Generative vs. Discriminative techniques for NLP

Nicolas Stroppa

NCLT, School of Computing, Dublin City University

11th January 2005

Outline

Goals

Goal 1: Be more familiar with discriminative techniques

- Classification, learning biases, memory-based approaches, kernel methods, re-ranking, etc.

Goal 2: Help you understand articles such as:

- M. Collins & T. Koo (2005). Discriminative Reranking for Natural Language Parsing. Computational Linguistics 31(1):25-69.
- B. Taskar, D. Klein, M. Collins, D. Koller & C. Manning (2004). Max-Margin Parsing. EMNLP 2004. (BP award.)
- W. Daelemans & A. van den Bosch (2005). Memory-based language processing.

Goals

Goal 3

- Understand the links between discriminative and generative techniques.

Goal 4

- Use our knowledge of NLP tasks (such as MT):
- . . . to cleverly use current ML techniques
- . . . to formalize (example-based) algorithms in ML terms.

A Machine Learning example

You see swans (s) and geese (g)

- bird 1: height 0.5m, 80% white (s)
- bird 2: height 0.7m, 72% white (s)
- bird 3: height 0.6m, 61% white (s)
- bird 4: height 0.4m, 69% white (g)
- bird 5: height 0.6m, 58% white (g)

Question

- Can you say if the following bird is a swan or a goose?
- bird: height 0.5m, 50% white

A Machine Learning example

You see swans (s) and geese (g)

- bird 1: height 0.5m, 80% white (s)
- bird 2: height 0.7m, 72% white (s)
- bird 3: height 0.6m, 61% white (s)
- bird 4: height 0.4m, 69% white (g)
- bird 5: height 0.6m, 58% white (g)

Question

- Can you say if the following bird is a swan or a goose?
- bird: height 0.5m, 50% white $\Rightarrow$ a goose

A second example

You see sick (s) and healthy (h) patients

- patient 1: temp 37, 0% spots (h)
- patient 2: temp 38, 20% spots (s)
- patient 3: temp 40, 20% spots (s)
- patient 4: temp 37, 50% spots (s)
- patient 5: temp 38, 5% spots (h)

Question

- Can you say if the following patient is sick?
- patient: temp 37.5, 10% spots

A second example

You see sick (s) and healthy (h) patients

- patient 1: temp 37, 0% spots (h)
- patient 2: temp 38, 20% spots (s)
- patient 3: temp 40, 20% spots (s)
- patient 4: temp 37, 50% spots (s)
- patient 5: temp 38, 5% spots (h)

Question

- Can you say if the following patient is sick?
- patient: temp 37.5, 10% spots $\Rightarrow$ healthy

Machine Learning tasks

Some tasks

- Written Caracters ⇒ Digits
- Emails ⇒ Spam/non-Spam
- Webpages ⇒ Theme (sports, economy, etc.)
- Word ⇒ yes/no (grammatical induction)

- Problem ⇒ Solution
- Question ⇒ Answer
- Request ⇒ Result
- *Input* ⇒ *Output*

Machine Learning is about *prediction*

NLP tasks and prediction

Inflectional analysis

marchaient ⇒ *MARCHER + Verb,Past,3P*

(graphemic string ⇒ lemma + set of features)

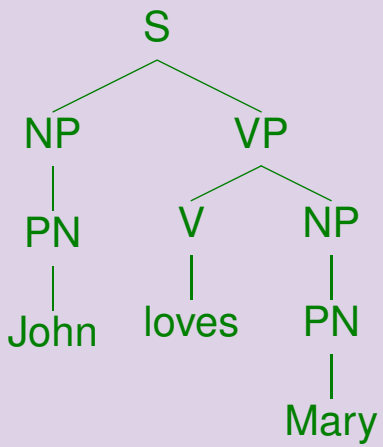
Pronunciation

live+Verb ⇒ */liv/*

(wordform + pos ⇒ phonetic string)

NLP tasks and prediction (2)

Syntactical analysis

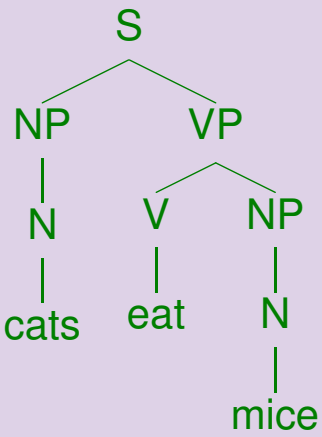


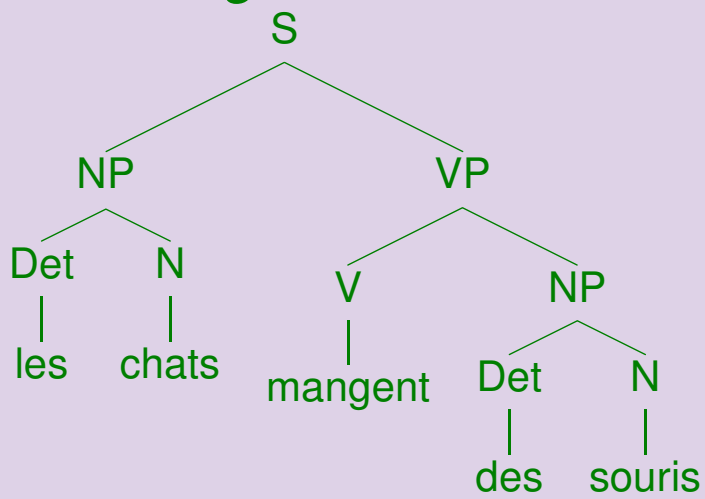
John/PN loves/V Mary/NP $\Rightarrow$
(tagged sentence $\Rightarrow$ syntactic tree)

NLP tasks and prediction (3)

Translation

cats eat mice ⇒ les chats mangent des souris





(syntactic tree (source language) ⇒ syntactic tree (target language))

Outline

Discriminate between swans and geese

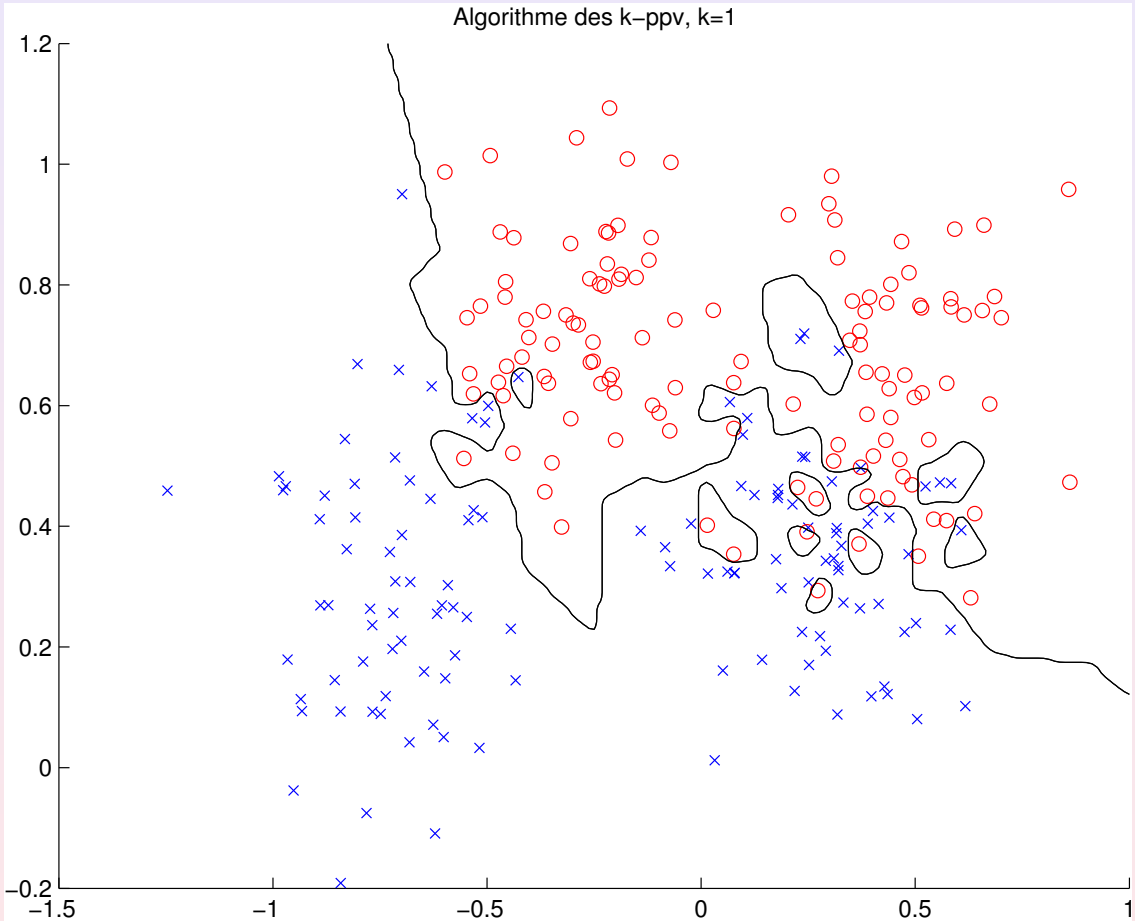


Figure: A first decision function

Discriminate between swans and geese

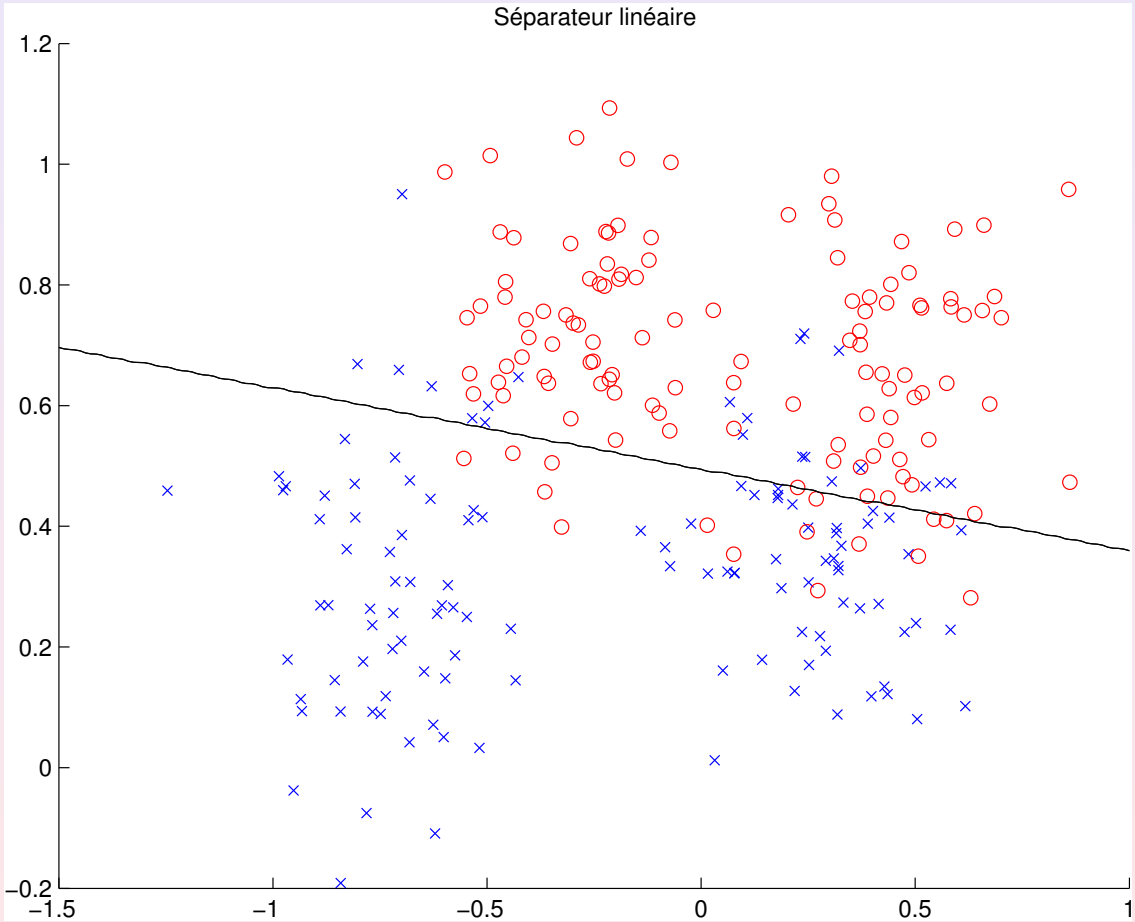


Figure: Another decision function

Discriminate between swans and geese

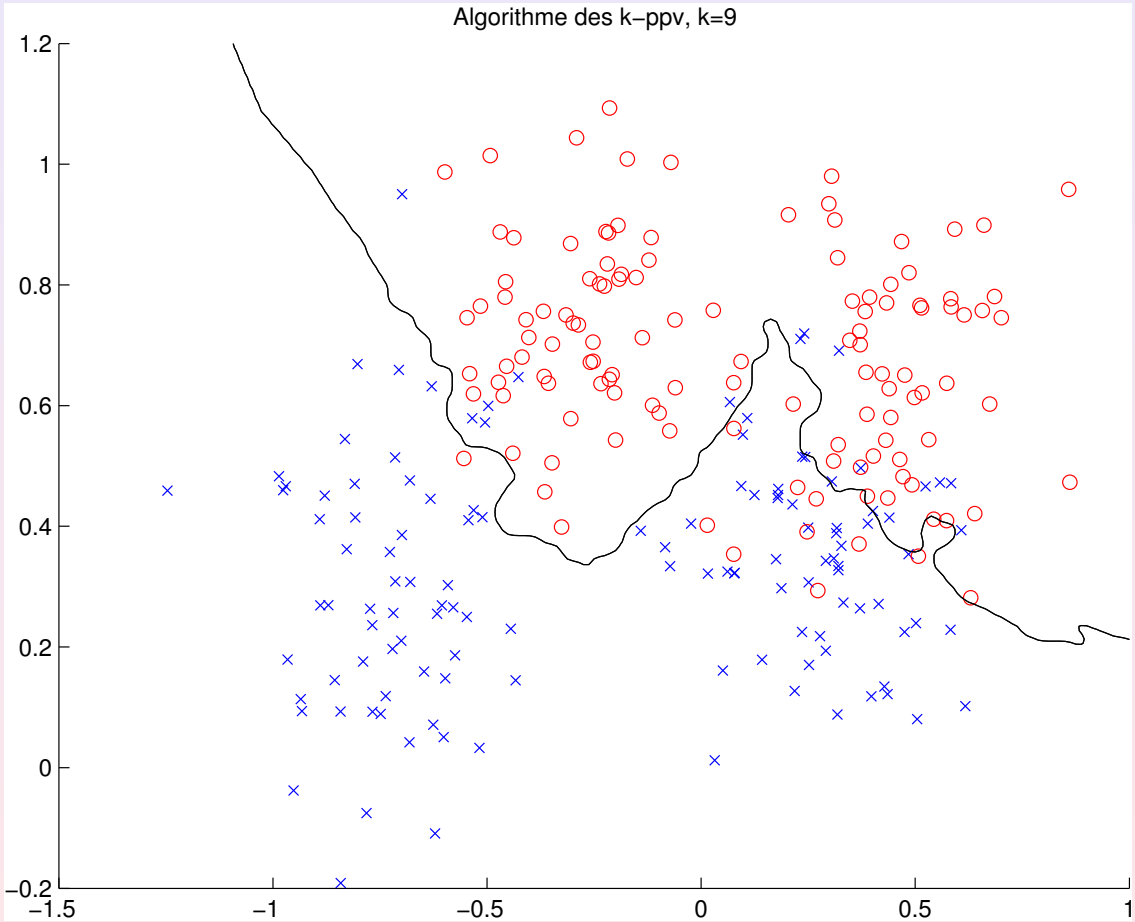


Figure: A last decision function

Supervised Learning framework

Available data are used to predict

- $\mathcal{S}$ is a *training set* $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in X \times Y$
- X is the *input space*, Y is the *output space*
- Usually [a], $X = \mathbb{R}^d$.
- When $|Y|$ is small, the task is called *classification* (if $|Y| = 2$, it is binary classification, $Y = \{-1, +1\}$)

a. Unless you are using kernels...

Machine Learning and NLP

Question

Can machine-learning techniques can readily be adapted to NLP tasks?

Answer

No, because linguistic representations are usually structured: both X and Y are complex. (It does not fit into a classification scheme.)

Similarity exploitation - The k-nn algorithm

- Main idea: similar inputs have similar outputs
- $\Rightarrow$ to classify a new input, look for neighbors which are already classified, and make them vote.

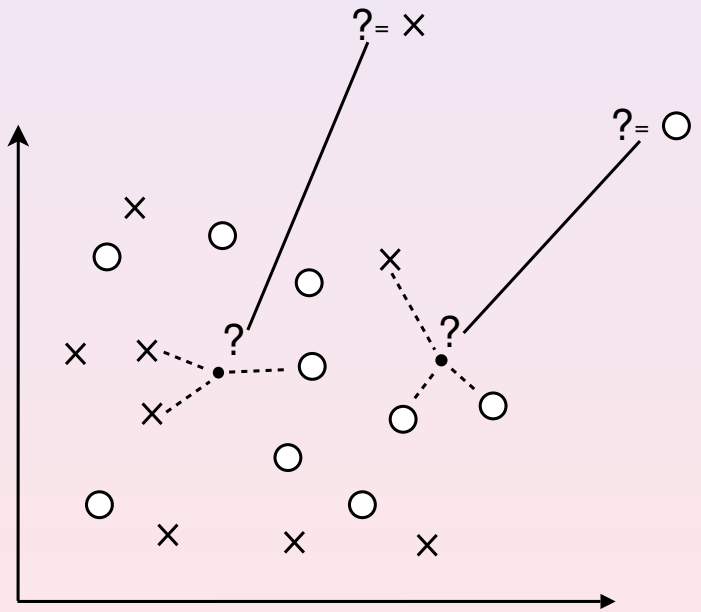


Figure: Principle of the k-nearest neighbors algorithm, $k = 3$

The k-nn algorithm

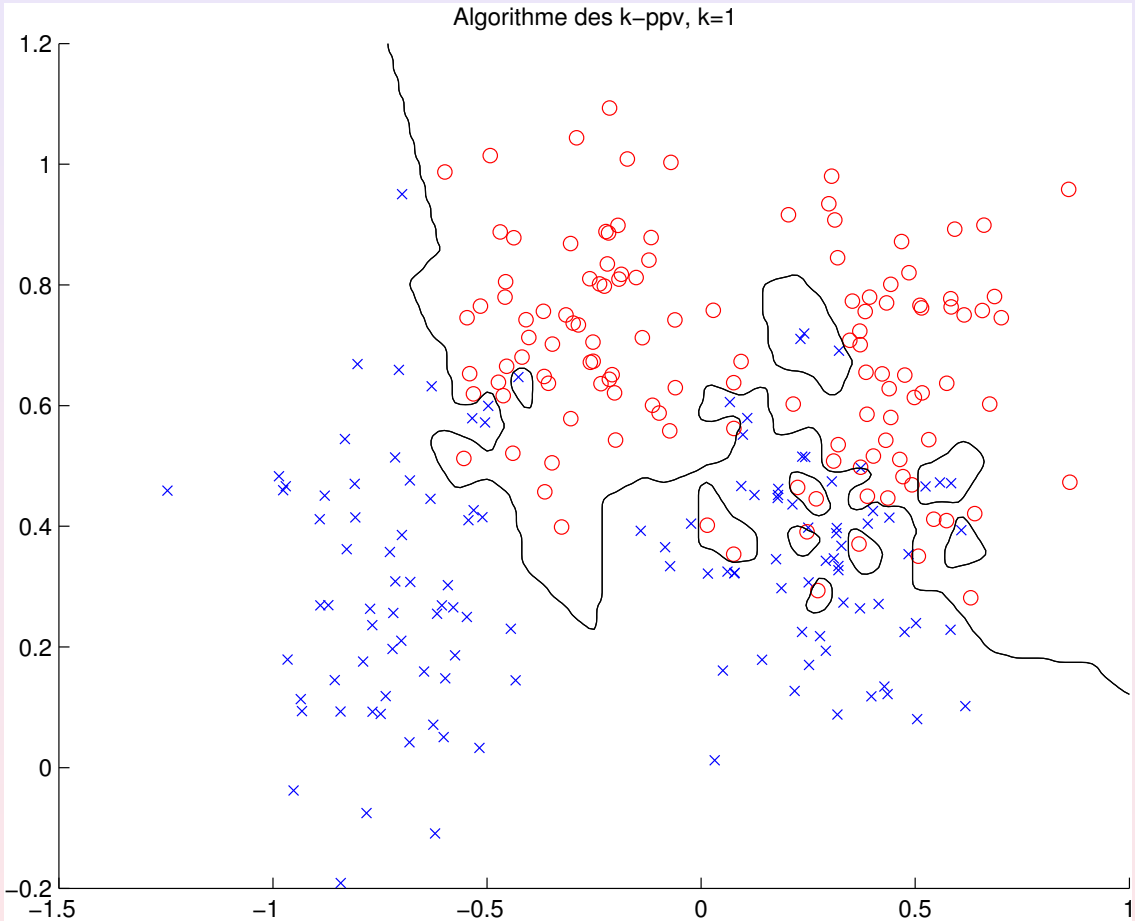


Figure: knn, $k = 1$

The k-nn algorithm

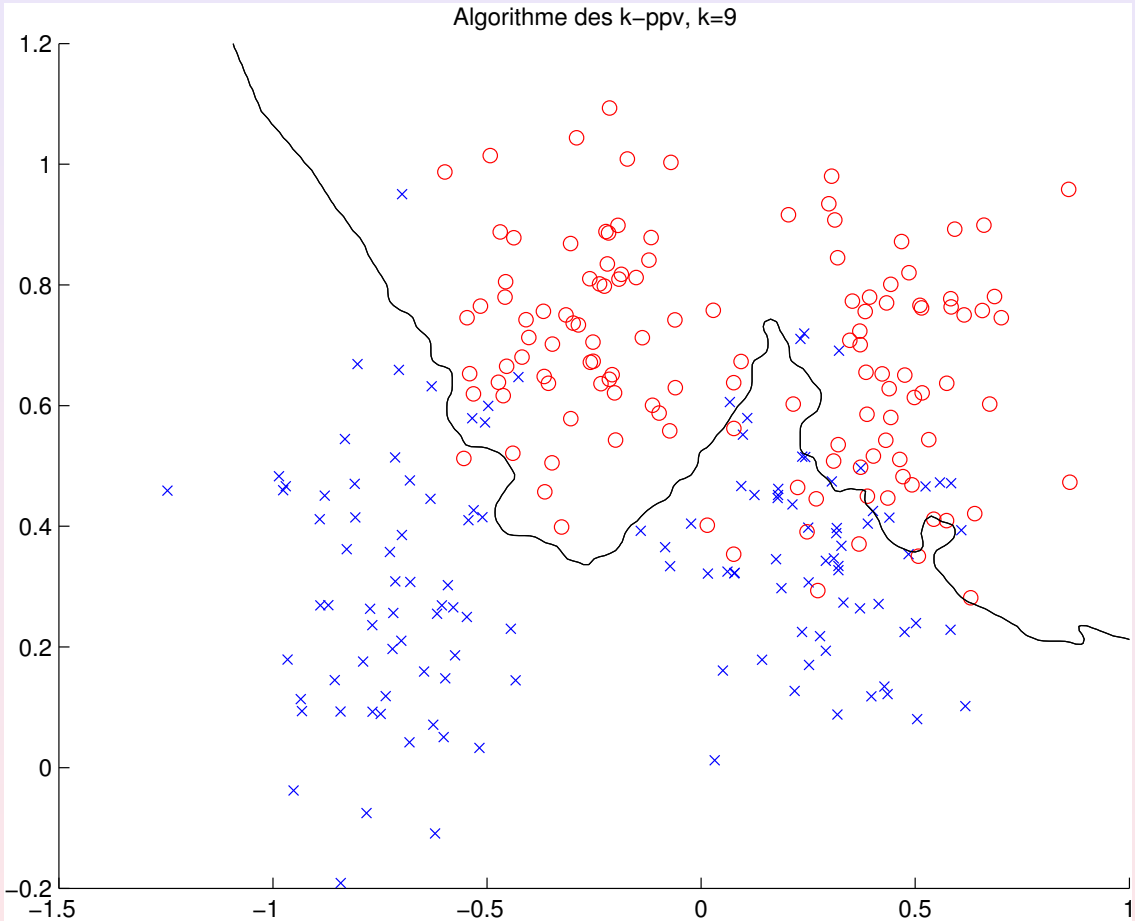


Figure: knn, $k = 9$

The k-nn algorithm

Main properties

- **Memory-based algorithm**. All the examples are stored in memory: no abstraction is performed.
- **Example-based algorithm**. Classification is done by comparing directly with known examples.
- **Lazy algorithm**. Since no abstraction is performed, all the processing is postponed until classifying a new example is required.
- Note: this is what the book of Daelemans and van den Bosch is about.

Large-margin classifiers

- Main idea: we should try to optimize the “space” (the margin) between examples of different classes.
- Consequence: only a small number of examples (the support vectors) are relevant.

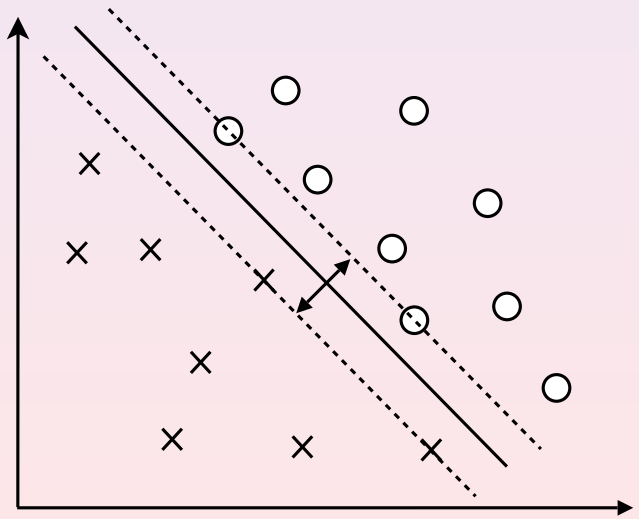


Figure: Support Vector Machines

Other common techniques

- Decision trees,
- Naive Bayes,
- Neural Networks,
- etc.

Supervised Learning framework

Available data are used to predict

- $\mathcal{S}$ is a *training set* $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in X \times Y$
- X is the *input space*, Y is the *output space*
- $X = \mathbb{R}^d$, $Y = \{-1, +1\}$

Assumption

The pairs $(x_i, y_i) \in X \times Y$ are independently identically distributed (iid) according to some unknown distribution P.

Supervised Learning framework

Available data are used to predict

- $\mathcal{S}$ is a *training set* $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in X \times Y$
- X is the *input space*, Y is the *output space*
- $X = \mathbb{R}^d$, $Y = \{-1, +1\}$

Goal

Construct a function g which correctly predict outputs from new inputs ($g : X \rightarrow Y$), i.e. with a low *risk*:

$$R(g) = P(g(X) \neq Y) = \mathbb{E}[1_{[g(X) \neq Y]}]$$

Supervised Learning framework

Available data are used to predict

- $\mathcal{S}$ is a *training set* $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in X \times Y$
- X is the *input space*, Y is the *output space*
- $X = \mathbb{R}^d$, $Y = \{-1, +1\}$

Problem 1

P is unknown: the real risk cannot be measured.

Solution 1

Empirical risk minimization (ERM) (error on data):

$$R_n(g) = \frac{1}{n} \sum_{i=1}^{n} 1_{[g(x_i) \neq y_i]}$$

Supervised Learning framework

Available data are used to predict

- $\mathcal{S}$ is a *training set* $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in X \times Y$
- X is the *input space*, Y is the *output space*
- $X = \mathbb{R}^d$, $Y = \{-1, +1\}$

Problem 2

Minimizing the empirical risk is not enough (overfitting)

Solution 2

Regularization:

$$RR_n(g) = \frac{1}{n} \sum_{i=1}^{n} 1_{[g(x_i) \neq y_i]} + \lambda L(g)$$

Interpretations of regularized risk

Regularized risk

$$RR_n(g) = \frac{1}{n}\sum_{i=1}^{n} 1_{[g(x_i)\neq y_i]} + \lambda L(g)$$

Bayesian intrepretation

Maximum a posterori (MAP)

$$argmax_g P(g|S) = argmax_g P(S|g)P(G)$$

$$argmax_g P(S|g)P(G) = argmax_g \Pi_{i=1}^{n} P((x_i, y_i)|g)P(g)$$

$$argmax_g logP(g|S) = argmax_g \sum_{i=1}^{n} logP((x_i, y_i)|g) + logP(g)$$

Interpretations of regularized risk

Regularized risk

$$RR_n(g) = \frac{1}{n}\sum_{i=1}^{n} 1_{[g(x_i)\neq y_i]} + \lambda L(g)$$

Minimum Description Length (MDL)

We are looking for the hypothesis which allow to code the data the most efficiently

$$g* = argmin_g L(S|g) + L(g)$$

Learning bias = Inductive bias = Prior

Fundamental problem of inductive learning

There is an infinite number of functions which agree with the data.

Example in grammatical induction

$$S = \{ab, aabb, aaabbb, aaaabbbb, aaaaabbbbb\}$$

Is the target language $\{(a^n b^n)_{n \in \mathbb{N}_+}\}$, $\{(a^n b^n)_{n \in \mathbb{N}_+}\} \cup \{bba\}$, or $\{(a^n b^n)_{n \in \{1,\ldots,31\}}\}$?

The need for learning bias

- **If there is no assumption on how the past is related to the future, prediction is impossible.**
- **If there is no restriction on the possible phenomena, generalization is impossible.** (Bousquet 2003)

Without these assumptions, nothing can be done.

See

- The Need for Biases in Learning Generalizations (Mitchell, 1980)
- No Free Lunch Theorems for Optimization (Wolpert, 1997)

Common assumptions/biases

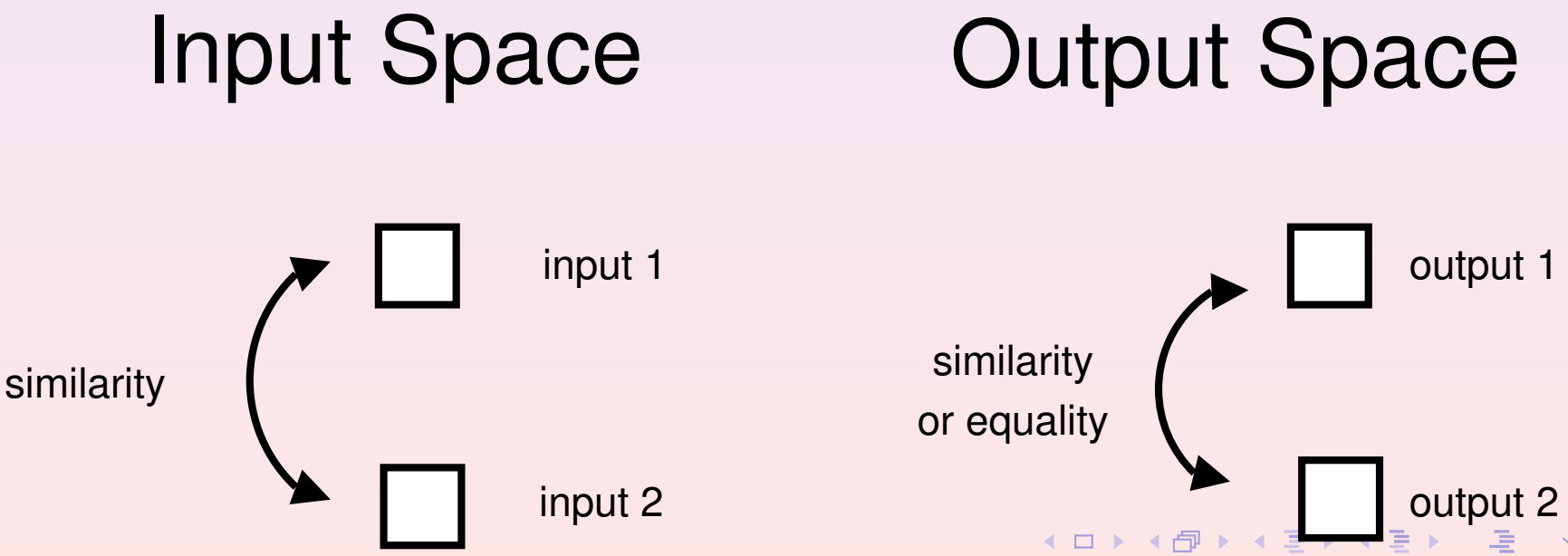
- Future data are similar to previous data (iid assumption)
- Similar inputs leads to similar outputs (regularity/continuity assumption)
- Occam's razzor: prefer "simple" functions (cf. regularization)

Take-home message

ML people will always need "specialists" to propose priors adapted to specific problems.

Remark

Lots of discriminative techniques rely on the exploitation of the conservation of similarity.
(Note: if the output space is finite, similarity amounts to equality).



Outline

Models of generation

Goal

To “explain” how objects have been produced.

$$P(o|i) \propto P(i|o)P(o)$$

The output "generates" the input.

Example: History-based models

Language modeling: compute $P(x_1 \dots x_n)$.

$$P(x_n|x_1 \dots x_{n-1}) = P(x_n|x_{n-k} \dots x_{n-1})$$

(Markovian assumption)

Alignement between inputs and outputs

Alignement

In order to express $P(i|o)$ when both i and o are complex, then objects are “decomposed” into smaller parts and inputs and outputs aligned.

Simple example: HMM

string to string correspondance. (ex: pronunciation)

```
live -> /liv-/
l -> /l/
i -> /i/
v -> /v/
e -> /-/
```

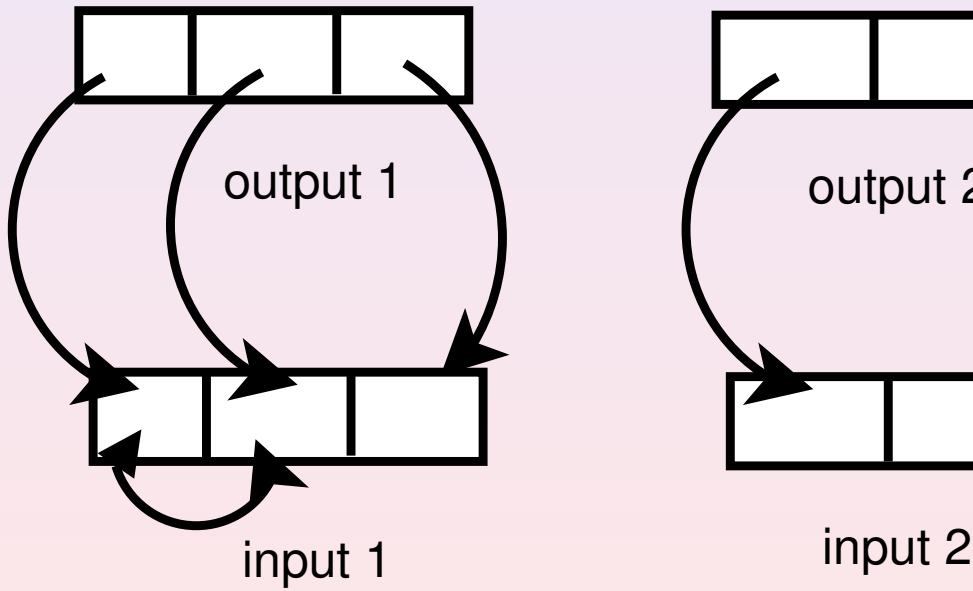
A more complex example: alignement for machine translation

The syntagmatic axe

History-based approaches try to exploit the **syntagmatic** (horizontal) organization of linguistic data.

Output Space

Input Space



Outline

Question

If history-based models exploit the **syntagmatic** (horizontal) axe, does machine-learning exploit the **paradimatic** axe?

Answer

- Not exactly.
- Explanation: similarity is too "simple".
- **But:** can be adapted.

“Extending” similarity

Question

What is the relationship between objects that is more complex than similarity and that is able to capture paradigmatic relationships between linguistic objects?

Possible answer

Analogical proportion: $x : y :: z : t$. (“x is to y what z is to t”)

The notion of analogical proportion in linguistics

Analogical proportion

X : *Y* :: *Z* : *T* ≡ *"X is to Y what Z is to T"*

- "*read* is to *unreadable* what *predict* is to *unpredictable*"
- **Paradigmatic** organisation of linguistic data
- ⇒ Good candidate for our purposes
- See Saussure, Bloomfield, Brugmann, . . .

Analogical proportions: example

reviewer,N : ?

Analogical proportions: example

search,V

reviewer,N : ?

Analogical proportions: example

search,V *view,V*

reviewer,N : ?

Analogical proportions: example

search,V	*view,V*
researcher,N	*reviewer,N* : ?

Analogical proportions: example

search,V	: s3J	*view,V*
researcher,N		*reviewer,N* : ?

Analogical proportions: example

search,V	: `s3J`	*view,V*	: `vju`
researcher,N		*reviewer,N*	: ?

Analogical proportions: example

search,V	: `s3J`	*view,V*	: `vju`
researcher,N	: `rIs3J@R`	*reviewer,N*	: ?

Analogical proportions: example

search,V	: `s3J`	*view,V*	: `vju`
researcher,N	: `rIs3J@R`	*reviewer,N*	: ?

The pronunciation of *reviewer* is `rIvju@R`.

Analogical proportions: example

search,V	: s3J	*view,V*	: vju
researcher,N	: rIs3J@R	*reviewer,N*	: rIvju@R

The pronunciation of *reviewer* is rIvju@R.

Analogical proportions. A MT example

un grand bateau : ?

Analogical proportions. A MT example

un *petit*
chien

: ?
un *grand*
bateau

Analogical proportions. A MT example

un petit chien

un petit bateau

: ?

un grand bateau

Analogical proportions. A MT example

un petit chien : *un petit bateau*

un grand chien : *un grand* ?

bateau

Analogical proportions. A MT example

un petit chien : a small dog :: *un petit bateau* : ?

un grand chien :: *un grand bateau*

Analogical proportions. A MT example

un	*petit*	: a small	*un*	*petit*	: a small
chien		dog	*bateau*		boat
un	*grand*		*un*	*grand*	: ?
chien			*bateau*		

Analogical proportions. A MT example

un petit chien	:	a small dog	*un petit bateau*	:	a small boat
un grand chien	:	a big dog	*un grand bateau*	:	?

Analogical proportions. A MT example

un	*petit*	: a small	*un*	*petit*	: a small
chien		dog	*bateau*		boat
un	*grand*	: a big	*un*	*grand*	: ?
chien		dog	*bateau*		

The translation of *un grand bateau* is a big boat.

Analogical proportions. A MT example

un	*petit*	: a small	*un*	*petit*	: a small
chien		dog	*bateau*		boat
un	*grand*	: a big	*un*	*grand*	: a big
chien		dog	*bateau*		boat

The translation of *un grand bateau* is a big boat.

Properties

The analogical learning scheme we presented is

- able to deal with structured objects (both in input and output spaces)
- does not need inter-level mappings (it only relies on relationships between objects in the same space)

Thanks for your attention.