Jacobs et al., 2000] investigate algorithms where a (set of) representative(s) is chosen for each label: eg the centroid of all training points with that label
- [Pękalska and Duin, 2001] consider combining classifiers based on different dissimilarity functions as well as building classifiers on combinations of different dissimilarity functions

# Other landmarking approaches

- [Weinshall et al., 1998], [Jacobs et al., 2000] investigate algorithms where a (set of) representative(s) is chosen for each label: eg the centroid of all training points with that label
- [Pękalska and Duin, 2001] consider combining classifiers based on different dissimilarity functions as well as building classifiers on combinations of different dissimilarity functions
- [Weinberger and Saul, 2009] propose methods to learn a Mahalanobis distance to improve NN classification

# Other landmarking approaches

- [Gottlieb et al., 2010] present efficient schemes for NN classifiers (Lipschitz extension classifiers) in doubling spaces

$$h(x) = \mathrm{sgn}\left[f(x)\right], f(x) = \min_{x_i \in T}\left(t(x_i) + 2\frac{d(x, x_i)}{d(T^+, T^-)}\right)$$

# Other landmarking approaches

- [Gottlieb et al., 2010] present efficient schemes for NN classifiers (Lipschitz extension classifiers) in doubling spaces

$$h(x) = \mathrm{sgn}\left[f(x)\right], f(x) = \min_{x_i \in T}\left(t(x_i) + 2\frac{d(x, x_i)}{d(T^+, T^-)}\right)$$

  - make use of approximate nearest neighbor search algorithms

# Other landmarking approaches

- [Gottlieb et al., 2010] present efficient schemes for NN classifiers (Lipschitz extension classifiers) in doubling spaces

$$h(x) = \operatorname{sgn}\left[f(x)\right], f(x) = \min_{x_i \in T}\left(t(x_i) + 2\frac{d(x, x_i)}{d(T^+, T^-)}\right)$$

  - make use of approximate nearest neighbor search algorithms
  - show that pseudo dimension of Lipschitz classifiers in doubling spaces is bounded

# Other landmarking approaches

- [Gottlieb et al., 2010] present efficient schemes for NN classifiers (Lipschitz extension classifiers) in doubling spaces

$$h(x) = \mathrm{sgn}\left[f(x)\right], f(x) = \min_{x_i \in T}\left(t(x_i) + 2\frac{d(x, x_i)}{d(T^+, T^-)}\right)$$

  - make use of approximate nearest neighbor search algorithms
  - show that pseudo dimension of Lipschitz classifiers in doubling spaces is bounded
  - are able to provides schemes for optimizing the bias-variance trade-off

# Data sensitive embeddings

- Landmarking based approaches can be seen as implicitly embedding the domain into an $n$ dimensional feature space

# Data sensitive embeddings

- Landmarking based approaches can be seen as implicitly embedding the domain into an $n$ dimensional feature space
- Perform an explicit embedding of training data to some vector space that is isometric and learn a classifier

# Data sensitive embeddings

- Landmarking based approaches can be seen as implicitly embedding the domain into an $n$ dimensional feature space
- Perform an explicit embedding of training data to some vector space that is isometric and learn a classifier
- Perform (approximately) isometric embeddings of test data into the same vector space to classify them

# Data sensitive embeddings

- Landmarking based approaches can be seen as implicitly embedding the domain into an $n$ dimensional feature space
- Perform an explicit embedding of training data to some vector space that is isometric and learn a classifier
- Perform (approximately) isometric embeddings of test data into the same vector space to classify them
- Exact for transductive problems, approximate for inductive ones

# Data sensitive embeddings

- Landmarking based approaches can be seen as implicitly embedding the domain into an $n$ dimensional feature space
- Perform an explicit embedding of training data to some vector space that is isometric and learn a classifier
- Perform (approximately) isometric embeddings of test data into the same vector space to classify them
- Exact for transductive problems, approximate for inductive ones
- Long history of such techniques from early AI - Multidimensional scaling

# The Minkowski space-time

## Definition

$\mathbb{R}^4 = \mathbb{R}^3 \oplus \mathbb{R}^1 := \mathbb{R}^{(3,1)}$ endowed with the inner product $\langle (x_1, y_1, z_1, t_1), (x_2, y_2, z_2, t_2) \rangle = x_1 x_2 + y_1 y_2 + z_1 z_2 - t_1 t_2$ is a 4-dimensional Minkowski space with signature $(3, 1)$. The norm imposed by this inner product is $\|(x_1, y_1, z_1, t_1)\|^2 = x_1^2 + y_1^2 + z_1^2 - t_1^2$

- Can have vectors of negative length due to the imaginary time coordinate

# The Minkowski space-time

## Definition

$\mathbb{R}^4 = \mathbb{R}^3 \oplus \mathbb{R}^1 := \mathbb{R}^{(3,1)}$ endowed with the inner product $\langle (x_1, y_1, z_1, t_1), (x_2, y_2, z_2, t_2) \rangle = x_1 x_2 + y_1 y_2 + z_1 z_2 - t_1 t_2$ is a 4-dimensional Minkowski space with signature $(3, 1)$. The norm imposed by this inner product is $\|(x_1, y_1, z_1, t_1)\|^2 = x_1^2 + y_1^2 + z_1^2 - t_1^2$

- Can have vectors of negative length due to the imaginary time coordinate
- The definition an be extended to arbitrary $\mathbb{R}^{(p,q)}$ (PE Spaces)

# The Minkowski space-time

## Definition

$\mathbb{R}^4 = \mathbb{R}^3 \oplus \mathbb{R}^1 := \mathbb{R}^{(3,1)}$ endowed with the inner product $\langle (x_1, y_1, z_1, t_1), (x_2, y_2, z_2, t_2) \rangle = x_1 x_2 + y_1 y_2 + z_1 z_2 - t_1 t_2$ is a 4-dimensional Minkowski space with signature $(3, 1)$. The norm imposed by this inner product is $\|(x_1, y_1, z_1, t_1)\|^2 = x_1^2 + y_1^2 + z_1^2 - t_1^2$

- Can have vectors of negative length due to the imaginary time coordinate
- The definition an be extended to arbitrary $\mathbb{R}^{(p,q)}$ (PE Spaces)

## Theorem ([Goldfarb, 1984], [Haasdonk, 2005])

*Any finite pseudo metric $(\mathcal{X}, d), |\mathcal{X}| = n$ can be isometrically embedded in $\left(\mathbb{R}^{(p,q)}, \|\cdot\|^2\right)$ for some values of $p + q < n$.*

# The Embedding

## Embedding the training set

Given a distance matrix $\mathbb{R}^{n\times n} \ni D = (d(x_i, x_j))$, find the corresponding inner products in the PE space as $G = -\frac{1}{2}JDJ$ where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Do an eigendecomposition of $B = Q\Lambda Q^\top = Q|\Lambda|^{\frac{1}{2}}M|\Lambda|^{\frac{1}{2}}Q^\top$ where $M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix}$. The representation of the points is $X = Q|\Lambda|^{\frac{1}{2}}$

## Embedding a new point

Perform a linear projection into the space found above. Given $d = (d(x, x_i))$, the vector of distances to the old points, the inner products to all the old points is found as $g = -\frac{1}{2}\left(d - \frac{1}{n}\mathbf{1}\mathbf{1}^\top D\right)J$. Now find the mean square error solution to $xMX^\top = b$ as $x = bX|\Lambda|^{-1}M$.

# Classification in PE spaces

- Earliest observations by [Goldfarb, 1984] who realized the link between landmarking and embedding approaches

# Classification in PE spaces

- Earliest observations by [Goldfarb, 1984] who realized the link between landmarking and embedding approaches
- [Pękalska and Duin, 2000],[Pękalska et al., 2001], [Pękalska and Duin, 2002] use this space to learn SVM, LPM, Quadratic Discriminant and Fisher Linear Discriminant classifiers

# Classification in PE spaces

- Earliest observations by [Goldfarb, 1984] who realized the link between landmarking and embedding approaches
- [Pękalska and Duin, 2000],[Pękalska et al., 2001], [Pękalska and Duin, 2002] use this space to learn SVM, LPM, Quadratic Discriminant and Fisher Linear Discriminant classifiers
- [Harol et al., 2006] propose enlarging the PE space to allow for lesser distortion in embeddings test points

# Classification in PE spaces

- Earliest observations by [Goldfarb, 1984] who realized the link between landmarking and embedding approaches
- [Pękalska and Duin, 2000],[Pękalska et al., 2001], [Pękalska and Duin, 2002] use this space to learn SVM, LPM, Quadratic Discriminant and Fisher Linear Discriminant classifiers
- [Harol et al., 2006] propose enlarging the PE space to allow for lesser distortion in embeddings test points
- [Duin and Pękalska, 2008] propose refinements to the distance measure by making modifications to the PE space allowing for better NN classification

# Classification in PE spaces

- Earliest observations by [Goldfarb, 1984] who realized the link between landmarking and embedding approaches
- [Pękalska and Duin, 2000],[Pękalska et al., 2001], [Pękalska and Duin, 2002] use this space to learn SVM, LPM, Quadratic Discriminant and Fisher Linear Discriminant classifiers
- [Harol et al., 2006] propose enlarging the PE space to allow for lesser distortion in embeddings test points
- [Duin and Pękalska, 2008] propose refinements to the distance measure by making modifications to the PE space allowing for better NN classification
- Guarantees for classifiers learned in PE spaces ?

# Data insensitive embeddings

- Possible if the distance measure can be isometrically embedded into some space

# Data insensitive embeddings

- Possible if the distance measure can be isometrically embedded into some space
- Learn a simple classifier there and interpret it in terms of the distance measure

# Data insensitive embeddings

- Possible if the distance measure can be isometrically embedded into some space
- Learn a simple classifier there and interpret it in terms of the distance measure
- Require algorithms that can work without explicit embeddings

# Data insensitive embeddings

- Possible if the distance measure can be isometrically embedded into some space
- Learn a simple classifier there and interpret it in terms of the distance measure
- Require algorithms that can work without explicit embeddings
- Exact for transductive as well as inductive problems

# Data insensitive embeddings

- Possible if the distance measure can be isometrically embedded into some space
- Learn a simple classifier there and interpret it in terms of the distance measure
- Require algorithms that can work without explicit embeddings
- Exact for transductive as well as inductive problems
- Recent interest due to advent of large margin classifiers

# Normed Spaces

## Definition

Given a vector space $V$ over a field $F \subseteq \mathbb{C}$, a norm is a function $\|\cdot\| : V \to \mathbb{R}$ such that $\forall \mathbf{u}, \mathbf{v} \in V, a \in F$, $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$, $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$. A vector space that is complete with respect to a norm is called a Banach space.

# Normed Spaces

## Definition

Given a vector space $V$ over a field $F \subseteq \mathbb{C}$, a norm is a function $\|\cdot\| : V \to \mathbb{R}$ such that $\forall \mathbf{u}, \mathbf{v} \in V, a \in F$, $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$, $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$. A vector space that is complete with respect to a norm is called a Banach space.

## Theorem ([von Luxburg and Bousquet, 2004])

*Given a metric space $\mathcal{M} = (\mathcal{X}, d)$ and the space of all Lipschitz functions $Lip(\mathcal{X})$ defined on $\mathcal{M}$, there exists a Banach Space $\mathcal{B}$ and maps $\Phi : \mathcal{X} \to \mathcal{B}$ and $\Psi : Lip(\mathcal{X}) \to \mathcal{B}'$, the operator norm on $\mathcal{B}'$ giving the Lipschitz constant for each function $f \in Lip(\mathcal{X})$ such that both can be realized simultaneously as isomorphic isometries.*

# Normed Spaces

## Definition

Given a vector space $V$ over a field $F \subseteq \mathbb{C}$, a norm is a function $\|\cdot\| : V \rightarrow \mathbb{R}$ such that $\forall \mathbf{u}, \mathbf{v} \in V, a \in F$, $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$, $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$. A vector space that is complete with respect to a norm is called a Banach space.

## Theorem ([von Luxburg and Bousquet, 2004])

*Given a metric space $\mathcal{M} = (\mathcal{X}, d)$ and the space of all Lipschitz functions $Lip(\mathcal{X})$ defined on $\mathcal{M}$, there exists a Banach Space $\mathcal{B}$ and maps $\Phi : \mathcal{X} \rightarrow \mathcal{B}$ and $\Psi : Lip(\mathcal{X}) \rightarrow \mathcal{B}'$, the operator norm on $\mathcal{B}'$ giving the Lipschitz constant for each function $f \in Lip(\mathcal{X})$ such that both can be realized simultaneously as isomorphic isometries.*

- The Kuratowski embedding gives a constructive proof

# Classification in Banach spaces

- [von Luxburg and Bousquet, 2004] proposes large margin classification schemes on Banach spaces relying on Convex hull interpretations of SVM classifiers

# Classification in Banach spaces

- [von Luxburg and Bousquet, 2004] proposes large margin classification schemes on Banach spaces relying on Convex hull interpretations of SVM classifiers

$$\inf_{p^+ \in C^+, p^- \in C^-} \|p^+ - p^-\| \tag{1}$$

# Classification in Banach spaces

- [von Luxburg and Bousquet, 2004] proposes large margin classification schemes on Banach spaces relying on Convex hull interpretations of SVM classifiers

$$\inf_{p^+ \in C^+, p^- \in C^-} \|p^+ - p^-\| \tag{1}$$

$$\sup_{t \in \mathcal{B}'} \inf_{p^+ \in C^+, p^- \in C^-} \frac{\langle T, p^+ - p^- \rangle}{\|T\|} \tag{2}$$

# Classification in Banach spaces

- [von Luxburg and Bousquet, 2004] proposes large margin classification schemes on Banach spaces relying on Convex hull interpretations of SVM classifiers

$$\inf_{p^+ \in C^+, p^- \in C^-} \|p^+ - p^-\| \tag{1}$$

$$\sup_{t \in \mathcal{B}'} \inf_{p^+ \in C^+, p^- \in C^-} \frac{\langle T, p^+ - p^- \rangle}{\|T\|} \tag{2}$$

$$\begin{aligned} \inf_{T \in \mathcal{B}', b \in \mathbb{R}} \quad & \|T\| = L(T) \\ \text{subject to} \quad & t(x_i)\left(\langle T, x_i \rangle + b\right) \geq 1, \forall i = 1, \ldots, n. \end{aligned} \tag{3}$$

# Classification in Banach spaces

- [von Luxburg and Bousquet, 2004] proposes large margin classification schemes on Banach spaces relying on Convex hull interpretations of SVM classifiers

$$\inf_{p^+ \in C^+, p^- \in C^-} \|p^+ - p^-\| \tag{1}$$

$$\sup_{t \in \mathcal{B}'} \inf_{p^+ \in C^+, p^- \in C^-} \frac{\langle T, p^+ - p^- \rangle}{\|T\|} \tag{2}$$

$$\begin{aligned} \inf_{T \in \mathcal{B}', b \in \mathbb{R}} \quad & \|T\| = L(T) \\ \text{subject to} \quad & t(x_i)\left(\langle T, x_i \rangle + b\right) \geq 1, \forall i = 1, \ldots, n. \end{aligned} \tag{3}$$

$$\begin{aligned} \inf_{T \in \mathcal{B}', b \in \mathbb{R}} \quad & L(T) + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & t(x_i)\left(\langle T, x_i \rangle + b\right) \geq 1 - \xi_i, \xi \geq 0 \forall i = 1, \ldots, n. \end{aligned} \tag{4}$$

# Representer Theorems

- Lets us escape the curse of dimensionality

**Theorem (Lipschitz extension)**

*Given a Lipschitz function $f$ defined on a finite subset $X \subset \mathcal{X}$, one can extend $f$ to $f'$ on the entire domain such that $Lip(f') = Lip(f)$.*

# Representer Theorems

- Lets us escape the curse of dimensionality

**Theorem (Lipschitz extension)**

*Given a Lipschitz function $f$ defined on a finite subset $X \subset \mathcal{X}$, one can extend $f$ to $f'$ on the entire domain such that $Lip(f') = Lip(f)$.*

- Solution to Program 3 is always of the form

$$f(x) = \frac{d(x, T^-) - d(x, T^+)}{d(T^+, T^-)}$$

# Representer Theorems

- Lets us escape the curse of dimensionality

**Theorem (Lipschitz extension)**

*Given a Lipschitz function $f$ defined on a finite subset $X \subset \mathcal{X}$, one can extend $f$ to $f'$ on the entire domain such that $Lip(f') = Lip(f)$.*

- Solution to Program 3 is always of the form

$$f(x) = \frac{d(x, T^-) - d(x, T^+)}{d(T^+, T^-)}$$

- Solution to Program 4 is always of the form

$$g(x) = \alpha \min_i \left(t(x_i) + L_0 d(x, x_i)\right) + (1 - \alpha) \max_i \left(t(x_i) - L_0 d(x, x_i)\right)$$

# But ...

- Not a representer theorem involving distances to individual training points

# But ...

- Not a representer theorem involving distances to individual training points
- Shown not to exist in certain cases - but the examples don't seem natural

# But ...

- Not a representer theorem involving distances to individual training points
- Shown not to exist in certain cases - but the examples don't seem natural
- By restricting oneself to different subspaces of Lip($\mathcal{X}$) one recovers the SVM, LPM and NN algorithms

# But ...

- Not a representer theorem involving distances to individual training points
- Shown not to exist in certain cases - but the examples don't seem natural
- By restricting oneself to different subspaces of Lip($\mathcal{X}$) one recovers the SVM, LPM and NN algorithms
- Can one use bi-Lipschitz embeddings instead ?

# But ...

- Not a representer theorem involving distances to individual training points
- Shown not to exist in certain cases - but the examples don't seem natural
- By restricting oneself to different subspaces of Lip($\mathcal{X}$) one recovers the SVM, LPM and NN algorithms
- Can one use bi-Lipschitz embeddings instead ?
- Can one define "distance kernels" that allow one to restrict oneself to specific subspaces of Lip($\mathcal{X}$)

# Other Banach Space Approaches

- [Hein et al., 2005] consider low distortion embeddings into Hilbert spaces giving a re-derivation of the SVM algorithm

# Other Banach Space Approaches

- [Hein et al., 2005] consider low distortion embeddings into Hilbert spaces giving a re-derivation of the SVM algorithm

**Definition**

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be conditionally positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c}^\top \mathbf{1} = 0$, $\mathbf{c}^\top A \mathbf{c} > 0$.

# Other Banach Space Approaches

- [Hein et al., 2005] consider low distortion embeddings into Hilbert spaces giving a re-derivation of the SVM algorithm

**Definition**

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be conditionally positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c}^\top \mathbf{1} = 0$, $\mathbf{c}^\top A\mathbf{c} > 0$.

**Definition**

A kernel $K$ defined on a domain $\mathcal{X}$ is said to be conditionally positive definite if $\forall n \in \mathbb{N}$, $\forall x_1, \ldots x_n \in \mathcal{X}$, the matrix $G = (G_{ij}) = (K(x_i, x_j))$ is conditionally positive definite.

# Other Banach Space Approaches

- [Hein et al., 2005] consider low distortion embeddings into Hilbert spaces giving a re-derivation of the SVM algorithm

**Definition**

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be conditionally positive definite if $\forall \mathbf{c} \in \mathbb{R}^n$, $\mathbf{c}^\top \mathbf{1} = 0$, $\mathbf{c}^\top A \mathbf{c} > 0$.

**Definition**

A kernel $K$ defined on a domain $\mathcal{X}$ is said to be conditionally positive definite if $\forall n \in \mathbb{N}, \forall x_1, \ldots x_n \in \mathcal{X}$, the matrix $G = (G_{ij}) = (K(x_i, x_j))$ is conditionally positive definite.

**Theorem**

*A metric $d$ is Hibertian if it can be isometrically embedded into a Hilbert space iff $-d^2$ is conditionally positive definite*

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics
  - Lack of symmetry and bi-linearity for semi inner products prevents such kernel tricks for general metrics

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics
  - Lack of symmetry and bi-linearity for semi inner products prevents such kernel tricks for general metrics
- [Zhang et al., 2009] propose Reproducing Kernel Banach Spaces akin to RKHS that admit kernel tricks

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics
  - Lack of symmetry and bi-linearity for semi inner products prevents such kernel tricks for general metrics
- [Zhang et al., 2009] propose Reproducing Kernel Banach Spaces akin to RKHS that admit kernel tricks
  - Use a bilinear form on $\mathcal{B} \times \mathcal{B}'$ instead of $\mathcal{B} \times \mathcal{B}$

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics
  - Lack of symmetry and bi-linearity for semi inner products prevents such kernel tricks for general metrics
- [Zhang et al., 2009] propose Reproducing Kernel Banach Spaces akin to RKHS that admit kernel tricks
  - Use a bilinear form on $\mathcal{B} \times \mathcal{B}'$ instead of $\mathcal{B} \times \mathcal{B}$
  - No succinct characterizations of what can yield an RKBS

# Other Banach Space Approaches

- [Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations
  - Aim for a kernel trick for general metrics
  - Lack of symmetry and bi-linearity for semi inner products prevents such kernel tricks for general metrics
- [Zhang et al., 2009] propose Reproducing Kernel Banach Spaces akin to RKHS that admit kernel tricks
  - Use a bilinear form on $\mathcal{B} \times \mathcal{B}'$ instead of $\mathcal{B} \times \mathcal{B}$
  - No succinct characterizations of what can yield an RKBS
  - For finite domains, any kernel is a reproducing kernel for some RKBS (trivial)

# Kernel Trick for Distances ?

**Theorem ([Schölkopf, 2000])**

*A kernel $C$ defined on some domain $\mathcal{X}$ is CPD iff for some fixed $x_0 \in \mathcal{X}$, the kernel $K(x, x') = C(x, x') - C(x, x_0) - C(x', x_0)$ is PD. Such a $C$ is also a Hilbertian metric.*

- The SVM algorithm is incapable of distinguishing between $C$ and $K$ [Boughorbel et al., 2005]

# Kernel Trick for Distances ?

**Theorem ([Schölkopf, 2000])**

*A kernel $C$ defined on some domain $\mathcal{X}$ is CPD iff for some fixed $x_0 \in \mathcal{X}$, the kernel $K(x, x') = C(x, x') - C(x, x_0) - C(x', x_0)$ is PD. Such a $C$ is also a Hilbertian metric.*

- The SVM algorithm is incapable of distinguishing between $C$ and $K$ [Boughorbel et al., 2005]
- $\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) = \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j C(x_i, x_j)$ subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$

# Kernel Trick for Distances ?

**Theorem ([Schölkopf, 2000])**

*A kernel $C$ defined on some domain $\mathcal{X}$ is CPD iff for some fixed $x_0 \in \mathcal{X}$, the kernel $K(x,x') = C(x,x') - C(x,x_0) - C(x',x_0)$ is PD. Such a $C$ is also a Hilbertian metric.*

- The SVM algorithm is incapable of distinguishing between $C$ and $K$ [Boughorbel et al., 2005]
- $\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K(x_i,x_j) = \sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j C(x_i,x_j)$ subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$
- What about higher order CPD kernels - their characterization ?

# Kernels as similarity

# The Kernel Trick

- Mercer's theorem tells us that a similarity space $(\mathcal{X}, K)$ is embeddable in a Hilbert space iff $K$ is a PSD kernel

# The Kernel Trick

- Mercer's theorem tells us that a similarity space $(\mathcal{X}, K)$ is embeddable in a Hilbert space iff $K$ is a PSD kernel
- Quite similar to what we had for Banach spaces only with more structure now

# The Kernel Trick

- Mercer's theorem tells us that a similarity space $(\mathcal{X}, K)$ is embeddable in a Hilbert space iff $K$ is a PSD kernel
- Quite similar to what we had for Banach spaces only with more structure now
- Can formulate large margin classifiers as before

# The Kernel Trick

- Mercer's theorem tells us that a similarity space $(\mathcal{X}, K)$ is embeddable in a Hilbert space iff $K$ is a PSD kernel
- Quite similar to what we had for Banach spaces only with more structure now
- Can formulate large margin classifiers as before
- Representer Theorem [Schölkopf and Smola, 2001] : solution of the form $f(x) = \sum_{i=1}^{n} K(x, x_i)$

# The Kernel Trick

- Mercer's theorem tells us that a similarity space $(\mathcal{X}, K)$ is embeddable in a Hilbert space iff $K$ is a PSD kernel
- Quite similar to what we had for Banach spaces only with more structure now
- Can formulate large margin classifiers as before
- Representer Theorem [Schölkopf and Smola, 2001] : solution of the form $f(x) = \sum\limits_{i=1}^{n} K(x, x_i)$
- Generalization Guarantees : method of Rademacher Averages [Mendelson, 2003]

# The Lazy approaches

- Why bother building a theory when one already exists !

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip
  - [Luss and d'Aspremont, 2007] folds this process into the SVM algorithm by treating an indefinite kernel as a noisy version of a Mercer kernel

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip
  - [Luss and d'Aspremont, 2007] folds this process into the SVM algorithm by treating an indefinite kernel as a noisy version of a Mercer kernel
  - Tries to handle test points consistently but no theoretical justification of the process

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip
  - [Luss and d'Aspremont, 2007] folds this process into the SVM algorithm by treating an indefinite kernel as a noisy version of a Mercer kernel
  - Tries to handle test points consistently but no theoretical justification of the process
  - Mercer kernels are not dense in the space of symmetric kernels

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip
  - [Luss and d'Aspremont, 2007] folds this process into the SVM algorithm by treating an indefinite kernel as a noisy version of a Mercer kernel
  - Tries to handle test points consistently but no theoretical justification of the process
  - Mercer kernels are not dense in the space of symmetric kernels
- [Haasdonk and Bahlmann, 2004] propose distance substitution kernels : substituting distance/similarity measures into kernels of the form $K(\|\mathbf{x}-\mathbf{y}\|), K(\langle\mathbf{x},\mathbf{y}\rangle)$

# The Lazy approaches

- Why bother building a theory when one already exists !
  - Use a PD approximation to the given indefinite kernel !!
- [Chen et al., 2009] Spectrum Shift, Spectrum Clip, Spectrum Flip
  - [Luss and d'Aspremont, 2007] folds this process into the SVM algorithm by treating an indefinite kernel as a noisy version of a Mercer kernel
  - Tries to handle test points consistently but no theoretical justification of the process
  - Mercer kernels are not dense in the space of symmetric kernels
- [Haasdonk and Bahlmann, 2004] propose distance substitution kernels : substituting distance/similarity measures into kernels of the form $K(\|\mathbf{x} - \mathbf{y}\|), K(\langle \mathbf{x}, \mathbf{y} \rangle)$
  - These yield PD kernels iff the distance measure is Hilbertian

# Working with Indefinite Similarities

- Embed Training sets into PE spaces (Minkowski spaces) as before

# Working with Indefinite Similarities

- Embed Training sets into PE spaces (Minkowski spaces) as before
- [Graepel et al., 1998] proposes to learn SVMs in this space - unfortunately not a large margin formulation

# Working with Indefinite Similarities

- Embed Training sets into PE spaces (Minkowski spaces) as before
- [Graepel et al., 1998] proposes to learn SVMs in this space - unfortunately not a large margin formulation
- [Graepel et al., 1999] propose LP machines in a $\nu$-SVM like formulation to obtain sparse classifiers

# Working with Indefinite Similarities

- Embed Training sets into PE spaces (Minkowski spaces) as before
- [Graepel et al., 1998] proposes to learn SVMs in this space - unfortunately not a large margin formulation
- [Graepel et al., 1999] propose LP machines in a $\nu$-SVM like formulation to obtain sparse classifiers
- [Mierswa, 2006] proposes using evolutionary algorithms to solve non-convex formulations involving indefinite kernels

# Working with Indefinite Similarities

- [Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there

# Working with Indefinite Similarities

- [Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there
- Not a margin maximization formulation

# Working with Indefinite Similarities

- [Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there
- Not a margin maximization formulation
- New points are not embedded into this space - rather the SVM like representation is used (without justification)

# Working with Indefinite Similarities

- [Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there
- Not a margin maximization formulation
- New points are not embedded into this space - rather the SVM like representation is used (without justification)
- Optimization not possible since program formulations are non-convex - stabilization used

# Working with Indefinite Similarities

- [Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there
- Not a margin maximization formulation
- New points are not embedded into this space - rather the SVM like representation is used (without justification)
- Optimization not possible since program formulations are non-convex - stabilization used
- Can any guarantees be given for this formulation ?

# Kreĭn spaces

## Definition

An inner product space $(\mathcal{K}, \langle , \rangle_{\mathcal{K}})$ is called a Kreĭn space if there exist two Hilbert spaces $\mathcal{H}_+$ and $H_-$ such $K = \mathcal{H}_+ \oplus \mathcal{H}_-$ and $\forall f, g \in \mathcal{K}$, $\langle f, g \rangle_{\mathcal{K}} = \langle f, g \rangle_{\mathcal{H}_+} - \langle f, g \rangle_{\mathcal{H}_-}$.

## Definition

Given a domain $\mathcal{X}$, a subset $\mathcal{K} \subset \mathbb{R}^{\mathcal{X}}$ is called a Reproducing Kernel Kreĭn space if the evaluation functional $T_x : f \mapsto f(x)$ is continuous on $\mathcal{K}$ with respect to its strong topology.

## Theorem ([Ong et al., 2004])

*A kernel $K$ on $\mathcal{X}$ is a reproducing kernel for some Kreĭn space $\mathcal{K}$ iff there exist PD kernels $K_+$ and $K_-$ such that $K = K_+ - K_-$.*

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers
- Prove that even stabilization leads to an SVM-like Representer Theorem

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers
- Prove that even stabilization leads to an SVM-like Representer Theorem
- No large margin formulations considered due to singularity issues

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers
- Prove that even stabilization leads to an SVM-like Representer Theorem
- No large margin formulations considered due to singularity issues
  - Instead regularization is performed by truncating the spectrum of $K$

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers
- Prove that even stabilization leads to an SVM-like Representer Theorem
- No large margin formulations considered due to singularity issues
  - Instead regularization is performed by truncating the spectrum of $K$
  - Iterative methods to minimize squared error lead to regularizations

# Classification in Kreĭn spaces

- [Ong et al., 2004] proves all the necessary results for learning large margin classifiers
- Prove that even stabilization leads to an SVM-like Representer Theorem
- No large margin formulations considered due to singularity issues
  - Instead regularization is performed by truncating the spectrum of $K$
  - Iterative methods to minimize squared error lead to regularizations
- Proves generalization error bounds using method of Rademacher averages

# Landmarking approaches

- [Graepel et al., 1999] consider landmarking with indefinite kernels

# Landmarking approaches

- [Graepel et al., 1999] consider landmarking with indefinite kernels
- Perform $L_1$ regularization for large margin classifier to obtain sparse solutions - yields an LP formulation

# Landmarking approaches

- [Graepel et al., 1999] consider landmarking with indefinite kernels
- Perform $L_1$ regularization for large margin classifier to obtain sparse solutions - yields an LP formulation
- Also propose the $\nu$-SVM formulation to get control over number of margin violations

# Landmarking approaches

- [Graepel et al., 1999] consider landmarking with indefinite kernels
- Perform $L_1$ regularization for large margin classifier to obtain sparse solutions - yields an LP formulation
- Also propose the $\nu$-SVM formulation to get control over number of margin violations
- Allows us to perform optimizations in the bias-variance trade-off

# Landmarking approaches

- [Graepel et al., 1999] consider landmarking with indefinite kernels
- Perform $L_1$ regularization for large margin classifier to obtain sparse solutions - yields an LP formulation
- Also propose the $\nu$-SVM formulation to get control over number of margin violations
- Allows us to perform optimizations in the bias-variance trade-off
- However no guarantees given - were provided later by [Hein et al., 2005], [von Luxburg and Bousquet, 2004]

# What is a *good* similarity function

## Definition

A kernel function $K$ is said to be $(\epsilon, \gamma)$-kernel good for a learning problem, if $\exists \beta \in \mathcal{K}_K$

$$\Pr_{x \in_R \mu}\left[t(x)(\langle \beta, \Phi_K(x) \rangle > \gamma)\right] \geq 1 - \epsilon.$$

## Definition

A kernel function $K$ is said to be strongly $(\epsilon, \gamma)$-good for a learning problem, if at least a $1 - \epsilon$ probablity mass of the domain satisfies

$$\mathop{\mathbb{E}}_{x' \in_R \mu^+}\left[K(x, x')\right] > \mathop{\mathbb{E}}_{x' \in_R \mu^-}\left[K(x, x')\right] + \gamma$$

# Learning with a good distance function

**Theorem ([Balcan et al., 2008a])**

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{16}{\gamma^2} lg\left(\frac{2}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in_R \mu^+, b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^n K(x, a_i) - \frac{1}{n}\sum_{i=1}^n K(x, b_i)$$

- Have to introduce a weighing function to extend scope of the algorithm

# Learning with a good distance function

**Theorem ([Balcan et al., 2008a])**

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{16}{\gamma^2} lg\left(\frac{2}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in_R \mu^+$, $b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} K(x, a_i) - \frac{1}{n}\sum_{i=1}^{n} K(x, b_i)$$

- Have to introduce a weighing function to extend scope of the algorithm
- Can be shown to imply that the landmarking kernel induced by a random sample is good kernel with high probability

# Learning with a good distance function

**Theorem ([Balcan et al., 2008a])**

*Given a strongly $(\epsilon, \gamma)$-good distance function, the following classifier h, for any $\epsilon, \delta > 0$, when given $n = \frac{16}{\gamma^2} lg\left(\frac{2}{\delta}\right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in_R \mu^+$, $b_i \in_R \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$*

$$h(x) = sgn\left[f(x)\right], f(x) = \frac{1}{n}\sum_{i=1}^{n} K(x, a_i) - \frac{1}{n}\sum_{i=1}^{n} K(x, b_i)$$

- Have to introduce a weighing function to extend scope of the algorithm
- Can be shown to imply that the landmarking kernel induced by a random sample is good kernel with high probability
- Yet another instance of weak$^*$-PAC learning

# Kernels as Kernels vs. Kernels as Similarity

- Similarity $\rightarrow$ Kernel : $(\epsilon, \gamma)$-good $\Rightarrow$ $(\epsilon + \delta, \gamma/2)$-kernel good

# Kernels as Kernels vs. Kernels as Similarity

- Similarity $\rightarrow$ Kernel : $(\epsilon, \gamma)$-good $\Rightarrow$ $(\epsilon + \delta, \gamma/2)$-kernel good
- Kernel $\rightarrow$ Similarity : $(\epsilon, \gamma)$-kernel good $\Rightarrow$ $(\epsilon + \epsilon_0, \frac{1}{2}(1 - \epsilon)\epsilon_0\gamma^2)$-kernel good

# Kernels as Kernels vs. Kernels as Similarity

- Similarity $\rightarrow$ Kernel : $(\epsilon, \gamma)$-good $\Rightarrow$ $(\epsilon + \delta, \gamma/2)$-kernel good
- Kernel $\rightarrow$ Similarity : $(\epsilon, \gamma)$-kernel good $\Rightarrow$ $(\epsilon + \epsilon_0, \frac{1}{2}(1 - \epsilon)\epsilon_0\gamma^2)$-kernel good
- [Srebro, 2007] There exist learning instances for which kernels perform better as kernels than as similarity functions

# Kernels as Kernels vs. Kernels as Similarity

- Similarity $\rightarrow$ Kernel : $(\epsilon, \gamma)$-good $\Rightarrow$ $(\epsilon + \delta, \gamma/2)$-kernel good
- Kernel $\rightarrow$ Similarity : $(\epsilon, \gamma)$-kernel good $\Rightarrow$ $(\epsilon + \epsilon_0, \frac{1}{2}(1 - \epsilon)\epsilon_0\gamma^2)$-kernel good
- [Srebro, 2007] There exist learning instances for which kernels perform better as kernels than as similarity functions
- [Balcan et al., 2008b] There exist function classes and distributions such that no kernel performs well on all the functions. However there exist similarity functions that give optimal performance

# Kernels as Kernels vs. Kernels as Similarity

- Similarity $\rightarrow$ Kernel : $(\epsilon, \gamma)$-good $\Rightarrow$ $(\epsilon + \delta, \gamma/2)$-kernel good
- Kernel $\rightarrow$ Similarity : $(\epsilon, \gamma)$-kernel good $\Rightarrow$ $(\epsilon + \epsilon_0, \frac{1}{2}(1 - \epsilon)\epsilon_0\gamma^2)$-kernel good
- [Srebro, 2007] There exist learning instances for which kernels perform better as kernels than as similarity functions
- [Balcan et al., 2008b] There exist function classes and distributions such that no kernel performs well on all the functions. However there exist similarity functions that give optimal performance
- Role of the weighing function not investigated

# Conclusion

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)
  - Work well in transductive settings

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)
  - Work well in transductive settings
  - Allow for support vector like effects

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)
  - Work well in transductive settings
  - Allow for support vector like effects
  - Not much work on generalization guarantees

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)
    - Work well in transductive settings
    - Allow for support vector like effects
    - Not much work on generalization guarantees
    - Not much known about distortion incurred when embedding test points

# The big picture

- Finite-dimensional embeddings (PE, Minkowski spaces)
  - Work well in transductive settings
  - Allow for support vector like effects
  - Not much work on generalization guarantees
  - Not much known about distortion incurred when embedding test points
  - Should work well owing to Representer Theorems

# The big picture

- Exact embeddings (Banach, Keĭn spaces)

# The big picture

- Exact embeddings (Banach, Keĭn spaces)
  - Work well in inductive settings

# The big picture

- Exact embeddings (Banach, Keĭn spaces)
  - Work well in inductive settings
  - Allow for support vector like effects

# The big picture

- Exact embeddings (Banach, Keĭn spaces)
  - Work well in inductive settings
  - Allow for support vector like effects
  - Generalization guarantees well studied

# The big picture

- Exact embeddings (Banach, Keĭn spaces)
  - Work well in inductive settings
  - Allow for support vector like effects
  - Generalization guarantees well studied
  - Embeddings are isometric or “isosimilar”

# The big picture

- Exact embeddings (Banach, Keĭn spaces)
  - Work well in inductive settings
  - Allow for support vector like effects
  - Generalization guarantees well studied
  - Embeddings are isometric or “isosimilar”
  - Too much power though ([von Luxburg and Bousquet, 2004], [Ong et al., 2004])

# The big picture

- Landmarking approaches

# The big picture

- Landmarking approaches
  - Work well in inductive settings

# The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Dont allow support vector like effects (got to keep all the landmarks)

# The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Dont allow support vector like effects (got to keep all the landmarks)
  - Generalization guarantees there

# The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Dont allow support vector like effects (got to keep all the landmarks)
  - Generalization guarantees there
  - But how does one find a “good” kernel ?

# The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Dont allow support vector like effects (got to keep all the landmarks)
  - Generalization guarantees there
  - But how does one find a “good” kernel ?

# Open questions

- Choosing the kernel : still requires one to attend Hogwarts

# Open questions

- Choosing the kernel : still requires one to attend Hogwarts
- Existing approaches to learning kernels are pathetic

# Open questions

- Choosing the kernel : still requires one to attend Hogwarts
- Existing approaches to learning kernels are pathetic
- [Balcan et al., 2008c] proposes to learn with multiple similarity functions

# Open questions

- Choosing the kernel : still requires one to attend Hogwarts
- Existing approaches to learning kernels are pathetic
- [Balcan et al., 2008c] proposes to learn with multiple similarity functions
- Need testable definitions of goodness of kernels

# Open questions

- Application of indefinite kernels to other tasks

# Open questions

- Application of indefinite kernels to other tasks
  - clustering [Balcan et al., 2008d]

# Open questions

- Application of indefinite kernels to other tasks
  - clustering [Balcan et al., 2008d]
  - principal components

# Open questions

- Application of indefinite kernels to other tasks
  - clustering [Balcan et al., 2008d]
  - principal components
  - multi-class classification [Balcan and Blum, 2006]

# Open questions

- Application of indefinite kernels to other tasks
  - clustering [Balcan et al., 2008d]
  - principal components
  - multi-class classification [Balcan and Blum, 2006]
- Analysis of the feature maps induced by embeddings into Banach, Keĭn spaces [Balcan et al., 2006]

# Bibliography I


Balcan, M.-F. and Blum, A. (2006).
On a Theory of Learning with Similarity Functions.
In *International Conference on Machine Learning*, pages 73–80.

Balcan, M.-F., Blum, A., and Srebro, N. (2008a).
A Theory of Learning with Similarity Functions.
*Machine Learning*, 71(1-2):89–112.

Balcan, M.-F., Blum, A., and Srebro, N. (2008b).
Improved Guarantees for Learning via Similarity Functions.
In *Annual Conference on Computational Learning Theory*, pages 287–298.

# Bibliography II


Balcan, M.-F., Blum, A., and Srebro, N. (2008c).
Learning with Multiple Similarity Functions.
Workshop on Kernel Learning: Automatic Selection of Optimal Kernels, Advances in Neural Information Processing Systems.

Balcan, M.-F., Blum, A., and Vempala, S. (2006).
Kernels as Features: On Kernels, Margins, and Low-dimensional Mappings.
*Machine Learning*, 65(1):79–94.

Balcan, M.-F., Blum, A., and Vempala, S. (2008d).
A Discriminative Framework for Clustering via Similarity Functions.
In *ACM Annual Symposium on Theory of Computing*, pages 671–680.

# Bibliography III

Boughorbel, S., Tarel, J.-P., and Boujemaa, N. (2005).
Conditionally Positive Definite Kernels for SVM Based Image Recognition.
In *IEEE International Conference on Multimedia & Expo*, pages 113–116.

Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., and Cazzanti, L. (2009).
Similarity-based Classification: Concepts and Algorithms.
*Journal of Machine Learning Research*, 10:747–776.

Der, R. and Lee, D. (2007).
Large Margin Classification in Banach Spaces.
In *International Conference on Artificial Intelligence and Statistics*, volume 2 of *JMLR Workshop and Conference Proceedings*, pages 91–98.

# Bibliography IV


Duda, R. O., Hart, P. E., and Stork, D. G. (2000).
*Pattern Classification*.
Wiley-Interscience, second edition.

Duin, R. P. W. and Pękalska, E. (2008).
On Refining Dissimilarity Matrices for an Improved NN Learning.
In *International Conference on Pattern Recognition*, pages 1–4.

Goldfarb, L. (1984).
A Unified Approach to Pattern Recognition.
*Pattern Recognition*, 17(5):575–582.

Gottlieb, L.-A., Kontorovich, A. L., and Krauthgamer, R. (2010).
Efficient Classification for Metric Data.
In *Annual Conference on Computational Learning Theory*.

# Bibliography V


Graepel, T., Herbrich, R., Bollmann-Sdorra, P., and Obermayer, K. (1998).
Classification on Pairwise Proximity Data.
In *Advances in Neural Information Processing Systems*, pages 438–444.

Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.-R., Obermayer, K., and Williamson, R. (1999).
Classication of Proximity Data with LP machines.
In *Ninth International Conference on Artificial Neural Networks*, pages 304–309.

Haasdonk, B. (2005).
Feature Space Interpretation of SVMs with Indefinite Kernels.
*IEEE Transactions on Pattern Analysis and Machince Intelligence*, 27(4):482–492.

# Bibliography VI


Haasdonk, B. and Bahlmann, C. (2004).
Learning with Distance Substitution Kernels.
In *Annual Symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung*, pages 220–227.

Harol, A., Pękalska, E., Verzakov, S., and Duin, R. P. W. (2006).
Augmented Embedding of Dissimilarity Data into (Pseudo-)Euclidean Spaces.
In *IAPR Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 613–621.

Hein, M., Bousquet, O., and Schölkopf, B. (2005).
Maximal Margin Classification for Metric Spaces.
*Journal of Computer and System Sciences*, 71(3):333–359.

# Bibliography VII


Jacobs, D. W., Weinshall, D., and Gdalyahu, Y. (2000).
Classification with Nonmetric Distances: Image Retrieval and Class Representation.
*IEEE Transactions on Pattern Analysis and Machince Intelligence*, 22(6):583–600.

Kearns, M. and Vazirani, U. (1997).
*An Introduction to Computational Learning Theory*.
The MIT Press.

Luss, R. and d'Aspremont, A. (2007).
Support Vector Machine Classification with Indefinite Kernels.
In *Advances in Neural Information Processing Systems*.

# Bibliography VIII


Mendelson, S. (2003).
A Few Notes on Statistical Learning Theory.
In Mendelson, S. and Smola, A. J., editors, *Advanced Lectures on Machine Learning, Machine Learning Summer School*, volume 2600 of *Lecture Notes in Computer Science*. Springer.

Mierswa, I. (2006).
Making Indefinite Kernel Learning Practical.
Technical report, Collaborative Research Center 475, University of Dortmund.

Ong, C. S., Mary, X., Canu, S., and Smola, A. J. (2004).
Learning with non-positive Kernels.
In *International Conference on Machine Learning*.

# Bibliography IX


Pękalska, E. and Duin, R. P. W. (2000).
Classifiers for Dissimilarity-Based Pattern Recognition.
In *International Conference on Pattern Recognition*, pages 2012–2016.

Pękalska, E. and Duin, R. P. W. (2001).
On Combining Dissimilarity Representations.
In *Multiple Classifier Systems*, pages 359–368.

Pękalska, E. and Duin, R. P. W. (2002).
Dissimilarity representations allow for building good classifiers.
*Pattern Recognition Letters*, 23(8):943–956.

# Bibliography X


Pękalska, E., Paclík, P., and Duin, R. P. W. (2001).
A Generalized Kernel Approach to Dissimilarity-based Classification.
*Journal of Machine Learning Research*, 2:175–211.

Schölkopf, B. (2000).
The Kernel Trick for Distances.
In *Advances in Neural Information Processing Systems*, pages 301–307.

Schölkopf, B. and Smola, A. J. (2001).
*Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*.
The MIT Press, first edition.

# Bibliography XI

Srebro, N. (2007).
How Good Is a Kernel When Used as a Similarity Measure?
In *Annual Conference on Computational Learning Theory*, pages 323–335.

von Luxburg, U. and Bousquet, O. (2004).
Distance-Based Classification with Lipschitz Functions.
*Journal of Machine Learning Research*, 5:669–695.

Wang, L., Yang, C., and Feng, J. (2007).
On Learning with Dissimilarity Functions.
In *International Conference on Machine Learning*, pages 991–998.

# Bibliography XII

Weinberger, K. Q. and Saul, L. K. (2009).
Distance Metric Learning for Large Margin Nearest Neighbor Classification.
*Journal of Machine Learning Research*, 10:207–244.

Weinshall, D., Jacobs, D. W., and Gdalyahu, Y. (1998).
Classification in Non-Metric Spaces.
In *Advances in Neural Information Processing Systems*, pages 838–846.

Zhang, H., Xu, Y., and Zhang, J. (2009).
Reproducing Kernel Banach Spaces for Machine Learning.
*Journal of Machine Learning*, 10:2741–2775.