\[ \langle \Phi(x), \Phi(x') \rangle = K(x, x') = C(x, x') - C(x, x_0) - C(x', x_0) \]

Learning in Indefiniteness

Purushottam Kar

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

August 2, 2010
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
A brief introduction to learning

Kernels - Definite and Indefinite

Using kernels as measures of distance
  - Landmarking based approaches
  - Approximate embeddings into Pseudo Euclidean spaces
  - Exact embeddings into Banach spaces

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
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 in Indefiniteness

August 2, 2010
Learning as pattern recognition

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

Purushottam Kar (CSE/IITK)
Binary classification

- Learning Dichotomies from examples
Learning

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
**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 $t \in \mathcal{F}$, 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
**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 $A$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all distributions $\mu$ defined on $\mathcal{X}$, all $t \in F$, all $\epsilon, \delta > 0 : A$, when given $(x_i, f(x_i))_{i=1}^n, x_i \in \mathcal{X} \mu$ 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

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

- $t$ 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 \( F \) defined on a domain \( X \) is said to be PAC-learnable if there exists a class of boolean functions \( H \) defined on \( X \), an algorithm \( A \) and a function \( S : \mathbb{R}^+ \times \mathbb{R}^+ \) such that for all distributions \( \mu \) defined on \( X \), all \( t \in F \), all \( \epsilon, \delta > 0 \) : \( A \), when given \((x_i, f(x_i))_{i=1}^n, x_i \in \mathbb{R} \mu \) 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 \( h \in H \) such that

\[
\Pr_{x \in \mathbb{R} \mu} \left[ h(x) \neq t(x) \right] \leq \epsilon.
\]

- \( t \) is the Target function, \( F \) the Concept Class
- \( h \) is the Hypothesis, \( H \) the Hypothesis Class
- \( S \) is the Sample Complexity of the algorithm \( A \)
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 e.g., 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 $\frac{1}{2}$. 

Weak-PAC learning restricts oneself to benign distributions (uniform, mixture of Gaussians) and 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)
Limitations of PAC learning

- Most interesting function classes are not PAC learnable with polynomial sample complexities e.g., 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 e.g., 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 $F$ defined on a domain $X$ is said to be weak*-PAC-learnable if for every $t \in F$ and distribution $\mu$ defined on $X$, there exists a class of boolean functions $H$ defined on $X$, an algorithm $A$ and a function $S : \mathbb{R}^+ \times \mathbb{R}^+$ such that for all $\epsilon, \delta > 0 : A$, 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 $h \in H$ such that

$$\Pr_{x \in_R \mu} [h(x) \neq t(x)] \leq \epsilon.$$
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 $f : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ (resp $f : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$) 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 $\forall c \in \mathbb{R}^n, c \neq 0$, $c^T A c > 0$. 

Definition

A kernel $K$ defined on a domain $X$ is said to be positive definite if $\forall n \in \mathbb{N}$, $\forall x_1, \ldots, x_n \in X$, the matrix $G = (G_{ij}) = (K(x_i, x_j))$ is positive definite. Alternatively, for every $g \in L^2(X)$, $\int \int_X g(x) g(x') K(x, x') \geq 0$.

Definition

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

**Definition**

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

**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 $g \in L_2(\mathcal{X})$, $\int \int_{\mathcal{X}} g(x)g(x')K(x, x') \geq 0$. 
**Definiteness**

**Definition**

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

**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 $g \in L_2(\mathcal{X})$, $\int\int_{\mathcal{X}} g(x)g(x')K(x, x') \geq 0$.

**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
- ...
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.
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 \(T = (x_i, t(x_i))_{i=1}^n, x_i \in X, y_i \in \{-1, +1\}, T = T^+ \cup T^-\)
Kernels as distances

Landmarking based approaches

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

\[
\text{Pr}_{x \in \mathcal{X}} \left[ d(x, X_{t(x)}) < d(x, X_{t(x)}) \right] \geq 1 - \epsilon
\]
Nearest neighbor classification [Duda et al., 2000]

- Learning domain is some distance (possibly metric) space $(\mathcal{X}, d)$
- Given $T = (x_i, t(x_i))_{i=1}^n$, $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$, $T = T^+ \cup T^-$
- 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
  - \(\Pr_{x \in R^\mu} \left[ d(x, \mathcal{X}^{t(x)}) < d(x, \mathcal{X}^{\bar{t}(x)}) \right] \geq 1 - \epsilon\)
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

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

- 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

$$
\Pr_{x, x'' \in R\mu} \left[ d(x, x') < d(x, x'') \mid x' \in X^t(x), x'' \in X^t(x) \right] \geq 1 - \epsilon + \gamma.
$$

- 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 \(n = \frac{1}{\gamma^2} \log \left( \frac{1}{\delta} \right)\) pairs of positive and negative training points, \((a_i, b_i)_{i=1}^n\), \(a_i \in \mathbb{R}^{\mu^+}\), \(b_i \in \mathbb{R}^{\mu^-}\) with probability greater than \(1 - \delta\), has an error no more than \(\epsilon + \delta\)

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

- 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 $n = \frac{1}{\gamma^2} \log \left( \frac{1}{\delta} \right)$ pairs of positive and negative training points, $(a_i, b_i)_{i=1}^n$, $a_i \in \mathbb{R}^+ \mu^+$, $b_i \in \mathbb{R}^- \mu^-$ with probability greater than $1 - \delta$, has an error no more than $\epsilon + \delta$

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

- 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 $n = \frac{1}{\gamma^2} \log \left( \frac{1}{\delta} \right)$ 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 $\epsilon + \delta$

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

- 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
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, \((a_i, b_i)_{i=1}^n\), \(a_i \in \mathbb{R} \mu^+\), \(b_i \in \mathbb{R} \mu^-\) with probability greater than \(1 - \delta\), has an error no more than \(\epsilon + \delta\)

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

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

\[ h(x) = \text{sgn} [f(x)], \quad f(x) = \min_{x_i \in T} \left( t(x_i) + 2 \frac{d(x, x_i)}{d(T^+, T^-)} \right) \]
Other landmarking approaches

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

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

- 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

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

- 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

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

- make use of approximate nearest neighbor search algorithms
- show that pseudo dimension of Lipschitz classifiers in doubling spaces is bounded
- are able to provide 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 can be extended to arbitrary \( \mathbb{R}^{(p,q)} \) (PE Spaces)
The Minkowski space-time

Definition

\[ \mathbb{R}^4 = \mathbb{R}^3 \oplus \mathbb{R}^1 : = \mathbb{R}^{(3,1)} \] 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 can be extended to arbitrary \( \mathbb{R}^{(p,q)} \) (PE Spaces)

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

Any finite pseudo metric \((X, d), |X| = n\) can be isometrically embedded in \( (\mathbb{R}^{(p,q)}, \| \cdot \|^2) \) 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 $J = I - \frac{1}{n}11^T$.

Do an eigendecomposition of $B = QQ^\top = Q|\Lambda|^{\frac{1}{2}}M|\Lambda|^{\frac{1}{2}}Q^\top$ 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 $g = -\frac{1}{2} \left( d - \frac{1}{n}11^TD \right) J$. Now find the mean square error solution to $xMX^\top = b$ as $x = bX|\Lambda|^{-1}M$. 

Purushottam Kar (CSE/IITK)
Learning in Indefiniteness
August 2, 2010 22 / 60
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 Discriminantant 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 $\forall u, v \in V, a \in F$, $\|av\| = |a|\|v\|$, $\|u + v\| \leq \|u\| + \|v\|$ and $\|v\| = 0$ if and only if $v = 0$. 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 $\forall u, v \in V, a \in F$, $\| av \| = |a| \| v \|$, $\| u + v \| \leq \| u \| + \| v \|$ and $\| v \| = 0$ if and only if $v = 0$. 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 $\Psi : \text{Lip}(\mathcal{X}) \to \mathcal{B}'$, the operator norm on $\mathcal{B}'$ giving the Lipschitz constant for each function $f \in \text{Lip}(\mathcal{X})$ 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 $\| \cdot \| : V \to \mathbb{R}$ such that $\forall u, v \in V, a \in F, \| av \| = |a|\|v\|$, $\| u + v \| \leq \| u \| + \| v \|$ and $\| v \| = 0$ if and only if $v = 0$. 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 $\text{Lip}(\mathcal{X})$ defined on $\mathcal{M}$, there exists a Banach Space $\mathcal{B}$ and maps $\Phi : \mathcal{X} \to \mathcal{B}$ and $\Psi : \text{Lip}(\mathcal{X}) \to \mathcal{B}'$, the operator norm on $\mathcal{B}'$ giving the Lipschitz constant for each function $f \in \text{Lip}(\mathcal{X})$ 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

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

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

\[ \inf_{p^+ \in C^+, p^- \in C^-} \| p^+ - p^- \| \] (1)

\[ \sup_{T \in B'} \inf_{p^+ \in C^+, p^- \in C^-} \frac{\langle T, p^+ - p^- \rangle}{\| T \|} \] (2)

\[ \inf_{T \in B', b \in \mathbb{R}} \| T \| = L(T) \] (3)

subject to \[ t(x_i) (\langle T, x_i \rangle + b) \geq 1, \forall i = 1, \ldots, n. \]
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

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

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

\[
\inf_{T \in B', b \in \mathbb{R}} \|T\| = L(T) \tag{3}
\]

subject to \( t(x_i) (\langle T, x_i \rangle + b) \geq 1, \forall i = 1, \ldots, n. \)

\[
\inf_{T \in B', b \in \mathbb{R}} L(T) + C \sum_{i=1}^{n} \xi_i \tag{4}
\]

subject to \( t(x_i) (\langle T, x_i \rangle + b) \geq 1 - \xi_i, \xi \geq 0 \forall i = 1, \ldots, n. \)
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 $\text{Lip}(f') = \text{Lip}(f)$. 

Solution to Program 3 is always of the form

$$f(x) = d(x, T^-) - d(x, T^+)$$

Solution to Program 4 is always of the form

$$g(x) = \alpha \min_i (t(x_i) + L_0 d(x, x_i)) + (1 - \alpha) \max_i (t(x_i) - L_0 d(x, x_i))$$
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 \( \text{Lip}(f') = \text{Lip}(f) \).*

- 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 $\text{Lip}(f') = \text{Lip}(f)$.*

- 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

$$g(x) = \alpha \min_i (t(x_i) + L_0 d(x, x_i)) + (1 - \alpha) \max_i (t(x_i) - L_0 d(x, x_i))$$
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(\(X\)) one recovers the SVM, LPM and NN algorithms
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

$\forall c \in \mathbb{R}^n$, $c^T 1 = 0$, $c^T Ac > 0$. 
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

$$\forall c \in \mathbb{R}^n, \ c^T 1 = 0, \ c^T Ac > 0.$$  

**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 $\forall c \in \mathbb{R}^n$, $c^T 1 = 0$, $c^T Ac > 0$.

**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.
[Der and Lee, 2007] consider exploiting the semi-inner product structure present in Banach space to yield SVM formulations.
[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 $\mathcal{B} \times \mathcal{B}$
[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)
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]
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]

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) = \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j C(x_i, x_j) \text{ subject to } \sum_{i=1}^{n} \alpha_i y_i = 0$$
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
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 \((X, K)\) 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 \(f(x) = \sum_{i=1}^{n} K(x, x_i)\).
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 $K(||x - y||), K(\langle x, y \rangle)$
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 $K(\|x - y\|), K(\langle x, y \rangle)$
  - 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
[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?
[Haasdonk, 2005] embeds training data into a PE space and formulates a $\nu$-SVM-like classifier there

Not a margin maximization formulation
[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)
[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
[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 \((\mathcal{K}, \langle \cdot, \cdot \rangle_{\mathcal{K}})\) is called a Kreĭn space if there exist two Hilbert spaces \(\mathcal{H}_+\) and \(\mathcal{H}_-\) such that \(\mathcal{K} = \mathcal{H}_+ \oplus \mathcal{H}_-\) 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.
[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 Krein 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
[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 $\exists \beta \in K$

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

**Definition**

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

$$\mathbb{E}_{x' \in R^\mu^+} [K(x, x')] > \mathbb{E}_{x' \in R^\mu^-} [K(x, x')] + \gamma.$$
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 \(n = \frac{16}{\gamma^2} \log \left( \frac{2}{\delta} \right)\) pairs of positive and negative training points, \((a_i, b_i)_{i=1}^n, a_i \in \mathbb{R}^+ \mu^+, b_i \in \mathbb{R}^\mu^-\) with probability greater than \(1 - \delta\), has an error no more than \(\epsilon + \delta\)

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

- 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, \((a_i, b_i)_{i=1}^n\), \(a_i \in \mathbb{R} \mu^+\), \(b_i \in \mathbb{R} \mu^-\) with probability greater than \(1 - \delta\), has an error no more than \(\epsilon + \delta\)

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

- 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 \(n = \frac{16}{\gamma^2} \log \left( \frac{2}{\delta} \right)\) 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 \(\epsilon + \delta\)

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

- 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 → Kernel: \((\epsilon, \gamma)\)-good \(\Rightarrow (\epsilon + \delta, \gamma/2)\)-kernel good
Kernels as Kernels vs. Kernels as Similarity

- **Similarity → Kernel**: \((\epsilon, \gamma)-\text{good} \Rightarrow (\epsilon + \delta, \gamma/2)-\text{kernel good}\)
- **Kernel → Similarity**: \((\epsilon, \gamma)-\text{kernel good} \Rightarrow (\epsilon + \epsilon_0, \frac{1}{2}(1 - \epsilon)\epsilon_0\gamma^2)-\text{kernel good}\)
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, Keǐn spaces)
The big picture

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

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

[Note: Additional details or references may be included in the full document, not transcribed here.]
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

- Exact embeddings (Banach, Kőn 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, Keǐn 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
  - Don't allow support vector like effects (got to keep all the landmarks)
The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Don’t allow support vector like effects (got to keep all the landmarks)
  - Generalization guarantees there
The big picture

- Landmarking approaches
  - Work well in inductive settings
  - Don’t 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
  - Don't allow support vector like effects (got to keep all the landmarks)
  - Generalization guarantees there
  - But how does one find a “good” kernel?
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, Kečn spaces [Balcan et al., 2006]


Bibliography XI


Bibliography XII

