Similarity and kernels in machine learning

Zalán Bodó

Faculty of Mathematics and Computer Science,
Babeș–Bolyai University, Cluj-Napoca/Kolozsvár/Klausenburg

Derby, June 2016

Overview of the presentation

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

\+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

> *"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"*
>
> [Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

\+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

+ semi-supervised learning (classification)

Machine learning

Arthur Samuel, 1959: *"field of study that gives computers the ability to learn without being explicitly programmed"*

"[. . .] machine learning is now an independent and mature field that has moved beyond psychologically or neurally inspired algorithms towards providing foundations for a theory of learning that is rooted in statistics and functional analysis"

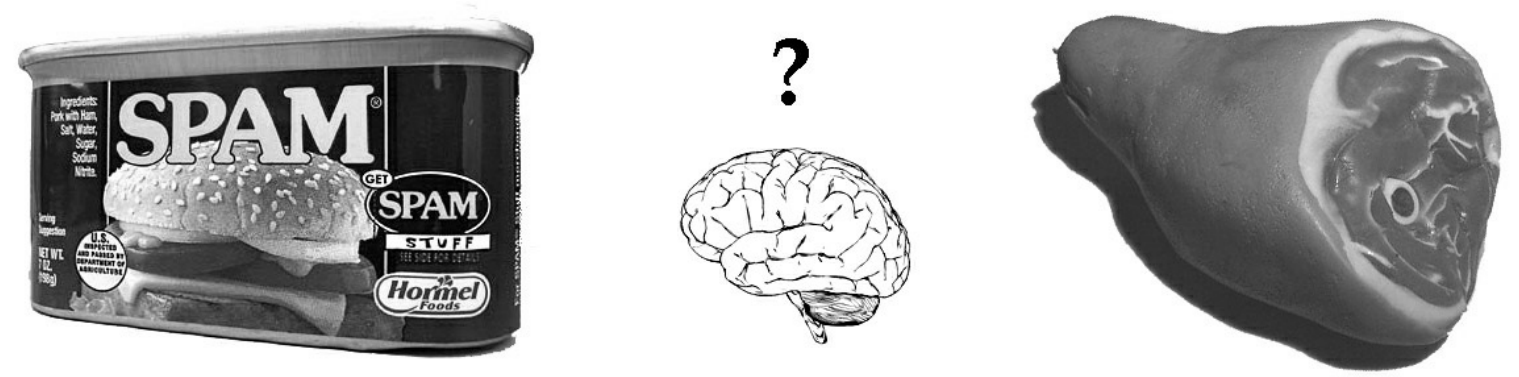
[Jäkel et al., 2007]

Machine learning =

- supervised learning – classification, regression
- unsupervised learning – clustering, density estimation
- reinforcement learning

+ semi-supervised learning (classification)

Example: Content-based spam filtering




```
Dear Sir,
Herewith you will find our Cover Letter which includes our Corporate
Rates 2015 in case you need host if you visit PERU someday, for you,
staff and guests.
Please note our services and benefits, for example airport-hotel and
back for just US$ 27, breakfast buffet, wifi and ethernet.
Hoping to be of your preference, we remain,
Cordially yours,
--
*Maria del Carmen Ovalle Reiley*
Ejecutiva de Ventas Corporativa
*Sonesta Posadas del Inca Miraflores - Perú*
Tel.: (511) 241-7688 Anexo 1013 - Dir: Calle Alcanfores No 329
Fax: (511) 447-1164
Reservas: (511) 712-6060
*www.ghlhoteles.com <http://www.ghlhoteles.com> - www.sonesta.com
<http://www.sonesta.com>*
*Operado por GHL Hoteles*

       Antes de imprimir este e-mail piense si es necesario: El medio ambiente
es responsabilidad de todos
```

Similarity. Similarity in (machine) learning

- similarity is fundamental to learning
- Shepard: in each individual there is an *"internal metric of similarity between possible situations"* [Shepard, 1987]
- generalization is based on similarity between situations/events/objects/...
- learning = generalize...
  (a) supervised scenarios: ...from labeled to unlabeled data
  (b) unsupervised scenarios: ...from familiar to novel data

"The fundamental challenge confronted by any system that is expected to generalize from familiar to unfamiliar stimuli is how to estimate similarity over stimuli in a principled and feasible manner."

[Shahbazi et al., 2016]

Similarity. Similarity in (machine) learning

- similarity is fundamental to learning
- Shepard: in each individual there is an *"internal metric of similarity between possible situations"* [Shepard, 1987]
- generalization is based on similarity between situations/events/objects/...
- learning = generalize...
  (a) supervised scenarios: ...from labeled to unlabeled data
  (b) unsupervised scenarios: ...from familiar to novel data

"The fundamental challenge confronted by any system that is expected to generalize from familiar to unfamiliar stimuli is how to estimate similarity over stimuli in a principled and feasible manner."

[Shahbazi et al., 2016]

Similarity of...

- **sets**, e.g. Jaccard similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- **sequences**, e.g. edit (Levenshtein) distance-based similarity

$$E(s, t) = 1 - \frac{edist(s, t)}{\max(|s|, |t|)}$$

- **vectors**, e.g. cosine similarity (= normalized dot product)

$$C(\boldsymbol{x}, \boldsymbol{z}) = \frac{\boldsymbol{x}'\boldsymbol{z}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}$$

- ...

Similarity of...

- **sets**, e.g. Jaccard similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- **sequences**, e.g. edit (Levenshtein) distance-based similarity

$$E(s, t) = 1 - \frac{edist(s, t)}{\max(|s|, |t|)}$$

- **vectors**, e.g. cosine similarity (= normalized dot product)

$$C(x, z) = \frac{x'z}{\|x\|\|z\|}$$

- ...

Similarity of...

- **sets**, e.g. Jaccard similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- **sequences**, e.g. edit (Levenshtein) distance-based similarity

$$E(s, t) = 1 - \frac{edist(s, t)}{\max(|s|, |t|)}$$

- **vectors**, e.g. cosine similarity (= normalized dot product)

$$C(\boldsymbol{x}, \boldsymbol{z}) = \frac{\boldsymbol{x}'\boldsymbol{z}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}$$

- ...

Kernels

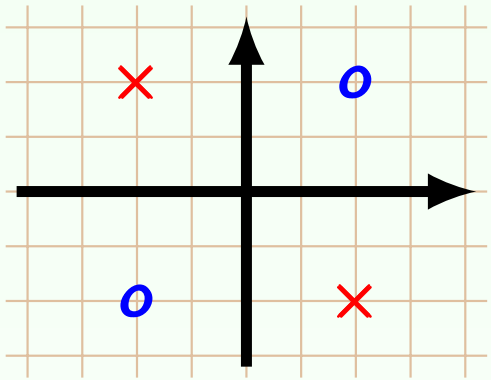


Figure : XOR problem: separate the o's from the x's

- **Marvin Minsky, Seymour Papert. Perceptrons: an introduction to computational geometry. MIT Press, Cambridge, Mass., 1969.** – a single artificial neuron/perceptron (= lin. class.) cannot solve the problem
- **M. A. Aizerman, E. M. Braverman, L. I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, vol. 25, pp. 821–837, 1964.** – use kernels!

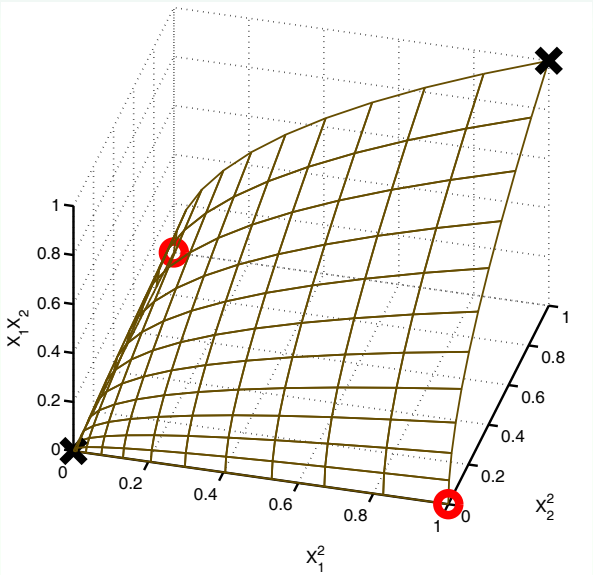


Figure : Using the polynomial kernel

- map the points using the function $\phi(\boldsymbol{x}) = [x_1^2 \; x_2^2 \; \sqrt{2}x_1x_2]'$
- this is equivalent to using $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle = (\boldsymbol{x}'\boldsymbol{z})^2$ (= polynomial kernel)
- polynomial kernel: *link the features using logical AND* (size of the group of "linked" features is determined by the order of the kernel)

Hilbert spaces

- we work in special Hilbert spaces
- called **reproducing kernel Hilbert spaces** (RKHS)

Def. Hilbert space

A Hilbert space is an **inner product** space that is also a complete metric space with respect to the **distance function** induced by the inner product.

Norm is defined as

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

Distance is defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle}$$

Example

Euclidean space: $\langle \boldsymbol{x}, \boldsymbol{z} \rangle = \sum_{i=1}^{d} x_i z_i, \quad \|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$

Kernel methods

- **1909**: James Mercer – *any continuous symmetric, positive semi-definite kernel function can be expressed as a dot product in a high-dimensional space* [Mercer, 1909]
- **1964**: Aizerman, Braverman and Rozonoer – first application [Aizerman et al., 1964]
- **1992**: Boser, Guyon and Vapnik – famous application (SVM) [Boser et al., 1992]
- linear algorithms $\rightarrow$ non-linear algorithms
- feature mapping: $\phi : X \rightarrow \mathcal{H}$ $(\phi : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2})$
- kernels: $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle = \phi(\boldsymbol{x})' \phi(\boldsymbol{z})$
- *covers all geometric constructions that can be formulated in terms of angles, lengths and distances*

Kernel trick

Given an algorithm which is formulated in terms of a positive definite kernel $k(\cdot, \cdot)$, one can construct an alternative algorithm by replacing $k(\cdot, \cdot)$ by another positive definite kernel $\widetilde{k}(\cdot, \cdot)$.

Kernel methods

- **1909**: James Mercer – *any continuous symmetric, positive semi-definite kernel function can be expressed as a dot product in a high-dimensional space* [Mercer, 1909]
- **1964**: Aizerman, Braverman and Rozonoer – first application [Aizerman et al., 1964]
- **1992**: Boser, Guyon and Vapnik – famous application (SVM) [Boser et al., 1992]
- linear algorithms $\rightarrow$ non-linear algorithms
- feature mapping: $\phi : X \rightarrow \mathcal{H}$ $(\phi : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2})$
- kernels: $k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle = \phi(\boldsymbol{x})' \phi(\boldsymbol{z})$
- *covers all geometric constructions that can be formulated in terms of angles, lengths and distances*

Kernel trick

Given an algorithm which is formulated in terms of a positive definite kernel $k(\cdot, \cdot)$, one can construct an alternative algorithm by replacing $k(\cdot, \cdot)$ by another positive definite kernel $\widetilde{k}(\cdot, \cdot)$.

Examples of general purpose kernels

- linear:

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{z}$$

- polynomial:

$$k(\boldsymbol{x}, \boldsymbol{z}) = (a\boldsymbol{x}'\boldsymbol{z} + b)^c$$

- Gaussian (RBF):

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_2^2\right)$$

Examples of general purpose kernels

- linear:

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{z}$$

- polynomial:

$$k(\boldsymbol{x}, \boldsymbol{z}) = (a\boldsymbol{x}'\boldsymbol{z} + b)^c$$

- Gaussian (RBF):

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_2^2\right)$$

Examples of general purpose kernels

- linear:

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{z}$$

- polynomial:

$$k(\boldsymbol{x}, \boldsymbol{z}) = (a\boldsymbol{x}'\boldsymbol{z} + b)^c$$

- Gaussian (RBF):

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_2^2\right)$$

Kernels and similarities

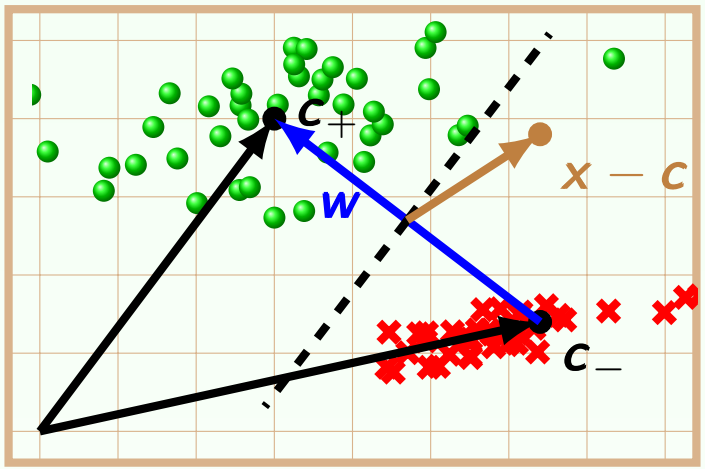
kernel	similarity
real-valued	real-valued
symmetric	not necessarily symmetric
positive definite	not necessarily p.d.
$k(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\left[k(\boldsymbol{x},\boldsymbol{x}) + k(\boldsymbol{z},\boldsymbol{z}) - \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{z})\|^2\right]$	$sim(\boldsymbol{x},\boldsymbol{z})$ = inverse of the distance between $\boldsymbol{x}$ and $\boldsymbol{z}$
$k(\boldsymbol{x},\boldsymbol{z}) = \langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\rangle$	
= the cosine similarity of the mapped vectors, provided they are normalized	

Kernels and similarities

kernel	similarity
real-valued	real-valued
symmetric	not necessarily symmetric
positive definite	not necessarily p.d.
$k(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\left[k(\boldsymbol{x},\boldsymbol{x}) + k(\boldsymbol{z},\boldsymbol{z}) - \|\phi(\boldsymbol{x}) - \phi(\boldsymbol{z})\|^2\right]$	$sim(\boldsymbol{x},\boldsymbol{z})$ = inverse of the distance between $\boldsymbol{x}$ and $\boldsymbol{z}$
$k(\boldsymbol{x},\boldsymbol{z}) = \langle\phi(\boldsymbol{x}),\phi(\boldsymbol{z})\rangle$	
= the cosine similarity of the mapped vectors, provided they are normalized	

$\Rightarrow$

A sample/simple method: prototype learning



- class centers (centroids, prototypes):

$$\boldsymbol{c}_+ = \frac{1}{N_+} \sum_{\boldsymbol{x}_i \in X_+} \boldsymbol{x}_i$$

$$\boldsymbol{c}_- = \frac{1}{N_-} \sum_{\boldsymbol{x}_i \in X_-} \boldsymbol{x}_i$$

- define the following vectors: $\boldsymbol{w} = \boldsymbol{c}_+ - \boldsymbol{c}_-$ and $\boldsymbol{c} = (\boldsymbol{c}_+ + \boldsymbol{c}_-)/2$
- then

$$\begin{aligned} y(\boldsymbol{x}) &= sgn\langle \boldsymbol{x} - \boldsymbol{c}, \boldsymbol{w} \rangle \\ &= sgn\left(\langle \boldsymbol{c}_+, \boldsymbol{x} \rangle - \langle \boldsymbol{c}_-, \boldsymbol{x} \rangle + b\right) \end{aligned}$$

with $b = \left(\|\boldsymbol{c}_-\|^2 - \|\boldsymbol{c}_+\|^2\right)/2$.

- using dot products between the $\boldsymbol{x}_i$s:

$$y(\boldsymbol{x}) = sgn\left(\frac{1}{N_+}\sum_{\boldsymbol{x}_i \in X_+} \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle - \frac{1}{N_-}\sum_{\boldsymbol{x}_i \in X_-} \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b\right)$$

where

$$b = \frac{1}{2}\left(\frac{1}{N_-^2}\sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X_-} \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \frac{1}{N_+^2}\sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X_+} \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle\right)$$

The representer theorem

Theorem (Schölkopf and Smola, 2002)

Let $\mathcal{H}$ be the feature space associated to a positive semi-definite kernel $k : X \times X \to \mathbb{R}$. Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function, and by $c : (X \times \mathbb{R}^2)^\ell \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer of the regularized risk

$$c((\boldsymbol{x}_1, y_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_\ell, y_\ell, f(\boldsymbol{x}_\ell))) + \Omega(\|f\|_{\mathcal{H}})$$

admits a representation of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

The "kernelization period"

199x – 200y

- 1992: SVM
- ?: kernel regularized least squares
- 1996: kernel PCA
- 1999: kernel Fisher discriminant analysis, transductive SVM
- 2001: kernel k-means clustering, kernel canonical correlation analysis, SVC (support vector clustering)
- 2005: first data-dependent non-parametric kernel, Laplacian regularized least squares, Laplacian SVM
- ...

The "kernelization period"

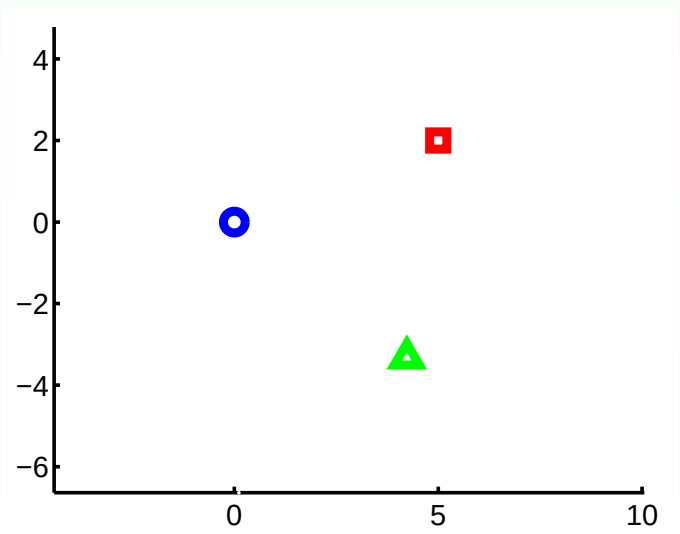
199x – 200y

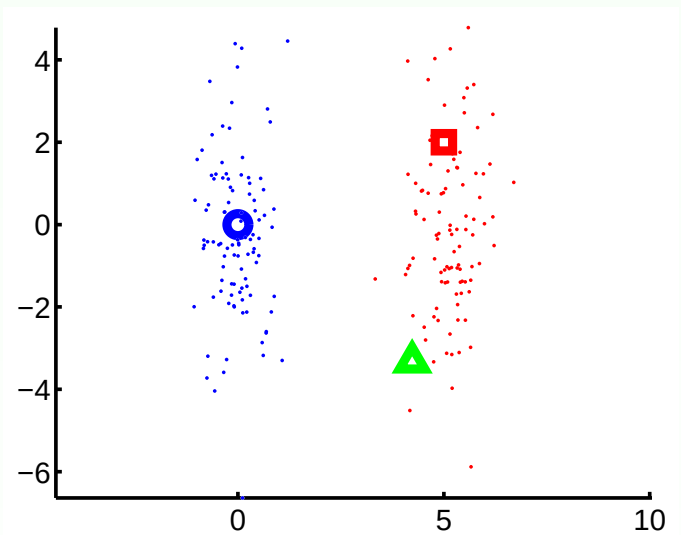
- 1992: SVM
- ?: kernel regularized least squares
- 1996: kernel PCA
- 1999: kernel Fisher discriminant analysis, transductive SVM
- 2001: kernel k-means clustering, kernel canonical correlation analysis, SVC (support vector clustering)
- 2005: first data-dependent non-parametric kernel, Laplacian regularized least squares, Laplacian SVM
- ...

Semi-supervised learning and kernels

Semi-supervised learning (SSL)

- supervised learning:
  $D = \{(\boldsymbol{x}_i, y_i) \,|\, \boldsymbol{x}_i \in X \subseteq \mathbb{R}^d, y_i \in \{-1, +1\}, i = 1, \ldots, \ell\}$;
  find $f : X \to \{-1, +1\}$ which agrees with D
- semi-supervised learning:
  $D = \{(\boldsymbol{x}_i, y_i) \,|\, i = 1, \ldots, \ell\} \cup \{\boldsymbol{x}_j \,|\, j = 1, \ldots, u\}$, $\ell \ll u$, $N = \ell + u$;
  - **inductive**: find $f : X \to \{-1, +1\}$ which agrees with D + use the information of D_U
  - **transductive**: find $f : D_U \to \{-1, +1\}$ by using $D = D_L \cup D_U$

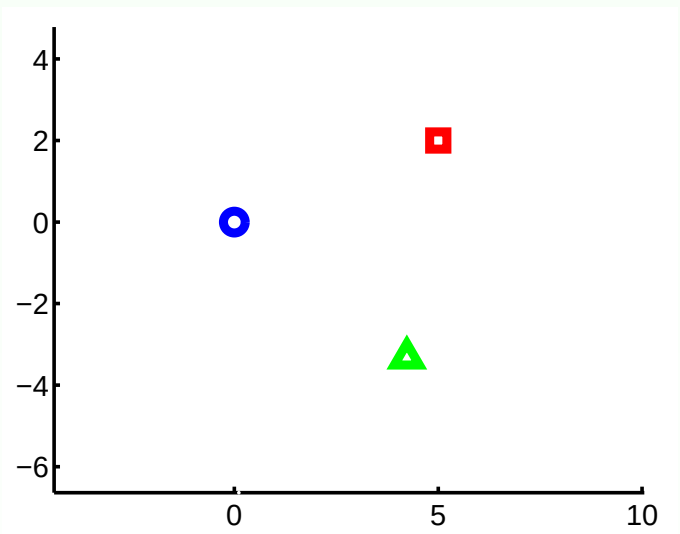


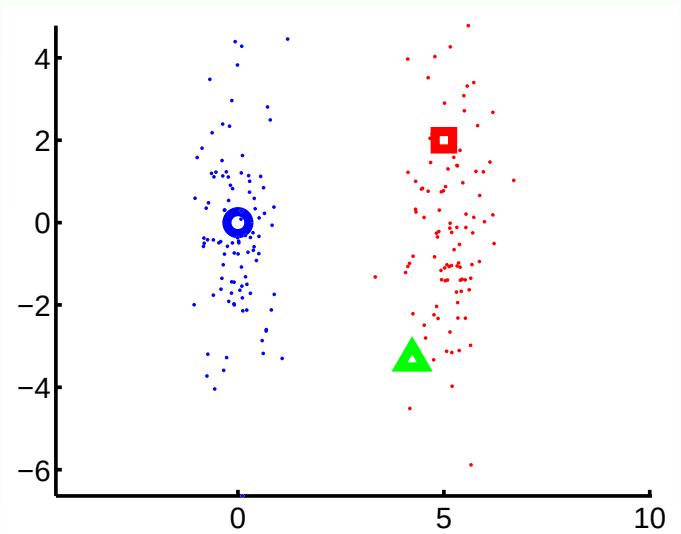


Semi-supervised learning and kernels

Semi-supervised learning (SSL)

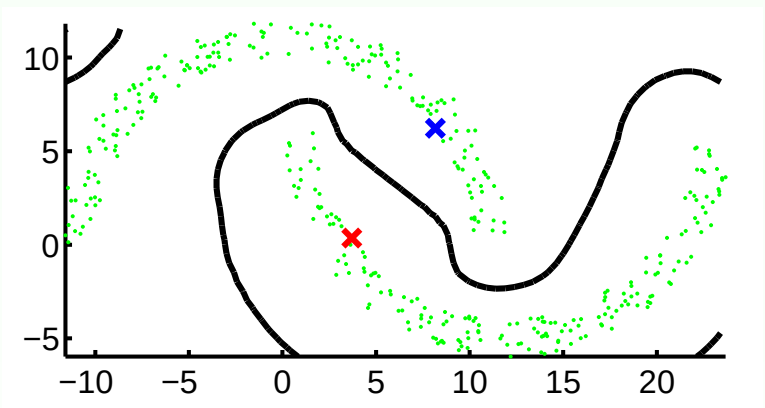
- supervised learning:
  $D = \{(\boldsymbol{x}_i, y_i) \,|\, \boldsymbol{x}_i \in X \subseteq \mathbb{R}^d, y_i \in \{-1, +1\}, i = 1, \ldots, \ell\}$;
  find $f : X \to \{-1, +1\}$ which agrees with D
- semi-supervised learning:
  $D = \{(\boldsymbol{x}_i, y_i) \,|\, i = 1, \ldots, \ell\} \cup \{\boldsymbol{x}_j \,|\, j = 1, \ldots, u\}$, $\ell \ll u$, $N = \ell + u$;
  - **inductive**: find $f : X \to \{-1, +1\}$ which agrees with D + use the information of D_U
  - **transductive**: find $f : D_U \to \{-1, +1\}$ by using $D = D_L \cup D_U$





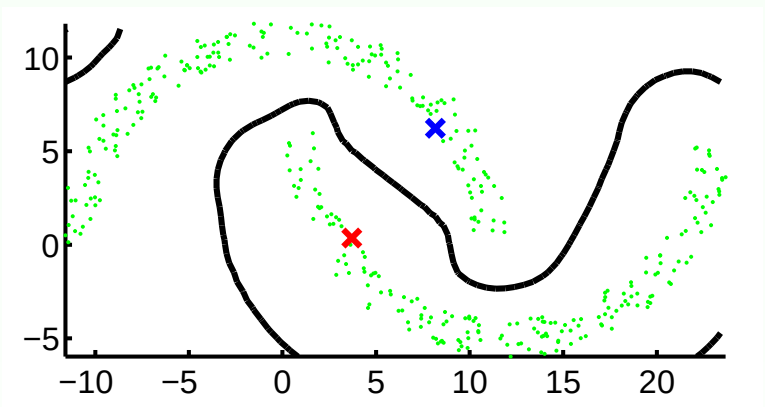
Assumptions in SSL

1. **smoothness assumption**: *If two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in a high density region are close, then so should be the corresponding outputs y_i and y_j.*
2. **cluster assumption**: *If two points are in the same cluster, they are likely to be of the same class.*
3. **manifold assumption** (a.k.a. graph-based learning): *The high dimensional data lie roughly on a low dimensional manifold.*



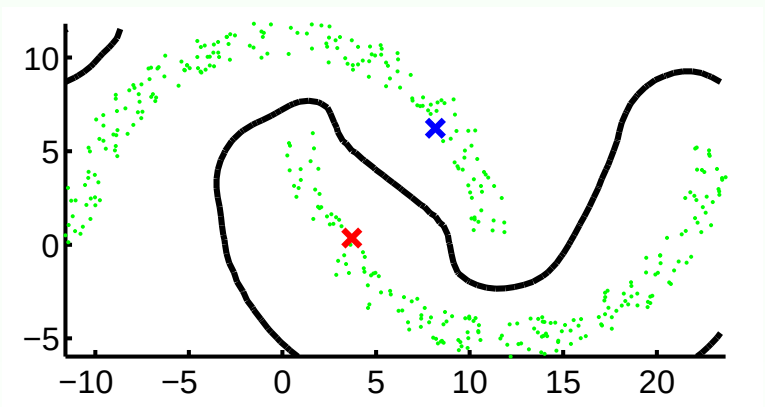
Assumptions in SSL

1. **smoothness assumption**: *If two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in a high density region are close, then so should be the corresponding outputs y_i and y_j.*
2. **cluster assumption**: *If two points are in the same cluster, they are likely to be of the same class.*
3. **manifold assumption** (a.k.a. graph-based learning): *The high dimensional data lie roughly on a low dimensional manifold.*



Assumptions in SSL

1. **smoothness assumption**: *If two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in a high density region are close, then so should be the corresponding outputs y_i and y_j.*
2. **cluster assumption**: *If two points are in the same cluster, they are likely to be of the same class.*
3. **manifold assumption** (a.k.a. graph-based learning): *The high dimensional data lie roughly on a low dimensional manifold.*



Humans and SSL

- humans do semi-supervised classification too
- 2007: experiment (Zhu and his colleagues), University of Wisconsin [Zhu et al., 2007]
- complex 3D shapes classified into two categories
- participants were told they see microscopic images of pollen particles from two fictitious flowers (*Belianthus* and *Nortulaca*)
- data given:
  - 2 labeled examples (each appearing 10 times in 20 trials)
  - test set of 21 evenly spaced unlabeled examples – to test the learned decision boundary
  - 3×230 unlabeled examples – the means are shifted away from the labeled examples (*left-shifted* or *right-shifted*)
  - test set of 21 evenly spaced unlabeled examples – to test whether the decision boundary has changed
- $\Rightarrow$ the learned decision boundary is determined by both labeled and unlabeled data

Data-dependent kernels

- supervised learning + data-dependent kernels = **semi-supervised learning**
- *conventional* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})$$

- *data-dependent* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}; D_1) \fallingdotseq k(\boldsymbol{x}, \boldsymbol{z}; D_2)$$

"$\fallingdotseq$" reads as *"not necessarily equal"*

Data-dependent kernels

- supervised learning + data-dependent kernels = **semi-supervised learning**
- *conventional* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})$$

- *data-dependent* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}; D_1) \fallingdotseq k(\boldsymbol{x}, \boldsymbol{z}; D_2)$$

"$\fallingdotseq$" reads as *"not necessarily equal"*

Data-dependent kernels

- supervised learning + data-dependent kernels = **semi-supervised learning**
- *conventional* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{x}, \boldsymbol{z})$$

- *data-dependent* kernels: given data sets $D_1 \neq D_2$, $\boldsymbol{x}, \boldsymbol{z} \in D_1 \cap D_2$

$$k(\boldsymbol{x}, \boldsymbol{z}; D_1) \fallingdotseq k(\boldsymbol{x}, \boldsymbol{z}; D_2)$$

"$\fallingdotseq$" reads as *"not necessarily equal"*

Reweighting cluster kernels

- idea borrowed from bagged cluster kernel [Weston et al., 2005; Chapelle et al., 2006]
- reweighting conventional kernels according to some clustering of the data [Bodó and Csató, 2010]
- kernel combinations: $\boldsymbol{K}_1 + \boldsymbol{K}_2$, $a \cdot \boldsymbol{K}$, $\boldsymbol{K}_1 \odot \boldsymbol{K}_2$
- cluster kernel: $\boldsymbol{K} = \boldsymbol{K}_{\mathrm{rw}} \odot \boldsymbol{K}_{\mathrm{b}}$ where
  - $\boldsymbol{K}_{\mathrm{b}}$ = base kernel (e.g. Gaussian, polynomial, etc.)
  - $\boldsymbol{K}_{\mathrm{rw}}$ = reweighting kernel
  - $\boldsymbol{K}$ = resulting cluster kernel used in the learning algorithm

$$k_{\mathrm{rw}}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{U}_{\cdot \boldsymbol{x}} - \boldsymbol{U}_{\cdot \boldsymbol{z}}\|^2}{2\sigma^2}\right)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \boldsymbol{U}'\boldsymbol{U} + \alpha \cdot \mathbf{1}\mathbf{1}', \quad \alpha \in [0, 1)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \beta \cdot \boldsymbol{U}'\boldsymbol{U} + \mathbf{1}\mathbf{1}', \quad \beta \in (0, \infty)$$

($\boldsymbol{U}$ = matrix of cluster membership vectors (columns) of size $\underbrace{K}_{\text{no. of clusters}} \times \underbrace{N}_{\text{no. of points}}$)

Reweighting cluster kernels

- idea borrowed from bagged cluster kernel [Weston et al., 2005; Chapelle et al., 2006]
- reweighting conventional kernels according to some clustering of the data [Bodó and Csató, 2010]
- kernel combinations: $\boldsymbol{K}_1 + \boldsymbol{K}_2$, $a \cdot \boldsymbol{K}$, $\boldsymbol{K}_1 \odot \boldsymbol{K}_2$
- cluster kernel: $\boldsymbol{K} = \boldsymbol{K}_{\mathrm{rw}} \odot \boldsymbol{K}_{\mathrm{b}}$ where
  - $\boldsymbol{K}_{\mathrm{b}}$ = base kernel (e.g. Gaussian, polynomial, etc.)
  - $\boldsymbol{K}_{\mathrm{rw}}$ = reweighting kernel
  - $\boldsymbol{K}$ = resulting cluster kernel used in the learning algorithm

$$k_{\mathrm{rw}}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{U}_{\cdot \boldsymbol{x}} - \boldsymbol{U}_{\cdot \boldsymbol{z}}\|^2}{2\sigma^2}\right)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \boldsymbol{U}'\boldsymbol{U} + \alpha \cdot \mathbf{1}\mathbf{1}', \quad \alpha \in [0, 1)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \beta \cdot \boldsymbol{U}'\boldsymbol{U} + \mathbf{1}\mathbf{1}', \quad \beta \in (0, \infty)$$

($\boldsymbol{U}$ = matrix of cluster membership vectors (columns) of size $\underbrace{K}_{\text{no. of clusters}} \times \underbrace{N}_{\text{no. of points}}$)

Reweighting cluster kernels

- idea borrowed from bagged cluster kernel [Weston et al., 2005; Chapelle et al., 2006]
- reweighting conventional kernels according to some clustering of the data [Bodó and Csató, 2010]
- kernel combinations: $\boldsymbol{K}_1 + \boldsymbol{K}_2$, $a \cdot \boldsymbol{K}$, $\boldsymbol{K}_1 \odot \boldsymbol{K}_2$
- cluster kernel: $\boldsymbol{K} = \boldsymbol{K}_{\mathrm{rw}} \odot \boldsymbol{K}_{\mathrm{b}}$ where
  - $\boldsymbol{K}_{\mathrm{b}}$ = base kernel (e.g. Gaussian, polynomial, etc.)
  - $\boldsymbol{K}_{\mathrm{rw}}$ = reweighting kernel
  - $\boldsymbol{K}$ = resulting cluster kernel used in the learning algorithm

$$k_{\mathrm{rw}}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{U}_{\cdot\boldsymbol{x}} - \boldsymbol{U}_{\cdot\boldsymbol{z}}\|^2}{2\sigma^2}\right)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \boldsymbol{U}'\boldsymbol{U} + \alpha \cdot \mathbf{1}\mathbf{1}', \quad \alpha \in [0, 1)$$

$$\boldsymbol{K}_{\mathrm{rw}} = \beta \cdot \boldsymbol{U}'\boldsymbol{U} + \mathbf{1}\mathbf{1}', \quad \beta \in (0, \infty)$$

($\boldsymbol{U}$ = matrix of cluster membership vectors (columns) of size $\underbrace{K}_{\text{no. of clusters}} \times \underbrace{N}_{\text{no. of points}}$)

Thank you!

References

Aizerman et al., 1964 M. A. Aizerman, E. M. Braverman, L. I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control, vol. 25, pp. 821–837, 1964.

Bodó and Csató, 2010 Z. Bodó, L. Csató. Hierarchical and Reweighting Cluster Kernels for Semi-Supervised Learning. Int. J. of Computers, Communications & Control, Vol. V, No. 4, pp. 469-476, 2010.

Boser et al., 1992 B. E. Boser, I. M. Guyon, V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. COLT, pp. 144–152, 1992.

Chapelle et al., 2006 **O. Chapelle, B. Schölkopf, A. Zien. Semi-Supervised Learning. MIT Press, 2006.**

Jäkel et al., 2007 F. Jäkel , B. Schölkopf , F. A. Wichmann. A Tutorial on Kernel Methods for Categorization. Journal of Mathematical Psychology 51(6), pp. 343–358, 2007.

Mercer, 1909 J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society, Series A, vol. 209, pp. 415–446, 1909.

Minsky and Papert, 1969 M. Minsky, S. Papert. Perceptrons: an introduction to computational geometry. MIT Press, Cambridge, Mass., 1969

Schölkopf and Smola, 2002 **B. Schölkopf, A. J. Smola. Learning with Kernels. MIT Press, Cambridge, Mass., 2002.**

Shahbazi et al., 2016 **R. Shahbazi, R. Raizada, S. Edelman. Similarity, kernels, and the fundamental constraints on cognition. Journal of Mathematical Psychology, vol. 70, pp. 21–34, 2016.**

Shepard, 1987 R. N. Shepard. Toward a universal law of generalization for psychological science. Science, 237, pp. 1317–1323, 1987.

Weston et al., 2005 J. Weston, C. Leslie, D. Zhou, A. Elisseeff, W. S. Noble. Semi-Supervised Protein Classification using Cluster Kernels. Bioinformatics, 21(15), pp. 3241–3247, 2005.

Zhu et al., 2007 X. Zhu, T. Rogers, R. Qian, C. Kalish. Humans perform semi-supervised classification too. AAAI, pp. 864–870, 2007.