Similarity-based Learning via Data Driven Embeddings*

Purushottam Kar\textsuperscript{1} \quad Prateek Jain\textsuperscript{2}

\textsuperscript{1}Indian Institute of Technology
Kanpur

\textsuperscript{2}Microsoft Research India
Bengaluru

November 3, 2011

*To appear in the proceedings of NIPS 2011
Outline

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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/

P. Kar and P. Jain (IITK/MSRI)

Similarity-based Learning
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/

P. Kar and P. Jain (IITK/MSRI)
Digit Classification†

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

† P. Kar and P. Jain (IITK/MSRI)
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/
Learning

Spam mail detection

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 ...
P. Kar and P. Jain (IITK/MSRI)
Learning

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
Learning

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 ...
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 $D$ 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.

$$\Pr_{x \sim D}[\hat{\ell}(x) \neq \ell(x)] < \epsilon$$
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 $\ell : \mathcal{X} \to \{-1, +1\}$.
- 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 $\hat{\ell} : \mathcal{X} \to \{-1, +1\}$ such that it mostly gives out the true labels.

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

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

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 \( \mathbb{R}^d \)
  \[ \Phi : \mathcal{X} \to \mathbb{R}^d \]
  - 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 \( \mathbb{R}^d \)
- 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 (e.g. BoW for text)
- SOLUTION 2: Work with some distance/similarity function over 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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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

\[
\hat{\ell}(x) = \ell(\text{NN}(x))
\]
\[
\text{NN}(x) = \arg \max_{x' \in S} [K(x, x')]
\]
Classical algorithms that learn with similarities

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

- Nearest neighbor classification

  $$\hat{\ell}(x) = \ell(\text{NN}(x))$$
  $$\text{NN}(x) = \arg\max_{x' \in S} [K(x, x')]$$

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

  $$\hat{\ell}(x) = \text{sgn}(\langle w, x \rangle) \quad \text{for some } w \in \mathbb{R}^d$$
Classical algorithms that learn with similarities

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

$$\hat{\ell}(x) = \ell(\text{NN}(x))$$

$$\text{NN}(x) = \arg\max_{x' \in S} [K(x, x')]$$

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

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

$$\hat{\ell}(x) = \text{sgn} \left( \sum_{x' \in S} \alpha(x') K(x, x') \ell(x') \right)$$

$$K(x, x') = \langle x, x' \rangle$$

$$w = \sum_{x' \in S} \alpha(x') \ell(x')$$

SVM allows use of arbitrary Positive semi-definite kernels

$$\hat{\ell}(x) = \text{sgn} \left( \sum_{x' \in S} \alpha \text{SVM}(x') K(x, x') \ell(x') \right)$$
Classical algorithms that learn with similarities

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

Nearest neighbor classification

$$\hat{\ell}(x) = \ell(\text{NN}(x))$$

$$\text{NN}(x) = \arg\max_{x' \in S} \left[ K(x, x') \right]$$

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

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

- SVM allows use of arbitrary Positive semi-definite kernels

$$\hat{\ell}(x) = \text{sgn} \left( \sum_{x' \in S} \alpha_{\text{SVM}}(x') K(x, x') \ell(x') \right)$$
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 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]
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].
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
A suitable similarity should intuitively give better classifier performance.
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
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.
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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
Outline

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
What is a good similarity function?

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

Definition ([Balcan and Blum, 2006]):
A similarity function $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be $(\epsilon, \gamma)$-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 $x \sim D$ satisfies

$$E_{x' \sim D, \ell(x')} = \ell(x), \quad x'' \sim D, \ell(x'') \neq \ell(x) \quad [w(x') K(x, x') - w(x'') K(x, x'')] \geq \gamma$$

In other words, according to the similarity function, most points, on average, are more similar to points of the same label.
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 $K : X \times X \rightarrow \mathbb{R}$ is said to be $(\epsilon, \gamma)$-good for a classification problem if for some weighing function $w : X \rightarrow [-1, 1]$, at least a $(1 - \epsilon)$ probability mass of examples $x \sim D$ satisfies

$$E_{x' \sim D}, \ell(x') = \ell(x)$$

$$x'' \sim D, \ell(x'') \neq \ell(x)$$

$$w(x')K(x, x') - w(x'')K(x, x'') \geq \gamma$$

In other words, according to the similarity function, most points, on average, are more similar to points of the same label.
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 $(\epsilon, \gamma)$-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

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

$$\mathbb{E}_{x'' \sim \mathcal{D}, \ell(x'') \neq \ell(x)}$$
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 $(\epsilon, \gamma)$-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

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

- 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 $(\epsilon, \gamma)$-good similarity function, for any $\delta > 0$, given $n = \frac{16}{\gamma^2} \log \frac{2}{\delta}$ 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 $1 - \delta$,

$$
\hat{\ell}(x) = \text{sgn} \left( \sum_{i=1}^n w(x_i)\ell(x_i)K(x, x_i) \right)
$$
Learning with a good similarity function

Theorem ([Balcan and Blum, 2006])

Given an \((\epsilon, \gamma)\)-good similarity function, for any \(\delta > 0\), given \(n = \frac{16}{\gamma^2} \log \frac{2}{\delta}\) 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 \(1 - \delta\),

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

- 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 $(\epsilon, \gamma)$-good similarity function, for any $\delta > 0$, given $n = \frac{16}{\gamma^2} \log \frac{2}{\delta}$ 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 $1 - \delta$,

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

- 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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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 $(\epsilon, \gamma, B)$-good for a classification problem if there exist two class conditional probability distributions $\tilde{D}_+$ and $\tilde{D}_-$ such that for all $x \in \mathcal{X}$, $\frac{\tilde{D}_+(x)}{D(x)} < \sqrt{B}$ and $\frac{\tilde{D}_-(x)}{D(x)} < \sqrt{B}$, such that at least a $(1 - \epsilon)$ probability mass of examples $x \sim D$ satisfies

$$\Pr_{x' \sim \tilde{D}_+, x'' \sim \tilde{D}_-} \left[ \ell(x) (\ell(x')d(x, x') - \ell(x'')d(x, x'')) < 0 \right] \geq \frac{1}{2} + \gamma$$
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 $(\epsilon, \gamma, B)$-good for a classification problem if there exist two class conditional probability distributions $\tilde{D}_+$ and $\tilde{D}_-$ such that for all $x \in \mathcal{X}$, $\frac{\tilde{D}_+(x)}{D(x)} < \sqrt{B}$ and $\frac{\tilde{D}_-(x)}{D(x)} < \sqrt{B}$, such that at least a $(1 - \epsilon)$ probability mass of examples $x \sim D$ satisfies

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

- 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 \((\epsilon, \gamma, B)\)-good for a classification problem if there exist two class conditional probability distributions \( \tilde{D}_+ \) and \( \tilde{D}_- \) such that for all \( x \in \mathcal{X} \), \( \frac{\tilde{D}_+(x)}{D(x)} < \sqrt{B} \) and \( \frac{\tilde{D}_-(x)}{D(x)} < \sqrt{B} \), such that at least a \((1 - \epsilon)\) probability mass of examples \( x \sim D \) satisfies

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

- 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 \((\epsilon, \gamma, B)\)-good distance function, for any \(\delta > 0\), given
\[ n = \frac{4B^2}{\gamma^2} \log \frac{1}{\delta} \] pairs of positive and negatively labeled points \( (x^+_i, x^-_i)_{i=1}^n \),
the classifier \( \hat{\ell} \) defined below has error at margin \( \frac{\gamma}{B} \) no more than \( \epsilon + \delta \)
with probability greater than \( 1 - \delta \),
\[
\hat{\ell}(x) = \text{sgn} \left( \sum_{i=1}^n \beta_i \text{sgn} \left( d(x, x^+_i) - d(x, x^-_i) \right) \right), \sum_{i=1}^n \beta_i = 1, \beta_i \geq 0
\]
**Theorem ([Wang et al., 2007])**

Given an $(\epsilon, \gamma, B)$-good distance function, for any $\delta > 0$, given
\[ n = \frac{4B^2}{\gamma^2} \log \frac{1}{\delta} \]
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 $1 - \delta$,

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

This naturally lends itself to a boosting-like implementation.
Learning with a good distance function

Theorem ([Wang et al., 2007])

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

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

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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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 $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be $(\epsilon, \gamma, B)$-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 $x \sim D$ satisfies

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

With appropriate setting of the weighing function and the transfer function, the previous two models can be recovered.
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 $(\epsilon, \gamma, B)$-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

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

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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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.
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 $T'$, $\ell_{\text{lin}} \leftarrow \text{LEARN-LINEAR}(T')$
5. Let $\hat{\ell} : \mathcal{X} \rightarrow \{-1, +1\}$ be defined as $\hat{\ell} : x \mapsto \ell_{\text{lin}} (\Phi_\mathcal{L}(x))$
6. return $\hat{\ell}$
Learning with data-sensitive notions of suitability

**Algorithm 1** LEARN-DISSIM

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

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

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

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

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

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

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

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 $\mathcal{L} = (x^+_i, x^-_i)_{i=1}^n$, a good transfer function $f$.

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

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 $T'$, $\ell_{\text{lin}} \leftarrow \text{LEARN-LINEAR}(T')$

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

6. return $\hat{l}$

- 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]
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 \((\epsilon, B)\)-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 \(w : \mathcal{X} \times \mathcal{X} \rightarrow [-B, B]\),

\[
E_{x \sim D} [L(G(x))] \leq \epsilon
\]

where

\[
G(x) = \mathbb{E}_{x' \sim D, \ell(x')=\ell(x), x'' \sim D, \ell(x'') \neq \ell(x)} [w(x', x'') f(K(x, x') - K(x, x''))]
\]
Working with surrogate loss functions

**Definition (K. and Jain, 2011)**

A similarity function is said to be $(\epsilon, B)$-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 $w : \mathcal{X} \times \mathcal{X} \rightarrow [-B, B]$, 

\[
E_{x \sim \mathcal{D}} [L(G(x))] \leq \epsilon
\]

where

\[
G(x) = \mathbb{E}_{
\begin{align*}
x' &\sim \mathcal{D}, \ell(x') = \ell(x) \\
x'' &\sim \mathcal{D}, \ell(x'') \neq \ell(x)
\end{align*}
\]

\[
[w(x', x'') f(K(x, x') - K(x, x''))] 
\]

**Theorem (K. and Jain, 2011)**

If $K$ is an $(\epsilon, B)$-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 $d = (16B^2C_L^2/\epsilon_1^2) \ln(4B/\delta\epsilon_1)$ landmark pairs, the expected loss of the classifier $\hat{\ell}(x)$ returned by LEARN-DISSIM with respect to $L$ satisfies

\[
E_x \left[ L(\hat{\ell}(x)) \right] \leq \epsilon + \epsilon_1.
\]
Outline

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
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 $L(f) = \mathbb{E}_{x \sim \mathcal{D}} [L(G(x))]$ 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 $\mathcal{L}$. 
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 $L(f) = \mathbb{E}_{x \sim D}[L(G(x))]$ 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 $\epsilon$-net over $\mathcal{F}$ with respect to the infinity norm at scale $r = \frac{\epsilon_1}{4CLB}$.

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 $(1 - \delta)$

$$\sup_{f \in \mathcal{F}} \left[ \| L(f, \mathcal{L}) - L(f) \| \right] \leq \epsilon_1$$
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 $\epsilon$-net over $\mathcal{F}$ with respect to the infinity norm at scale $r = \frac{\epsilon_1}{4C_LB}$.

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 $(1 - \delta)$

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

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: \quad $w_f \leftarrow \text{LEARN-DISSIM}(K, \mathcal{L}, f)$,
4: \quad $L_f \leftarrow L(f, \mathcal{L})$
5: end for
6: $f^* \leftarrow \arg \min_{f \in \mathcal{F}} L_f$
7: 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.

Algorithm 3

DSELECT

Require:
A training set $T$.

Ensure:
A set of $n$ landmark pairs.

1: $S \leftarrow \text{RANDOM-ELEMENT}(T)$, $L \leftarrow \emptyset$
2: for $j = 2$ to $n$ do
3:   $z \leftarrow \arg \min_{x \in T} \sum_{x' \in S} K(x, x')$
4:   $S \leftarrow S \cup \{z\}$, $T \leftarrow T \setminus \{z\}$
5: end for
6: for $j = 1$ to $n$ do
7:   Sample $z_1, z_2$ from $S$ with replacement s.t.
8:     $\ell(z_1) = 1$, $\ell(z_2) = -1$
9:   $L \leftarrow L \cup \{(z_1, z_2)\}$
10: end for
11: return $L$
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.
1: \( S \leftarrow \text{RANDOM-ELEMENT}(T), \mathcal{L} \leftarrow \emptyset \)
2: for \( j = 2 \) to \( n \) do
3: \( z \leftarrow \arg \min_{x \in T} \sum_{x' \in S} K(x, x') \).
4: \( S \leftarrow S \cup \{z\}, \ T \leftarrow T \setminus \{z\} \)
5: end for
6: for \( j = 1 \) to \( n \) do
7: Sample \( z_1, z_2 \) from \( S \) with replacement s.t. \( \ell(z_1) = 1, \ell(z_2) = -1 \)
8: \( \mathcal{L} \leftarrow \mathcal{L} \cup \{(z_1, z_2)\} \)
9: end for
10: return \( \mathcal{L} \)
Outline

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References
Results

AmazonBinary (Accuracy vs Landmarks)

Amazon47 (Accuracy vs Landmarks)

Mirex07 (Accuracy vs Landmarks)

FaceRec (Accuracy vs Landmarks)
Results

- Isolet (Accuracy vs Landmarks)
- Letters (Accuracy vs Landmarks)
- Pen-digits (Accuracy vs Landmarks)
- Opt-digits (Accuracy vs Landmarks)
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

1. An Introduction to Learning
2. A Brief History of Learning with Similarities
3. Learning with Suitable Similarities
   - Learning with a Suitable Similarity Function
   - Learning with a Suitable Distance Function
4. Data-sensitive Notions of Suitability
   - Learning with Data-sensitive Notions of Suitability
   - Learning the Best Notion of Suitability
   - Results
5. References


References II


