Transductive Learning: Motivation, Model, Algorithms

Olivier Bousquet
Centre de Mathématiques Appliquées
Ecole Polytechnique, FRANCE
olivier.bousquet@m4x.org

University of New Mexico, January 2002

Goal

- Provide motivation/potential applications

- Sketch algorithmic issues

- Sketch theoretical problems

Overview

→ Induction vs Transduction

- Algorithms
- Formalization
- Open issues

Induction

We consider a phenomenon f that maps inputs (instances) $\boldsymbol{x}$ to outputs (labels) $y = f(\boldsymbol{x})$ (here $y \in \{-1, 1\}$)

- Given a set of example pairs (training set) $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$,
- the goal is to recover f

$\longrightarrow$ This will allow to predict the label y_{n+1} of a previously unseen instance $\boldsymbol{x}_{n+1}$.

Example: Face recognition
Train on pictures of a person and recognize him/her the next day

Shortcomings

But there are situations in which

- Obtaining labels is expensive
- Obtaining instances is cheap
- We know in advance the instances to be classified
- We do not care about the classification function

$\longrightarrow$ Transduction applies

Examples

Information retrieval

Information retrieval with relevance feedback

- User enters a query
- Machine returns sample documents
- User labels the documents (relevant/non-relevant)
- Machine selects most relevant documents from database

Relevance

- Obtaining labels requires work from the user
- Obtaining documents is automatic (from database)
- Instances to be classified: documents of the database
- No need to know the classification function (changes for each query)

Transduction

We consider a phenomenon f that maps inputs (instances) $\boldsymbol{x}$ to outputs (labels) $y = f(\boldsymbol{x})$ (here $y \in \{-1, 1\}$)

- Given a set of labeled examples $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$,
- and a set of unlabeled examples $x'_1, \ldots, x'_m$
- the goal is to find the labels $y'_1, \ldots, y'_m$

$\longrightarrow$ No need to construct a function f, the output of the transduction algorithm is a vector of labels.

$\longrightarrow$ Transfer the information from labeled examples to unlabeled.

Using Transduction for Prediction

Given training data and data to be classified, one can either

- Use induction: build $\hat{f}$ and classify the data with it
- Use transduction directly for classifying data

Even in an inductive setting, one can use transduction.

Example: News filtering

- First day user classifies news according to interest
- Subsequent days, machine classifies incoming news based on first day labels

$\longrightarrow$ Train on the fly, when receiving the data to be classified
Retrain the machine every day

$\longrightarrow$ Maximally use the information and tune the result to the news of the day

Three Learning Tasks

- Induction: $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\} \mapsto f$
- Induction with unlabeled data: $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\} \cup \{x'_1, \ldots, x'_m\} \mapsto f$
- Transduction: $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\} \cup \{x'_1, \ldots, x'_m\} \mapsto (y'_1, \ldots, y'_m)$.

The choice will depend on

- Availability of unlabeled data
- Need for interpretability
- Time considerations

Overview

- Induction vs Transduction

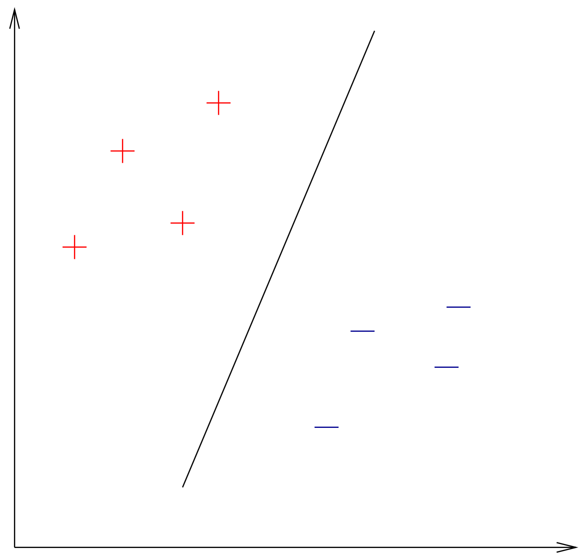
→ Algorithms

- Formalization

- Open issues

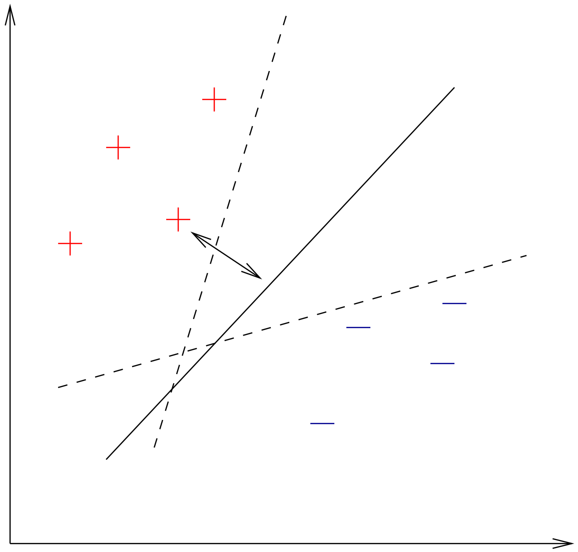
Linear classification

Instances represented in $\mathbb{R}^d$.
Find a linear separation.



Large margin classification

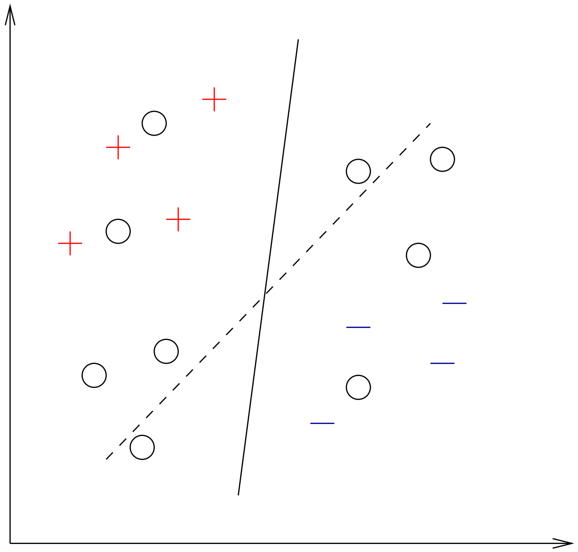
Margin = distance from the hyperplane to the closest point



Maximize the margin → leads to 'robust' solution
→ Support Vector Machines

Transduction

- Assumption: separated classes
- Maximize the margin on unlabeled instances.



Implementation

Goal: Maximize the margin on all examples

Algorithmic issues

- no unlabeled data $\longrightarrow$ quadratic optimization (n^3)
- unlabeled data $\longrightarrow$ combinatorial problem (NP)

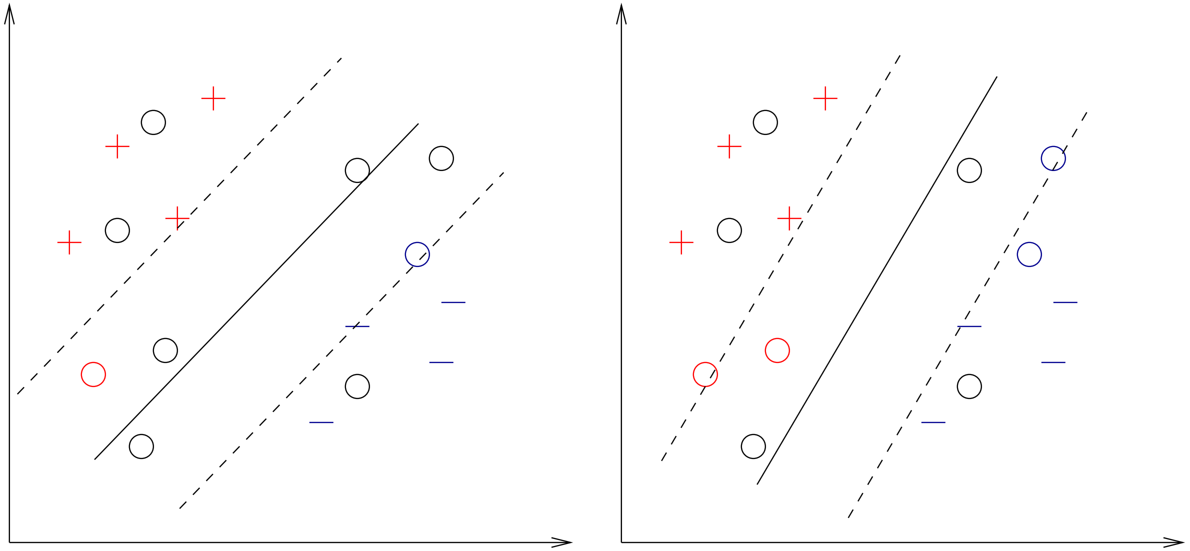
$\longrightarrow$ Need heuristics

$\longrightarrow$ Greedy optimization

Algorithms

Greedy

- Only the examples in the margin have an influence
- Label the ones with largest confidence (largest margin)



$\longrightarrow$ May add backtracking

Comments

- Influenced by starting point (induction)
- Not fully transductive because builds an $\hat{f}$
- Assumption that data is separated

$\longrightarrow$ Can we make the data separated ?

Kernel Machines

Support Vector Machines

- Map data into a feature space

$$\boldsymbol{x} \in \mathcal{X} \to \Phi(\boldsymbol{x}) \in \mathcal{F}$$

- Perform maximal margin classification in feature space

Kernel trick

- Algorithm can be implemented by computing inner products

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

- Simply choose a kernel and run the linear algorithm on the matrix

$$K = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j \in \{1,\dots,n\}}$$

$\to$ k is a measure of similarity. Algorithm works on similarity matrix.

Alignment

- Choice of Kernel = choice of feature space
- Ideal kernel = feature space contains label
- Ideal kernel matrix

$$k_I(\boldsymbol{x}_i, \boldsymbol{x}_j) = y_i y_j$$

Measure distance from ideal kernel: Alignment

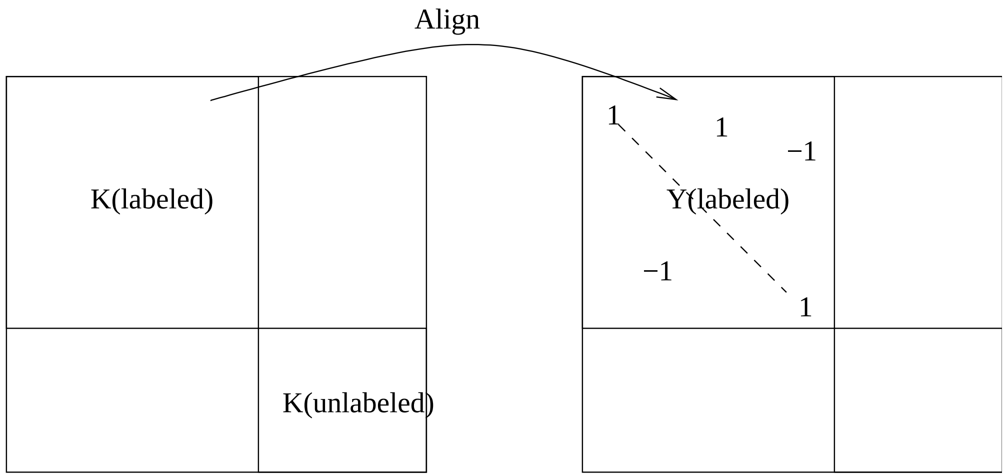
$$A(K) = \sum_{i,j} K_{ij} y_i y_j$$

Measures the data separation:

$$A(K) = \sum_{y_i = y_i} k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{y_i \neq y_j} k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Transduction as Optimization

- Maximize alignment on the labeled data
- Corresponds to maximizing data separation
- Diagonalize, fix eigenvectors, optimize eigenvalues



Overview

- Induction vs Transduction
- Algorithms

$\longrightarrow$ Formalization

- Open issues

Formalization

- Data is fixed

$$x_1, \dots, x_{n+m} \in \mathcal{X}$$

$$y_1, \dots, y_{n+m} \in \{-1, 1\}$$

- Oracle (teacher) chooses randomly a subset

$$I \subset \{1, \dots, n+m\}$$

- Input to algorithm

$$x_1, \dots, x_{n+m}$$

$$I$$

$$(y_i)_{i \in I}$$

- Output of algorithm

$$(\hat{y}_i)_{i \in \{1,\dots,n+m\}}$$

Formalization

Random choice of I

Randomness models

- Fixed size
  Choose n examples among $n+m$ with uniform probability for every choice, $\binom{n+m}{n}^{-1}$. $|I| = n$.
- Variable size
  For each $i \in \{1, \dots, n+m\}$ choose independently with probability $\frac{n}{n+m}$ to include it.
  $\longrightarrow \mathbb{E}\left[|I|\right] = n$.

$\longrightarrow$ We want to make statements that hold with high probability over the random choice of I.

Formalization

Risk

Recall output $\hat{\boldsymbol{y}} = \hat{y}_1, \ldots, \hat{y}_{n+m}$.
$\hat{\boldsymbol{y}}$ is an $n + m$ dimensional vector in $\{-1, 1\}^{n+m}$.

- Test error

$$R(\bar{I}, \boldsymbol{y}) = \frac{1}{|\bar{I}|} \sum_{i \in \bar{I}} \mathbb{I}\{\hat{y}_i \neq y_i\}$$

- Cannot be computed: need to estimate it from the data

Error bounds

We estimate the test error by the empirical error

$$R(I, \hat{\boldsymbol{y}})$$

We want to prove

$$\mathbb{P}_I \left[R(\bar{I}, \hat{\boldsymbol{y}}) - R(I, \hat{\boldsymbol{y}}) > \epsilon \right] \leq \delta$$

Choose a set of vectors $\mathcal{Y} \subset \{-1, 1\}^{n+m}$. We want to bound

$$\mathbb{P}_I \left[\sup_{\boldsymbol{y} \in \mathcal{Y}} R(\bar{I}, \boldsymbol{y}) - R(I, \boldsymbol{y}) > \epsilon \right]$$

Results

When $n = m$,

$$R(\bar{I}, \boldsymbol{y}) \leq R(I, \boldsymbol{y}) + KC(\mathcal{Y}) + O\left(\frac{1}{\sqrt{n}}\right)$$

Where C Rademacher complexity of $\mathcal{Y}$.

When $m > n$,

$$R(\bar{I}, \boldsymbol{y}) \leq R(I, \boldsymbol{y}) + K\bar{C}(\mathcal{Y}_{2n}) + O\left(\frac{1}{\sqrt{n}}\right)$$

where $\bar{C}(\mathcal{Y}_{2n})$ is the average Rademacher complexity computed on subsets of size $2n$ of the data.

$\longrightarrow$ Complexity can be computed from x_i only. Labels don't play any role !

Overview

- Induction vs Transduction

- Algorithms

- Formalization

$\longrightarrow$ Open issues

Model Selection

Induction

- Define a structure without any data
- Compute empirical complexity

Transduction

- Define a structure with all the x_i
- Know exact complexity of this structure

$\longrightarrow$ Data-dependent classes.

$\longrightarrow$ Justifies the margin approach.

Open Problems

- Analyze alignement algorithm in that framework
- Provide model selection methods
- Provide Rademacher estimates
- Prove that unlabeled data really help

Conclusion

- Different framework with potentially interesting applications
- Very few people studied it: a lot remains to be done
- Challenges
  - Good empirical evidence $\longrightarrow$ justification ?
  - Algorithmic $\longrightarrow$ make transduction efficient
  - Theoretical $\longrightarrow$ provide guarantees