Similarity-based Learning via Data Driven Embeddings*

Purushottam Kar[1] Prateek Jain[2]

[1]Indian Institute of Technology
Kanpur

[2]Microsoft Research India
Bengaluru

November 3, 2011

*To appear in the proceedings of NIPS 2011

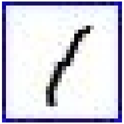
Outline

Outline

Digit Classification†

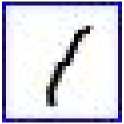
† MNIST database : http://yann.lecun.com/exdb/mnist/

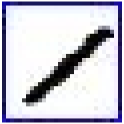
Digit Classification†



†MNIST database : http://yann.lecun.com/exdb/mnist/

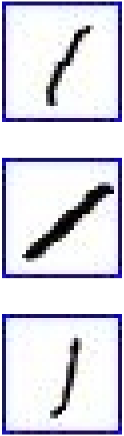
Digit Classification†





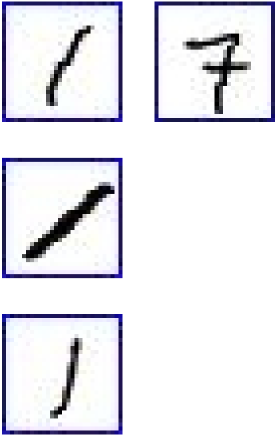
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



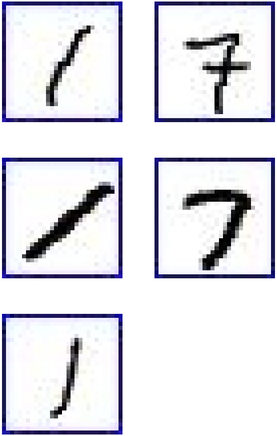
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



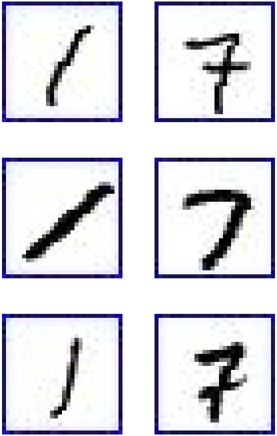
†MNIST database : `http://yann.lecun.com/exdb/mnist/`

Digit Classification†



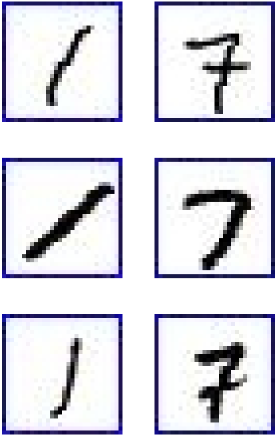
† MNIST database : `http://yann.lecun.com/exdb/mnist/`

Digit Classification†



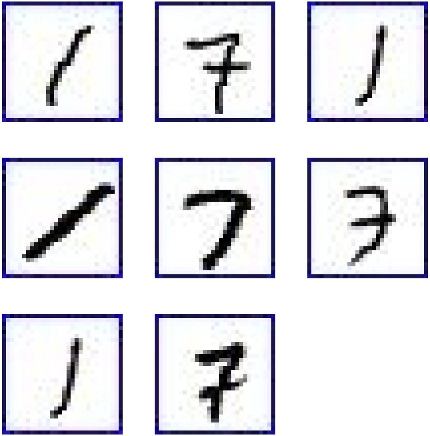
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



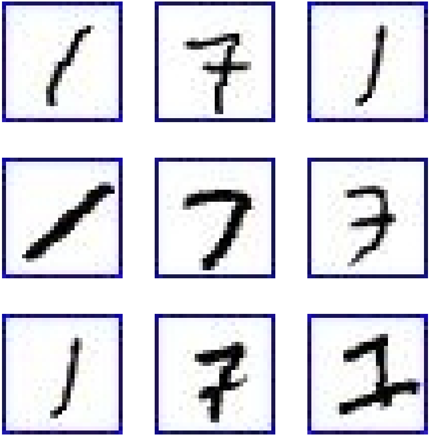
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification[†]



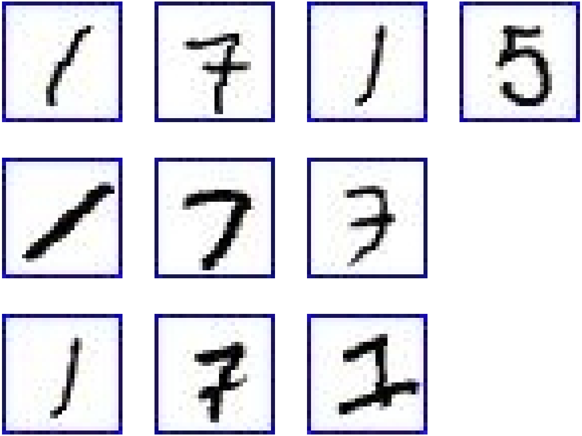
[†]MNIST database : `http://yann.lecun.com/exdb/mnist/`

Digit Classification†



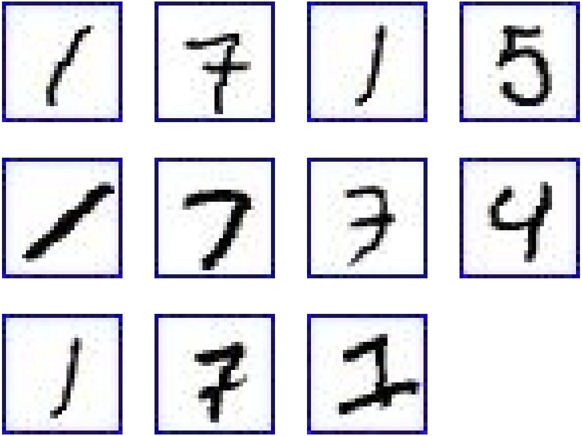
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



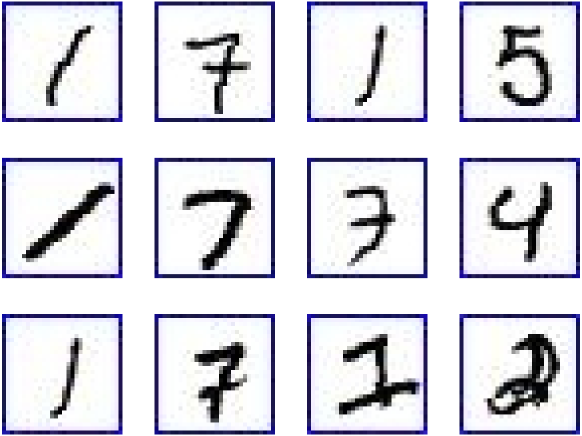
†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



†MNIST database : http://yann.lecun.com/exdb/mnist/

Digit Classification†



†MNIST database : `http://yann.lecun.com/exdb/mnist/`

Spam mail detection

Spam mail detection

```
Dear Junta,

The Hall-8 mess will be closed for the
occasion of Diwali at lunch & dinner
time.  The breakfast will be served
along with Lunch packets tomorrow (26th
October, 2011).

Please collect your Lunch Packet.  The
mess would resume its normal working from
27th October.
```

A legitimate mail

Spam mail detection

```
Dear Junta,

The Hall-8 mess will be closed for the
occasion of Diwali at lunch & dinner
time.  The breakfast will be served
along with Lunch packets tomorrow (26th
October, 2011).

Please collect your Lunch Packet.  The
mess would resume its normal working from
27th October.
```

A legitimate mail

```
Hello,
I am resending my previous mail to you,
I hope you do get it this time around
and understand its content fully.  I
am contacting you briefly based on the
Investment of Forty Five Million Dollars
(US$ 45,000,000:00) in your country, as I
presently have a client who is interested
in investing in your country.
Sincerely Yours,
J. Costa
```

Most likely a spam mail

Spam mail detection

```
Dear Junta,

The Hall-8 mess will be closed for the
occasion of Diwali at lunch & dinner
time.  The breakfast will be served
along with Lunch packets tomorrow (26th
October, 2011).

Please collect your Lunch Packet.  The
mess would resume its normal working from
27th October.
```

A legitimate mail

```
Hello,
I am resending my previous mail to you,
I hope you do get it this time around
and understand its content fully.  I
am contacting you briefly based on the
Investment of Forty Five Million Dollars
(US$ 45,000,000:00) in your country, as I
presently have a client who is interested
in investing in your country.
Sincerely Yours,
J. Costa
```

Most likely a spam mail

```
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SEMINAR SERIES
Departmental Colloquium

Title:  Similarity-based Learning via Data Driven Embeddings

Speaker:  Purushottam Kar

Affiliation:  Ph.D. Scholar, CSE Dept., IIT Kanpur
```

To each his own ...

More formally ...

- We are working over a domain $\mathcal{X}$ and wish to learn a target classifier over the domain $\ell : \mathcal{X} \rightarrow \{-1, +1\}$.

More formally ...

- We are working over a domain $\mathcal{X}$ and wish to learn a target classifier over the domain $\ell : \mathcal{X} \rightarrow \{-1, +1\}$.
- We are given *training points* $S = \{x_1, x_2, \ldots, x_n\}$ sampled from some distribution $\mathcal{D}$ over $\mathcal{X}$ and their true labels $\{\ell(x_1), \ldots, \ell(x_n)\}$.

More formally ...

- We are working over a domain $\mathcal{X}$ and wish to learn a target classifier over the domain $\ell : \mathcal{X} \rightarrow \{-1, +1\}$.
- We are given *training points* $S = \{x_1, x_2, \ldots, x_n\}$ sampled from some distribution $\mathcal{D}$ over $\mathcal{X}$ and their true labels $\{\ell(x_1), \ldots, \ell(x_n)\}$.
- Our goal is to output a classifier $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$ such that it mostly gives out the true labels.

$$\Pr_{x \sim \mathcal{D}} \left[\hat{\ell}(x) \neq \ell(x) \right] < \epsilon$$

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?
- SOLUTION 1 : Force a numeric representation by embedding all data in some Euclidean space $\mathbb{R}^d$

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?
- SOLUTION 1 : Force a numeric representation by embedding all data in some Euclidean space $\mathbb{R}^d$

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

  - Easy to do for images : $(n \times n)$ pixels $\mapsto \mathbb{R}^{3n^2}$ for RGB images

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?
- SOLUTION 1 : Force a numeric representation by embedding all data in some Euclidean space $\mathbb{R}^d$

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

  - Easy to do for images : $(n \times n)$ pixels $\mapsto \mathbb{R}^{3n^2}$ for RGB images
  - Easier said than done for text, emails, web data (eg. BoW for text)

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?
- SOLUTION 1 : Force a numeric representation by embedding all data in some Euclidean space $\mathbb{R}^d$

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

  - Easy to do for images : $(n \times n)$ pixels $\mapsto \mathbb{R}^{3n^2}$ for RGB images
  - Easier said than done for text, emails, web data (eg. BoW for text)
- SOLUTION 2 : Work with some distance/similarity function over the data

Representing the data

- Most learning algorithms (Perceptron, MRF, DBN, SVM, ...) like working with numeric data i.e. $\mathcal{X} \subset \mathbb{R}^d$
- How to make heterogeneous data (images, sound, web data) numeric ?
- SOLUTION 1 : Force a numeric representation by embedding all data in some Euclidean space $\mathbb{R}^d$

$$\Phi : \mathcal{X} \rightarrow \mathbb{R}^d$$

  - Easy to do for images : $(n \times n)$ pixels $\mapsto \mathbb{R}^{3n^2}$ for RGB images
  - Easier said than done for text, emails, web data (eg. BoW for text)
- SOLUTION 2 : Work with some distance/similarity function over the data ✓

Outline

Classical algorithms that learn with similarities

- Let K be a similarity measure (or w.l.o.g. a distance measure)

Classical algorithms that learn with similarities

- Let K be a similarity measure (or w.l.o.g. a distance measure)
- Nearest neighbor classification

$$\begin{aligned} \hat{\ell}(x) &= \ell(\mathrm{NN}(x)) \\ \mathrm{NN}(x) &= \underset{x' \in S}{\arg\max} \left[K(x, x') \right] \end{aligned}$$

Classical algorithms that learn with similarities

- Let K be a similarity measure (or w.l.o.g. a distance measure)
- Nearest neighbor classification

$$\begin{aligned} \hat{\ell}(x) &= \ell(\text{NN}(x)) \\ \text{NN}(x) &= \arg\max_{x' \in S} \left[K(x, x') \right] \end{aligned}$$

- Perceptron algorithm : $\mathcal{X} \subset \mathbb{R}^d$

$$\hat{\ell}(x) = \text{sgn}\left(\langle w, x \rangle \right) \qquad \text{for some } w \in \mathbb{R}^d$$

Classical algorithms that learn with similarities

- Let K be a similarity measure (or w.l.o.g. a distance measure)
- Nearest neighbor classification

$$\begin{aligned} \hat{\ell}(x) &= \ell(\mathrm{NN}(x)) \\ \mathrm{NN}(x) &= \underset{x' \in S}{\arg\max} \left[K(x, x') \right] \end{aligned}$$

- Perceptron algorithm : $\mathcal{X} \subset \mathbb{R}^d$

$$\begin{aligned} \hat{\ell}(x) &= \mathrm{sgn}\left(\langle w, x \rangle\right) \qquad \text{for some } w \in \mathbb{R}^d \\ \hat{\ell}(x) &= \mathrm{sgn}\left(\sum_{x' \in S} \alpha(x') K(x, x') \ell(x') \right) \\ K(x, x') &= \langle x, x' \rangle \\ w &= \sum_{x' \in S} \alpha(x') \ell(x') \end{aligned}$$

Classical algorithms that learn with similarities

- Let K be a similarity measure (or w.l.o.g. a distance measure)
- Nearest neighbor classification

$$\begin{aligned}\hat{\ell}(x) &= \ell(\mathrm{NN}(x)) \\ \mathrm{NN}(x) &= \arg\max_{x' \in S} \left[K(x, x')\right]\end{aligned}$$

- Perceptron algorithm : $\mathcal{X} \subset \mathbb{R}^d$

$$\hat{\ell}(x) = \mathrm{sgn}\left(\langle w, x\rangle\right) \qquad \text{for some } w \in \mathbb{R}^d$$

- SVM allows use of arbitrary Positive semi-definite kernels

$$\hat{\ell}(x) = \mathrm{sgn}\left(\sum_{x' \in S} \alpha_{\mathrm{SVM}}(x') K(x, x') \ell(x')\right)$$

Learning with Similarities

- A lot of work was done in trying to incorporate various similarity measures, distance measures into such frameworks [Pękalska and Duin, 2001, Weinberger and Saul, 2009]

Learning with Similarities

- A lot of work was done in trying to incorporate various similarity measures, distance measures into such frameworks [Pękalska and Duin, 2001, Weinberger and Saul, 2009]
- A fair amount went into algorithms that did not require PSD kernels as SVMs do [Goldfarb, 1984]

Learning with Similarities

- A lot of work was done in trying to incorporate various similarity measures, distance measures into such frameworks [Pękalska and Duin, 2001, Weinberger and Saul, 2009]
- A fair amount went into algorithms that did not require PSD kernels as SVMs do [Goldfarb, 1984]
- Some very nice work involving isometric embeddings to (pseudo)Hilbert / Banach spaces [Gottlieb et al., 2010, von Luxburg and Bousquet, 2004, Haasdonk, 2005]

Learning with Similarities

- A lot of work was done in trying to incorporate various similarity measures, distance measures into such frameworks [Pękalska and Duin, 2001, Weinberger and Saul, 2009]
- A fair amount went into algorithms that did not require PSD kernels as SVMs do [Goldfarb, 1984]
- Some very nice work involving isometric embeddings to (pseudo)Hilbert / Banach spaces [Gottlieb et al., 2010, von Luxburg and Bousquet, 2004, Haasdonk, 2005]
- However, none addressed the issue of suitability of the similarity/distance measure to the learning task

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance
- In general, several domains have preferred notions of similarity (e.g. earth mover's distance for images)

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance
- In general, several domains have preferred notions of similarity (e.g. earth mover's distance for images)
- Can formal notions of suitability lead to guaranteed performance ?

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance
- In general, several domains have preferred notions of similarity (e.g. earth mover's distance for images)
- Can formal notions of suitability lead to guaranteed performance ?
  - For SVMs, suitability is formalized in terms of the *margin* offered by the PSD kernel in its RKHS

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance
- In general, several domains have preferred notions of similarity (e.g. earth mover's distance for images)
- Can formal notions of suitability lead to guaranteed performance ?
  - For SVMs, suitability is formalized in terms of the *margin* offered by the PSD kernel in its RKHS
  - Having large margin does lead to generalization bounds [Shawe-Taylor et al., 1998, Balcan et al., 2006]

Suitable Similarities

- A suitable similarity should intuitively give better classifier performance
- It is very well known that the choice of the kernel has a significant impact on SVM classifier performance
- In general, several domains have preferred notions of similarity (e.g. earth mover's distance for images)
- Can formal notions of suitability lead to guaranteed performance ?
  - For SVMs, suitability is formalized in terms of the *margin* offered by the PSD kernel in its RKHS
  - Having large margin does lead to generalization bounds [Shawe-Taylor et al., 1998, Balcan et al., 2006]
- Can we do the same for non-PSD similarities ?

Outline

Outline

What is a good similarity function ?

- Intuitively, a good similarity function should at least respect the labeling of the domain

What is a good similarity function ?

- Intuitively, a good similarity function should at least respect the labeling of the domain
- It should not assign small similarity to points with same label and large similarity to distinctly labeled points

What is a good similarity function ?

- Intuitively, a good similarity function should at least respect the labeling of the domain
- It should not assign small similarity to points with same label and large similarity to distinctly labeled points

Definition ([Balcan and Blum, 2006])

A similarity $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be (ϵ, γ)-good for a classification problem if for some weighing function $w : \mathcal{X} \rightarrow [-1, 1]$, at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\mathop{\mathbb{E}}_{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}} \left[w\left(x'\right) K(x, x') - w\left(x''\right) K(x, x'') \right] \geq \gamma$$

What is a good similarity function ?

- Intuitively, a good similarity function should at least respect the labeling of the domain
- It should not assign small similarity to points with same label and large similarity to distinctly labeled points

Definition ([Balcan and Blum, 2006])

A similarity $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be (ϵ, γ)-good for a classification problem if for some weighing function $w : \mathcal{X} \to [-1, 1]$, at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\mathop{\mathbb{E}}_{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}} \left[w\left(x'\right) K(x, x') - w\left(x''\right) K(x, x'') \right] \geq \gamma$$

- In other words, according to the similarity function, most points, on an average, are more similar to points of the same label

Learning with a good similarity function

Theorem ([Balcan and Blum, 2006])

Given an (ϵ, γ)-good similarity function, for any $\delta > 0$, given $n = \frac{16}{\gamma^2} \lg \frac{2}{\delta}$ labeled points $(x_i)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{2}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} w(x_i)\ell(x_i)K(x, x_i)\right)$$

Learning with a good similarity function

Theorem ([Balcan and Blum, 2006])

Given an (ϵ, γ)-good similarity function, for any $\delta > 0$, given $n = \frac{16}{\gamma^2} \lg \frac{2}{\delta}$ labeled points $(x_i)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{2}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} w(x_i)\ell(x_i)K(x, x_i)\right)$$

- Notice that the classifier is very similar in form to the SVM and Perceptron classifiers

Learning with a good similarity function

Theorem ([Balcan and Blum, 2006])

Given an (ϵ, γ)-good similarity function, for any $\delta > 0$, given $n = \frac{16}{\gamma^2} \lg \frac{2}{\delta}$ labeled points $(x_i)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{2}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} w(x_i)\ell(x_i)K(x, x_i)\right)$$

- Notice that the classifier is very similar in form to the SVM and Perceptron classifiers
- Consequently one can use these algorithms to learn this classifier as well

Outline

What is a good distance function

Definition ([Wang et al., 2007])

A distance function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be (ϵ, γ, B)-good for a classification problem if there exist two class conditional probability distributions $\tilde{\mathcal{D}}_+$ and $\tilde{\mathcal{D}}_-$ such that for all $x \in \mathcal{X}$, $\frac{\tilde{\mathcal{D}}_+(x)}{\mathcal{D}(x)} < \sqrt{B}$ and $\frac{\tilde{\mathcal{D}}_-(x)}{\mathcal{D}(x)} < \sqrt{B}$, such that at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\Pr_{\substack{x' \sim \tilde{\mathcal{D}}_+ \\ x'' \sim \tilde{\mathcal{D}}_-}} \left[\ell(x) \left(\ell(x') d(x, x') - \ell(x'') d(x, x'') \right) < 0 \right] \geq \frac{1}{2} + \gamma$$

What is a good distance function

Definition ([Wang et al., 2007])

A distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be (ϵ, γ, B)-good for a classification problem if there exist two class conditional probability distributions $\tilde{\mathcal{D}}_+$ and $\tilde{\mathcal{D}}_-$ such that for all $x \in \mathcal{X}$, $\frac{\tilde{\mathcal{D}}_+(x)}{\mathcal{D}(x)} < \sqrt{B}$ and $\frac{\tilde{\mathcal{D}}_-(x)}{\mathcal{D}(x)} < \sqrt{B}$, such that at least a $(1-\epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\Pr_{\substack{x' \sim \tilde{\mathcal{D}}_+ \\ x'' \sim \tilde{\mathcal{D}}_-}} \left[\ell(x) \left(\ell(x')d(x,x') - \ell(x'')d(x,x'') \right) < 0 \right] \geq \frac{1}{2} + \gamma$$

- The definition expects the distance function to set dissimilarly labeled points farther off than similarly labeled points

What is a good distance function

Definition ([Wang et al., 2007])

A distance function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be (ϵ, γ, B)-good for a classification problem if there exist two class conditional probability distributions $\tilde{\mathcal{D}}_+$ and $\tilde{\mathcal{D}}_-$ such that for all $x \in \mathcal{X}$, $\frac{\tilde{\mathcal{D}}_+(x)}{\mathcal{D}(x)} < \sqrt{B}$ and $\frac{\tilde{\mathcal{D}}_-(x)}{\mathcal{D}(x)} < \sqrt{B}$, such that at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\Pr_{\substack{x' \sim \tilde{\mathcal{D}}_+ \\ x'' \sim \tilde{\mathcal{D}}_-}} \left[\ell(x) \left(\ell(x') d(x, x') - \ell(x'') d(x, x'') \right) < 0 \right] \geq \frac{1}{2} + \gamma$$

- The definition expects the distance function to set dissimilarly labeled points farther off than similarly labeled points
- Yet again this yields a classifier with guaranteed generalization properties

Learning with a good distance function

Theorem ([Wang et al., 2007])

Given an (ϵ, γ, B)-good distance function, for any $\delta > 0$, given $n = \frac{4B^2}{\gamma^2} \lg \frac{1}{\delta}$ pairs of positive and negatively labeled points $\left(x_i^+, x_i^-\right)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{B}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} \beta_i \, sgn\left(d(x, x_i^+) - d(x, x_1^-)\right)\right), \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0$$

Learning with a good distance function

Theorem ([Wang et al., 2007])

Given an (ϵ, γ, B)-good distance function, for any $\delta > 0$, given $n = \frac{4B^2}{\gamma^2} \lg \frac{1}{\delta}$ pairs of positive and negatively labeled points $\left(x_i^+, x_i^-\right)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{B}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} \beta_i \, sgn\left(d(x, x_i^+) - d(x, x_1^-)\right)\right), \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0$$

- This naturally lends itself to a boosting-like implementation

Learning with a good distance function

Theorem ([Wang et al., 2007])

Given an (ϵ, γ, B)-good distance function, for any $\delta > 0$, given $n = \frac{4B^2}{\gamma^2} \lg \frac{1}{\delta}$ pairs of positive and negatively labeled points $\left(x_i^+, x_i^-\right)_{i=1}^n$, the classifier $\hat{\ell}$ defined below has error at margin $\frac{\gamma}{B}$ no more than $\epsilon + \delta$ with probability greater than $1 - \delta$,

$$\hat{\ell}(x) = sgn\left(\sum_{i=1}^{n} \beta_i\, sgn\left(d(x, x_i^+) - d(x, x_1^-)\right)\right), \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0$$

- This naturally lends itself to a boosting-like implementation
- Each of the pairs yields a stump $\mathrm{sgn}\left(d(x, x_i^+) - d(x, x_1^-)\right)$

Outline

A unified notion of what is a good similarity/distance

- Disparate as the last two models may seem, they are, in fact, quite related to each other

A unified notion of what is a good similarity/distance

- Disparate as the last two models may seem, they are, in fact, quite related to each other
- Motivated by this observation we propose a notion of goodness that is data-sensitive

A unified notion of what is a good similarity/distance

- Disparate as the last two models may seem, they are, in fact, quite related to each other
- Motivated by this observation we propose a notion of goodness that is data-sensitive
- This notion allows us to tune the goodness notion itself, allowing for better classifiers

A unified notion of what is a good similarity/distance

- Disparate as the last two models may seem, they are, in fact, quite related to each other
- Motivated by this observation we propose a notion of goodness that is data-sensitive
- This notion allows us to tune the goodness notion itself, allowing for better classifiers
- The resulting model subsumes the previous two models

A unified notion of what is a good similarity/distance

- Disparate as the last two models may seem, they are, in fact, quite related to each other
- Motivated by this observation we propose a notion of goodness that is data-sensitive
- This notion allows us to tune the goodness notion itself, allowing for better classifiers
- The resulting model subsumes the previous two models
- Consequently, the model does not require separate treatment for similarity and distance functions either

What is a good similarity/distance function

Definition (K. and Jain, 2011)

A similarity function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be (ϵ, γ, B)-good for a classification problem if for some antisymmetric *transfer* function $f : \mathbb{R} \to [-C_f, C_f]$ and some weighing function $w : \mathcal{X} \times \mathcal{X} \to [-B, B]$, at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\mathop{\mathbb{E}}_{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}} \left[w(x', x'') f(K(x, x') - K(x, x'')) \right] \geq 2C_f \gamma$$

What is a good similarity/distance function

Definition (K. and Jain, 2011)

A similarity function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be (ϵ, γ, B)-good for a classification problem if for some antisymmetric *transfer* function $f : \mathbb{R} \rightarrow [-C_f, C_f]$ and some weighing function $w : \mathcal{X} \times \mathcal{X} \rightarrow [-B, B]$, at least a $(1 - \epsilon)$ probability mass of examples $x \sim \mathcal{D}$ satisfies

$$\mathop{\mathbb{E}}_{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}} \left[w\left(x', x''\right) f\left(K(x, x') - K(x, x'')\right)\right] \geq 2C_f \gamma$$

- With appropriate setting of the weighing function and the transfer function, the previous two models can be recovered.

Outline

Learning with data-sensitive notions of suitability

- The learning algorithm is not as simple as before since the guarantees we give hold only if the a good transfer function is chosen.

Learning with data-sensitive notions of suitability

- The learning algorithm is not as simple as before since the guarantees we give hold only if the a good transfer function is chosen.
- Let us first see how, given a (good) transfer function, can we learn a (good) classifier.

Learning with data-sensitive notions of suitability

- The learning algorithm is not as simple as before since the guarantees we give hold only if the a good transfer function is chosen.
- Let us first see how, given a (good) transfer function, can we learn a (good) classifier.
- We will later on plug in the routines to learn the transfer function as well.

Learning with data-sensitive notions of suitability

Algorithm 1 LEARN-DISSIM

Require: A similarity function K, landmark pairs $\mathcal{L} = \left(x_i^+, x_i^-\right)_{i=1}^n$, a good transfer function f.

Ensure: A classifier $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$

1: Define $\Phi_{\mathcal{L}} : \mathcal{X} \rightarrow \mathbb{R}^n$ as $\Phi_{\mathcal{L}} : x \mapsto \left(f(K(x, x_i^+) - K(x, x_i^-))\right)_{i=1}^n$

2: Get a labeled training set $T = \{t_j\}_{j=1}^{n'} \subset \mathcal{X}$ sampled from $\mathcal{D}$.

3: $T' \leftarrow \{\Phi_{\mathcal{L}}(t_j)\}_{j=1}^{n'} \subset \mathbb{R}^n$ be the data set embedded in $\mathbb{R}^n$

4: Learn a linear hyperplane over $\mathbb{R}^n$ using T', $\ell_{\text{lin}} \leftarrow$ LEARN-LINEAR(T')

5: Let $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$ be defined as $\hat{\ell} : x \mapsto \ell_{\text{lin}}\left(\Phi_{\mathcal{L}}(x)\right)$

6: **return** $\hat{\ell}$

Learning with data-sensitive notions of suitability

Algorithm 1 LEARN-DISSIM

Require: A similarity function K, landmark pairs $\mathcal{L} = \left(x_i^+, x_i^-\right)_{i=1}^{n}$, a good transfer function f.

Ensure: A classifier $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$

1: Define $\Phi_{\mathcal{L}} : \mathcal{X} \rightarrow \mathbb{R}^n$ as $\Phi_{\mathcal{L}} : x \mapsto \left(f(K(x, x_i^+) - K(x, x_i^-))\right)_{i=1}^{n}$

2: Get a labeled training set $T = \{t_j\}_{j=1}^{n'} \subset \mathcal{X}$ sampled from $\mathcal{D}$.

3: $T' \leftarrow \{\Phi_{\mathcal{L}}(t_j)\}_{j=1}^{n'} \subset \mathbb{R}^n$ be the data set embedded in $\mathbb{R}^n$

4: Learn a linear hyperplane over $\mathbb{R}^n$ using T', $\ell_{\text{lin}} \leftarrow$ LEARN-LINEAR(T')

5: Let $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$ be defined as $\hat{\ell} : x \mapsto \ell_{\text{lin}}\left(\Phi_{\mathcal{L}}(x)\right)$

6: **return** $\hat{\ell}$

- LEARN-LINEAR may be taken to be any linear hyperplane learning algorithm such as Perceptron, SVM.

Learning with data-sensitive notions of suitability

Algorithm 1 LEARN-DISSIM

Require: A similarity function K, landmark pairs $\mathcal{L} = \left(x_i^+, x_i^-\right)_{i=1}^n$, a good transfer function f.

Ensure: A classifier $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$

1: Define $\Phi_{\mathcal{L}} : \mathcal{X} \rightarrow \mathbb{R}^n$ as $\Phi_{\mathcal{L}} : x \mapsto \left(f(K(x, x_i^+) - K(x, x_i^-))\right)_{i=1}^n$

2: Get a labeled training set $T = \{t_j\}_{j=1}^{n'} \subset \mathcal{X}$ sampled from $\mathcal{D}$.

3: $T' \leftarrow \{\Phi_{\mathcal{L}}(t_j)\}_{j=1}^{n'} \subset \mathbb{R}^n$ be the data set embedded in $\mathbb{R}^n$

4: Learn a linear hyperplane over $\mathbb{R}^n$ using T', $\ell_{\text{lin}} \leftarrow$ LEARN-LINEAR(T')

5: Let $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$ be defined as $\hat{\ell} : x \mapsto \ell_{\text{lin}}\left(\Phi_{\mathcal{L}}(x)\right)$

6: **return** $\hat{\ell}$

- LEARN-LINEAR may be taken to be any linear hyperplane learning algorithm such as Perceptron, SVM.
- The above procedure essentially creates a *data-driven*, problem specific embedding of the domain $\mathcal{X}$ into a Euclidean space

Learning with data-sensitive notions of suitability

- The results given earlier guarantee small classification error at large margin

Learning with data-sensitive notions of suitability

- The results given earlier guarantee small classification error at large margin
- Not amenable to efficient algorithms as hyperplane classification error is NP-hard to minimize
  [Garey and Johnson, 1979, Arora et al., 1997]

Learning with data-sensitive notions of suitability

- The results given earlier guarantee small classification error at large margin
- Not amenable to efficient algorithms as hyperplane classification error is NP-hard to minimize [Garey and Johnson, 1979, Arora et al., 1997]
- We provide our guarantees in terms of smooth Lipschitz losses like hinge-loss, log-loss etc that can be efficiently minimized over large datasets.

Working with surrogate loss functions

Definition (K. and Jain, 2011)

A similarity function is said to be (ϵ, B)-good with respect to a loss function $L : \mathbb{R} \rightarrow \mathbb{R}^+$ if for some transfer function $f : \mathbb{R} \rightarrow \mathbb{R}$ and some weighing function $w : \mathcal{X} \times \mathcal{X} \rightarrow [-B, B]$, $\underset{x \sim \mathcal{D}}{\mathbb{E}} [L(G(x))] \leq \epsilon$ where

$$G(x) = \underset{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}}{\mathbb{E}} \left[w(x', x'') \, f(K(x, x') - K(x, x'')) \right]$$

Working with surrogate loss functions

Definition (K. and Jain, 2011)

A similarity function is said to be (ϵ, B)-good with respect to a loss function $L : \mathbb{R} \to \mathbb{R}^+$ if for some transfer function $f : \mathbb{R} \to \mathbb{R}$ and some weighing function $w : \mathcal{X} \times \mathcal{X} \to [-B, B]$, $\underset{x \sim \mathcal{D}}{\mathbb{E}} [L(G(x))] \leq \epsilon$ where

$$G(x) = \underset{\substack{x' \sim \mathcal{D}, \ell(x') = \ell(x) \\ x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}}{\mathbb{E}} \left[w(x', x'') \, f(K(x, x') - K(x, x'')) \right]$$

Theorem (K. and Jain, 2011)

If K is an (ϵ, B)-good similarity function with respect to a C_L-Lipschitz loss function L then for any $\epsilon_1 > 0$, with probability at least $1 - \delta$ over the choice of $d = (16B^2 C_L^2 / \epsilon_1^2) \ln(4B/\delta\epsilon_1)$ landmark pairs, the expected loss of the classifier $\hat{\ell}(x)$ returned by LEARN-DISSIM with respect to L satisfies $\underset{x}{\mathbb{E}} \left[L(\hat{\ell}(x)) \right] \leq \epsilon + \epsilon_1$.

Outline

Learning the transfer function

- We give uniform convergence guarantees that enable standard ERM-based routines to recover the best transfer from any compact class of antisymmetric functions.

Learning the transfer function

- We give uniform convergence guarantees that enable standard ERM-based routines to recover the best transfer from any compact class of antisymmetric functions.
- This will yield a nested learning problem with the ERM-based transfer function learning algorithm calling the classifier learning algorithm as a subroutine.

Learning the transfer function

- We give uniform convergence guarantees that enable standard ERM-based routines to recover the best transfer from any compact class of antisymmetric functions.
- This will yield a nested learning problem with the ERM-based transfer function learning algorithm calling the classifier learning algorithm as a subroutine.
- For any transfer function f and arbitrary set of landmarks $\mathcal{L}$, let $L(f) = \underset{x\sim\mathcal{D}}{\mathbb{E}}\left[L(G(x))\right]$ and let $L(f, \mathcal{L})$ denote the generalization loss of the best classifier that uses the embedding $\Phi_{\mathcal{L}}$ defined by the landmarks $\mathcal{L}$.

Learning the transfer function

- We give uniform convergence guarantees that enable standard ERM-based routines to recover the best transfer from any compact class of antisymmetric functions.
- This will yield a nested learning problem with the ERM-based transfer function learning algorithm calling the classifier learning algorithm as a subroutine.
- For any transfer function f and arbitrary set of landmarks $\mathcal{L}$, let $L(f) = \underset{x \sim \mathcal{D}}{\mathbb{E}}[L(G(x))]$ and let $L(f, \mathcal{L})$ denote the generalization loss of the best classifier that uses the embedding $\Phi_{\mathcal{L}}$ defined by the landmarks $\mathcal{L}$.
- The earlier result shows that for a *fixed* f, for a large enough random $\mathcal{L}$, $L(f, \mathcal{L}) \leq L(f) + \epsilon_1$.

Learning the transfer function

Theorem (K. and Jain, 2011)

Let $\mathcal{F}$ be a compact class of transfer functions with respect to the infinity norm and $\epsilon_1, \delta > 0$. *Let* $\mathcal{N}(\mathcal{F}, r)$ *be the size of the smallest* ϵ*-net over* $\mathcal{F}$ *with respect to the infinity norm at scale* $r = \frac{\epsilon_1}{4C_L B}$. *Taking* $n = \frac{64B^2C_L^2}{\epsilon_1^2}\ln\left(\frac{16B\cdot\mathcal{N}(\mathcal{F},r)}{\delta\epsilon_1}\right)$ *random landmark pairs, we have with probability greater than* $(1-\delta)$

$$\sup_{f\in\mathcal{F}}\left[\left|L(f,\mathcal{L}) - L(f)\right|\right] \leq \epsilon_1$$

Learning the transfer function

Theorem (K. and Jain, 2011)

Let $\mathcal{F}$ be a compact class of transfer functions with respect to the infinity norm and $\epsilon_1, \delta > 0$. Let $\mathcal{N}(\mathcal{F}, r)$ be the size of the smallest ϵ-net over $\mathcal{F}$ with respect to the infinity norm at scale $r = \frac{\epsilon_1}{4C_L B}$. Taking $n = \frac{64B^2C_L^2}{\epsilon_1^2} \ln\left(\frac{16B\cdot\mathcal{N}(\mathcal{F},r)}{\delta\epsilon_1}\right)$ random landmark pairs, we have with probability greater than $(1-\delta)$

$$\sup_{f\in\mathcal{F}} \left[\left|L(f,\mathcal{L}) - L(f)\right|\right] \leq \epsilon_1$$

Algorithm 2 FTUNE

Require: A family of transfer functions $\mathcal{F}$, a similarity function K, a loss function L.
Ensure: An optimal transfer function $f^* \in \mathcal{F}$.

1: Select d landmark pairs $\mathcal{L}$.
2: **for all** $f \in \mathcal{F}$ **do**
3: $w_f \leftarrow$ LEARN-DISSIM$(K, \mathcal{L}, f)$, $L_f \leftarrow L(f, \mathcal{L})$
4: **end for**
5: $f^* \leftarrow \arg\min_{f\in\mathcal{F}} L_f$
6: **return** f^*.

Intelligent choice of landmark points

- If landmarks are clumped together, then all points will get a similar embedding and linear separation would be impossible

Intelligent choice of landmark points

- If landmarks are clumped together, then all points will get a similar embedding and linear separation would be impossible
- Thus we promote *diversity* among the landmarks as a heuristic on small datasets

Intelligent choice of landmark points

- If landmarks are clumped together, then all points will get a similar embedding and linear separation would be impossible
- Thus we promote *diversity* among the landmarks as a heuristic on small datasets
- On large datasets FTUNE itself is able to recover the best transfer function as it does not over-fit

Intelligent choice of landmark points

- If landmarks are clumped together, then all points will get a similar embedding and linear separation would be impossible
- Thus we promote *diversity* among the landmarks as a heuristic on small datasets
- On large datasets FTUNE itself is able to recover the best transfer function as it does not over-fit

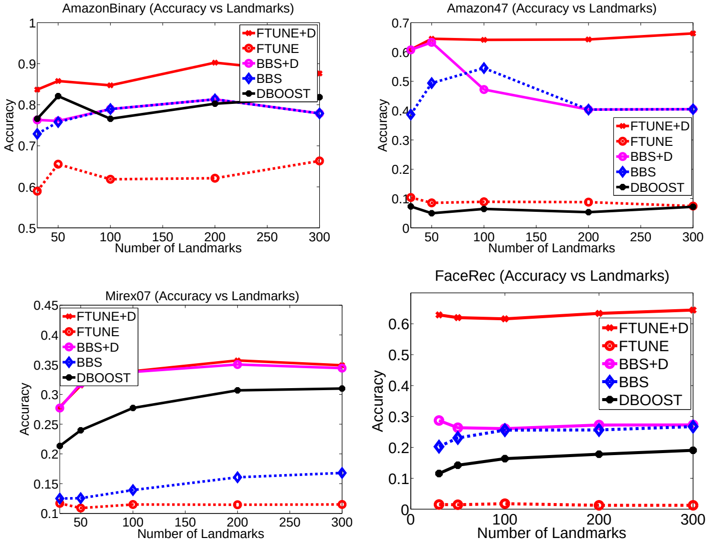
Algorithm 3 DSELECT

Require: A training set T.
Ensure: A set of n landmark pairs.

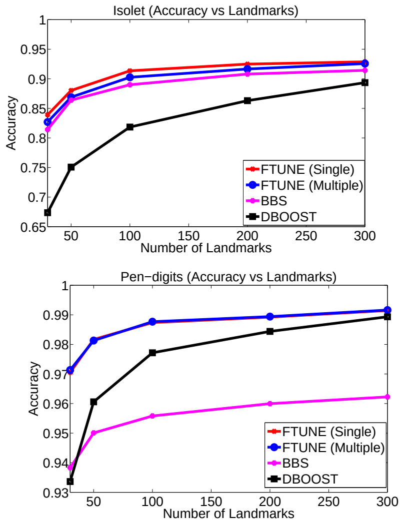
$S \leftarrow \textsc{random-element}(T), \mathcal{L} \leftarrow \emptyset$
for $j = 2$ **to** n **do**
  $z \leftarrow \arg\min_{x \in T} \sum_{x' \in S} K(x, x')$.
  $S \leftarrow S \cup \{z\}, \quad T \leftarrow T \backslash \{z\}$
end for
for $j = 1$ **to** n **do**
  Sample z_1, z_2 from S with replacement s.t. $\ell(z_1) = 1,\ \ell(z_2) = -1$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(z_1, z_2)\}$
end for
return $\mathcal{L}$

Outline

Results


AmazonBinary (Accuracy vs Landmarks)
Accuracy
Number of Landmarks
FTUNE+D
FTUNE
BBS+D
BBS
DBOOST
Amazon47 (Accuracy vs Landmarks)
Mirex07 (Accuracy vs Landmarks)
FaceRec (Accuracy vs Landmarks)

Results


Isolet (Accuracy vs Landmarks)
Accuracy
Number of Landmarks
Letters (Accuracy vs Landmarks)
Pen−digits (Accuracy vs Landmarks)
Opt−digits (Accuracy vs Landmarks)
FTUNE (Single)
FTUNE (Multiple)
BBS
DBOOST

Discussion

- BBS performs reasonably well for small landmarking sizes while DBOOST performs well for large landmarking sizes.

Discussion

- BBS performs reasonably well for small landmarking sizes while DBOOST performs well for large landmarking sizes.
- In contrast, our method consistently outperforms the existing methods in both the scenarios.

Discussion

- BBS performs reasonably well for small landmarking sizes while DBOOST performs well for large landmarking sizes.
- In contrast, our method consistently outperforms the existing methods in both the scenarios.
- Since FTUNE selects its output by way of validation, it is susceptible to over-fitting on small datasets.

Discussion

- BBS performs reasonably well for small landmarking sizes while DBOOST performs well for large landmarking sizes.
- In contrast, our method consistently outperforms the existing methods in both the scenarios.
- Since FTUNE selects its output by way of validation, it is susceptible to over-fitting on small datasets.
- In these cases, DSELECT (intuitively) removes redundancies in the landmark points thus allowing FTUNE to recover the best transfer function.

Thanks

Preprint available at
`http://www.cse.iitk.ac.in/users/purushot/`

Outline

References I

Arora, S., Babai, L., Stern, J., and Sweedyk, Z. (1997).
The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations.
Journal of Computer and System Sciences, 54(2):317–331.

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

Garey, M. R. and Johnson, D. (1979).
Computers and Intractability: A Guide to the theory of NP-Completeness.
Freeman, San Francisco.

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

References II

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

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

Kar, P. and Jain, P. (2011).
Similarity-based Learning via Data Driven Embeddings.
In *25th Annual Conference on Neural Information Processing Systems*.
(to appear).

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

Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., and Anthony, M. (1998).
Structural Risk Minimization Over Data-Dependent Hierarchies.
IEEE Transactions on Information Theory, 44(5):1926–1940.

References III

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.

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.