Learning from Labeled and Unlabelled Data: When the Smoothness Assumption Holds

Michelangelo Ceci

http://www.di.uniba.it/~ceci/index.htm

ceci@di.uniba.it
Transductive/semi-supervised vs Inductive Learning in predictive tasks

**Inductive learning**: Only labeled examples used for generating a classifier
Discards large amount of information potentially conveyed by unlabeled instances

**Transductive/semi-supervised learning**: Uses both labeled (training set) & unlabeled (working set) data to build classifier.
Learning from Labeled and Unlabelled Data: When the Smoothness Assumption Holds

Labeled data only

Transductive SVM
Motivation for Transductive/Semi-Supervised Learning

Philosophical:
  • Human brain can exploit unlabeled data.

Pragmatic:
  • Unlabeled data is usually cheap to collect.
Motivation for Transductive/Semi-Supervised Learning

Labeled training data is scarce and expensive.
- Eg., experiments in computational biology.
- Need for expert knowledge.
- Tedious and time consuming.

Unclassified instances are abundant and cheap.
- Extract vectorized maps from satellite images.
- Assess primary structure of proteins from DNA/RNA.
Transductive vs Semi-supervised learning: differences

Different assumptions on the test data.

Transductive learning takes a “closed-world” assumption, i.e., the test data set is known in advance and the goal of learning is to optimize the generalization ability on this test data set, while the unlabeled examples are exactly the test examples.

Semi-supervised learning takes an “open-world” assumption, i.e., the test data set is not known and the unlabeled examples are not necessary test examples.
Transductive vs Semi-supervised learning: differences

The difference is also clear in the experimental protocol:

- **L** number of labelled cases
- **U** number of unlabelled cases
- **N** number of examples (possibly not available during learning)

**Transductive setting** $N = U + L$: the training set comprises of **N** examples, **L** of which are labeled. Performance evaluated in predicting **N-L** unlabeled examples.

**Semi-supervised setting** $N \neq U + L$: the training set comprises of **L+U** examples. Performance evaluated in predicting **N-L-U** unlabeled examples.
Transductive vs Semi-supervised learning: differences

This means that transductive learning is an inference mechanism "from particular to particular" whose goal is to classify (only) unlabeled data as accurately as possible (no general rule valid for all possible instances is generated).

Making predictions on test examples at hand is easier than seeking the best hypothesis for all potential examples.

We should not try to solve a problem by solving a more difficult intermediate problem.

It always possible to learn a classifier after transductive learning.
Transductive learning vs. inductive learning vs. semi-supervised learning

A transductive algorithm is one whose solution depends on the test points that are given.

An inductive algorithm is the very opposite: the solution depends exclusively on the labelled data.

A semi-supervised algorithm also depends on the unlabelled set.
Transductive learning: early references

This name was used for the first time by Vapnik (Vapnik & Chervonenkis, 1974; Vapnik & Sterin, 1977).

An early instance of transduction (albeit without explicitly considering it as a concept) was already proposed by Hartley and Rao (1968), who suggested a combinatorial optimization on the labels of the test points in order to maximize the likelihood of their model.

Interest for transductive learning increased in the 1990s, mostly due to applications in text classification.
Smoothness assumption in supervised learning

If two points $x_1$ and $x_2$ are close, then so should be the corresponding outputs $y_1$, $y_2$.

Without such assumption, it would never be possible to generalize from a finite training set to a set of possibly infinitely many unseen test cases.

The application of this assumption is evident in similarity-based learning:

Training instances are stored in memory and a similarity metric is used to compare new instances to those stored. New instances are classified according to the close exemplars in memory.
Semi-supervised smoothness assumption

If two points $x_1$ and $x_2$ in a high-density region are close, then so should be the corresponding outputs $y_1$, $y_2$.

The label function is smoother in high-density regions than in low-density regions.

This assumption entails that if two points are separated by a low-density region, then their outputs need not be close.

It is also called label smoothness assumption.
Semi-supervised smoothness assumption

Main difference smoothness assumption in supervised learning: closeness between points is not a decisive factor, if, considered by itself. It has to be considered in the context of the underlying distribution.

(a) The unknown point, denoted by “?”, is classified in the same class as point “*”. (b) The setup after a number of unlabeled data have been provided, which leads us to reconsider our previous classification decision.
Semi-supervised smoothness assumption

Main difference smoothness assumption in supervised learning: closeness between points is not a decisive factor, if, considered by itself. It has to be considered in the context of the underlying distribution.

(a) The unknown point, denoted by “?”, is classified in the same class as point “*”. (b) The setup after a number of unlabeled data have been provided, which leads us to reconsider our previous classification decision.
Cluster assumption

If points are in the same cluster, they are likely to be of the same class.

Idea: run a clustering algorithm and use the labeled points to assign a class to each cluster. This is in fact one of the earliest forms of semi-supervised learning.

The cluster assumption can be seen as a special case of the semi-supervised smoothness assumption, when clusters are defined by considering only high-density regions.

The cluster assumption can be formulated in an equivalent way:

Low density separation: the decision boundary should lie in a low-density region.
Transductive learning: which assumptions

These assumptions underlay semi-supervised learning but it is argued that transduction relies on the same kind of assumptions as semi-supervised learning.

(Chapelle 2006)
Transductive learning vs. local learning

Local learning: given a test point, one should focus on the training points which are in a neighborhood of this test point, construct a local decision rule, and predict the label of the test point according to this ad hoc rule.

Similar to transduction!

However, local learning is still inductive because there exists an implicit decision function, even though it is never explicitly constructed.

The concept of local learning is actually almost the same as transduction with one test point.
Autocorrelation

Given a random variable $Y$ representing the output of some observations $x_i$, and a distance function defined on observations, autocorrelation is the correlation among output values $y_i$ strictly attributable to the proximity of observations according to the distance function.

Autocorrelation introduces a deviation from the independent observations assumption of classical statistics.

Positive (negative) autocorrelation is the tendency for similar (dissimilar) values to cluster.

Positive autocorrelation is more common than negative autocorrelation in spatial and social phenomena.
Positive Autocorrelation

Infant (under-five) mortality rate in Nigeria. Local Moran's I show statistically significance spatial autocorrelation (Moran's I = 0.654, p=.001).
Positive autocorrelation and the smoothness assumption

Positive autocorrelation entails the smoothness assumption. Moreover, when the similarity between two observations is defined so that two observations are never considered similar when they are separated by low-density regions, then positive autocorrelation entails the semi-supervised smoothness assumption which underlies transductive learning as well.
Positive autocorrelation and the smoothness assumption

Autocorrelation is valid in networked data, in Spatial Data (spatial autocorrelation) but also in many other fields such as

- sociology (e.g., social relations affect social influence),
- web mining (e.g., related pages on the same topic)
- bioinformatics (e.g., proteins located in the same place in a cell are more likely to share the same function than randomly selected proteins).

In these fields, “distance” should reflect the properties of interest.
Spatial Data

Spatial Autocorrelation: The property of random features taking values, at pairs of locations a certain distance apart, that are more similar than expected for randomly associated pairs of observations.
Networked Data

**Nodes** represent entities

**Links** represent existing relations between entities

Nodes with known labels are interlinked with nodes for which labels have to be estimated

Labels are sparse

Examples:
- Internet
- Social networks
- Sensor networks ...
Opportunities for Transductive learning

- HSGT: Hierarchical classification of textual documents
- SPRECO (SPatial REgression with CO-training)
- TRANSC (TRANsductive Structural Classifier)
HSGT: Hierarchical classification of textual documents
HSGT in WebClass: the problem

Hierarchical classification of textual documents in the transductive setting

Given:

– A set $Y=\{C_1, C_2, \ldots, C_L\}$ where each $C_i$ represents a category;

– A hierarchical function $H: Y \rightarrow Y$ that associates each category (except the root) with its parent category;

– A training set $TS$ of pairs $\{(d_i, y_i)\}$ where $d_i$ represents a document and $y_i \in Y$ represents the class;

– A working set $WS$ of unlabeled documents.

Find:

a prediction of the class value of each document in the working set $WS$ which is as accurate as possible.
HSGT in WebClass: basic idea

When a document $d$ reaches an internal node $C$, it is represented according to a feature set associated to $C$.

The system decides to which direct subcategory $C'$ of $C$ the document has to be passed.

Decision is taken on the basis of a $(l$-of-$m)$ classifier associated to each internal node that returns a score $\gamma_{C \rightarrow C'}(d)$ associated to the decision.

Documents can be classified in internal nodes when scores do not exceed a threshold value.

Thresholds are automatically determined by means of a bottom-up search strategy that minimizes the error of the classification.

(Ceci & Malerba, JIIS 2007)
HSGT in WebClass: the problem

HSGT: hierarchical version of the original SGT classifier (Joachims, 2003)

Given

An internal category $C$ and a direct subcategory $C'$ of $C$;

A set of labeled documents belonging to $C$ and its descendants. Positive examples (+) refer to documents in $C'$ and all its descendants, while negative examples (-) refer to documents in $C'$ and in all other categories descendants of $C$;

A set of unlabeled examples (possibly) belonging to $C$ and its descendants;

Compute the score $\gamma_{C \rightarrow C'}(d)$ for each document $d \in TS \cup WS$ such that error is minimized.
Document representation

- Tag, punctuation marks, numbers, tokens shorter than three characters and stopwords are removed,
- Stemming
- Features are sorted according to:

\[ v_i = TF_{c'}(w_i) \times DF_{c'}^2(w_i) \times \frac{1}{CF_c(w_i)} \]

where \( c' \) is direct subcategory of \( c \). Then, the best \( ndict \) are selected.
- Union of dictionaries extracted from documents of categories \( C' \) forms the category feature set \( \text{FeatSet}_C \).
- Documents are represented according to the classical TF x idf measure (Sebastiani, 2002). The used dissimilarity measure is the cosine correlation.
HSGT in WebClass: the problem

SGT builds a nearest neighbor graph $G=(N,E)$ with labeled and unlabeled documents as vertexes, dissimilarity measure $d(d_i,d_j)$ between the neighboring documents as edge weights.

SGT assigns labels to unlabeled examples by cutting $G$ into two subgraphs $G^-$ and $G^+$ such that a normalized cut cost is maximized

$$\max_y \frac{\text{cut}(G^+, G^-)}{|\{i|y_i = +1\}| \cdot |\{i|y_i = -1\}|}$$
HSGT in WebClass: the problem

Two maximization procedures are considered:

- RATIO CUT
- NORMALIZED CUT

The score $\gamma_{c \rightarrow c'}(d_i)$ is computed as:

$$
\gamma_{c \rightarrow c'}(d_i) = \frac{(z_i^* - \min_j z_j^*)}{(\max_j z_j^* - \min_j z_j^*)}
$$

where $Z^* = [z_i^*]_{i=1,...,n}$ is the transformed predictor vector with comparable scores (used during maximization step).
Relevant Example Selection

• Examples to be sampled:
  – Training Set and Working Set

Two techniques:
  – Clustering based (taking the centroid - \textit{KmSGT} or a representative example - \textit{SelectSGT})
  – Class border identification (taking support vectors - \textit{SVSGT})
Experiments

• RCV1: 800,000 newswire stories organized in a set of 104 categories distributed on 4-levels. **Lewis et al.’s split**
  – a subset rooted in the category “C3” (1,647 training documents, 50,345 working documents);
  – a subset rooted in the category “C18” (1,438 training documents, 44,148 working documents);
  – a subset rooted in the category “MCAT” (10,715 training documents, 163,592 working documents).

• Dmoz data is obtained by ODP: 3,668 documents in 203 categories. **3-fold cross-validation.**
  – dataset rooted in the category “Cancer”;
  – a subset rooted in the category “Cardiovascular disorders”.
Some Experimental Results

<table>
<thead>
<tr>
<th>DATASET</th>
<th>cut</th>
<th>HSGT</th>
<th>kmSGT</th>
<th>SelectSGT</th>
<th>SVSGT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCER</td>
<td>RATIO</td>
<td>64%</td>
<td>65%</td>
<td>59%</td>
<td>32%</td>
</tr>
<tr>
<td></td>
<td>NORMALIZED</td>
<td>60%</td>
<td>60%</td>
<td>54%</td>
<td>28%</td>
</tr>
<tr>
<td>CARDIOVASCULAR</td>
<td>RATIO</td>
<td>63%</td>
<td>64%</td>
<td>57%</td>
<td>33%</td>
</tr>
<tr>
<td></td>
<td>NORMALIZED</td>
<td>60%</td>
<td>60%</td>
<td>51%</td>
<td>34%</td>
</tr>
<tr>
<td>CONDITION</td>
<td>RATIO</td>
<td>37%</td>
<td>39%</td>
<td>36%</td>
<td>13%</td>
</tr>
<tr>
<td></td>
<td>NORMALIZED</td>
<td>34%</td>
<td>37%</td>
<td>32%</td>
<td>14%</td>
</tr>
<tr>
<td>C3</td>
<td>RATIO</td>
<td>–</td>
<td>31%</td>
<td>28%</td>
<td>18%</td>
</tr>
<tr>
<td></td>
<td>NORMALIZED</td>
<td>–</td>
<td>31%</td>
<td>12%</td>
<td>29%</td>
</tr>
<tr>
<td>C18</td>
<td>RATIO</td>
<td>–</td>
<td>69%</td>
<td>77%</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>RATIO</td>
<td>–</td>
<td>49%</td>
<td>50%</td>
<td>5%</td>
</tr>
<tr>
<td>MCAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SPRECO (SPatial REgression with CO-training)
Transductive Spatial Regression

Let:

• $D = T \cup W$ be a the set of spatial observations in the form $(id, u, v, x, y)$

• $(u, v)$ : a position on $\mathbb{R}^2$ (field based model);

• $x$ : the vector of values measured for explanatory attributes $X_1, \ldots, X_n$ at $(u, v)$;

• $y$ : a possibly unknown (in $W$) response with range in $\mathbb{R}$

• $d$ be the spatial distance

\[ d : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \]

used to determine the spatial neighborhood of a position $(u, v)$
Transductive Spatial Regression

The goal is to predict unknown $y$ of each observation of $W$ as accurate as possible
The solution

**SPatial REgression with CO-training**

Learning in co-training style (Blum and Mitchell, 1998) by

- Deriving **two separate views** of observations in D
- Iteratively
  - learning two regression models, $t_1$ and $t_2$, from the separate views of labeled data, and
  - using $t_1$ to add observations to the labeled set that $t_2$ will be able to use for learning, and vice-versa
- Using an **ensemble** of $t_1$ and $t_2$ to predict unlabeled observation in D
The solution

SPatial REgression with CO-training

Labeled Set

Unabeled Set

Extract alternate views

The basic inductive learner is the model tree learner proposed in (Appice & Džeroski, 2007)
Two Views of Spatial Data

Two feature spaces to describe the same set of observations

\[ D = T \cup W \text{ that is the data distribution over } X_1, \ldots X_n \]

\[ \overline{D} = \overline{T} \cup \overline{W} \text{ that is the data distribution over } \overline{X}_1, \ldots \overline{X}_n \]

\[ \overline{X}_{i_e} = \frac{\sum_{p \in \Omega_D(e)} x_{i_p} \times w(e, p)}{\sum_{p \in \Omega_D(e)} w(e, p)} \quad \text{with} \quad \begin{cases} |\Omega_D| = h \\ w(e, p) = e \\ b_e = \max_{p \in \Omega_D(e)} \text{distance}(e, p) \\ \frac{\text{distance}(e, p)^2}{b_e^2} \end{cases} \]
Determine reliability of labels

Let:

• $e$ be an unlabeled observation in $W$
• $\hat{y}_e$ the label predicted for $e$
• $N_k(e,T)$ be the set of $k$ nearest neighbors of $e$ in the labeled set $T$

Reliability of $\hat{y}_e$ is evaluated based on a K-NN re-prediction of labeled data falling in $N_k(e,T)$
Determine reliability of labels

\[
\text{for } p \in N_k(e, T) \text{ do }
\]
\[
p = (y_p - k-\text{NN}(p, T))^2 - (y_p - k-\text{NN}(p, T \cup \{e, \widehat{y}_e\}))^2;
\]
\[
\text{if } p \geq 0 \text{ then } \text{Pos} = \text{Pos} + 1;
\]
\[
\text{else } \text{Neg} = \text{Neg} + 1;
\]
end for

\(\widehat{y}_e\) is reliable iff \(\text{pos} \geq \text{neg}\)
Labeling

- Use the regression models, \( t_1 \) and \( t_2 \), learned in the last iteration of SPRECO, to predict unlabeled data.
- For each unlabeled observation \( e \in W \),
  - two labels are predicted by using \( t_1 \) and \( t_2 \), respectively
  - labels are averaged

\[
\text{Avg}.y_e = \frac{\text{response}(t_0, e) \omega_0 + \text{response}(t_1, \bar{e}) \omega_1}{\omega_0 + \omega_1}
\]
Experiments

Databases

• USA Geographical Analysis Spatial Data (GASD) - 3,107 observations
• Forest Fires - 512 observations
• North-West England (NWE)- 979 observations
• Sigmea-Real MF - 817 observations
• Sigmea-Real MS - 817 observations

For each trial a single fold forms the training set and the hold-out \( K - 1 \) folds form the working set.
GASD

Forest Fires

NWE

Sigmea-Real MS

Sigmea-Real MF

Learning from Labeled and Unlabelled Data: When the Smoothness Assumption Holds
TRANSC (TRANsductive Structural Classifier)
TRANSC (TRANsductive Structural Classifier)

Based on the Transductive Learning framework
Exploits expressive power of Relational Learning to deal with relational data in their original form
Classification is probabilistic
Extends the system Mr-SBC (Ceci et al. PKDD 2003) in order to work in the transductive learning setting
TRANSC: problem definition

Given:
- A database schema $S$ which consists of a set of $h$ relational tables $\{T_0, ..., T_{h-1}\}$, a set $PK$ of primary key constraints on the tables in $S$, and a set $FK$ of foreign key constraints on the tables in $S$
- A target relation $T \in S$ and a target discrete attribute $y$ in $T$ (class), different from the primary key of $T$, whose domain is the finite set $\{C_1, C_2, \ldots, C_L\}$
- A training set that is an instance $TS$ of the database schema $S$ with known values for $y$
- A working set that is an instance $WS$ of the database schema $S$ with unknown values for $y$

Find the most accurate prediction of $y$ for examples in $WS$ represented as a tuple of $t \in WS.T$ and all tuples related to $t$ in $WS$ according to $FK$. 
TRANSC: algorithm

Algorithm : Top level transductive algorithm description

1: transductiveClassifier(initialClassification, TS, WS)
2: classification1 ← initialClassification; changedExamples ← Ø; i ← 0;
3: repeat
   4: prevClassification ← classification1;
   5: prevChangedExamples ← changedExamples;
   6: classification2 ← reclassifyExamplesKNN(classification1, TS, WS);
   7: (classification1, changedExamples) ← changeClass(classification2);
8: until (computeOverlap(prevChangedExamples, changedExamples) ≥ MAXOVERLAP) OR (++i ≥ MAX_ITERS) )
9: return prevClassification
TRANSC

- **initialClassification** of $E \in (WS \cup TS)$:
  - IF $E \in TS$
    - preclass($E$) that returns class($E$)
  - Else
    - BayesianClassification($E$) is the initial inductive classifier built from the training set $TS$.

- The initial classifier is obtained by means of the relational probabilistic learning algorithm Mr-SBC
The examples are then **reclassified** by means of a variant of the k-NN algorithm tailored for transductive inference in MRDM. The idea is to classify each example $E \in (\text{WS U TS})$: on the basis of a $k$-sized neighborhood $N_k(E) = \{E_1, \ldots, E_k\}$ consisting of the $k$ examples of $(\text{WS U TS})$ closest to $E$ with respect to a dissimilarity measure $d$.

The **changeClass** procedure is in charge of changing the classification of borderline examples for which the entropy of the decision made by the classifier is maximum.

The entropy for each example is computed from the probabilities associated with each class $C_i$:

$$Entropy(E) = - \sum_{i=1,\ldots,L} P(\text{class}(E) = C_i) \times \log(P(\text{class}(E) = C_i))$$
The examples are then ordered according to the entropy function and the class label of at most the first k examples having Entropy(E) > MINENTROPY is changed.

The threshold k is necessary in order to avoid changing the class of several examples that would lead to erroneously change class of entire “clusters”.

Two distinct stopping criteria are used.
- Maximum number of iterations (MAX ITERS)
- Overlap between two sets of examples given by `computeOverlap` is significant
North-West England Census Data

Obtained from both census and digital maps data provided by the European project SPIN.

The goal of the classification task is to predict the value of the Jarman index (low or high value) deprivation factor by exploiting both other deprivation factors, mortality rate and geographical factors represented in topographic maps of the area.

The objects on each layer (such as urban area, green area, road net, rail net and water net) have been stored as tuples of relational tables including information on the object type (TYPE).
TRANSC vs. Mr-SBC on NWE Census Data: average misclassification error on the working sets. Number of bins (Nb) in Mr-SBC discretization is set to 10.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TRANSC</th>
<th>Mr-SBC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k=4</td>
<td>k=7</td>
</tr>
<tr>
<td>Avg 10-CV Error</td>
<td>23.38%</td>
<td>21.10%</td>
</tr>
<tr>
<td>%error loss</td>
<td>-2.97%</td>
<td>7.06%</td>
</tr>
<tr>
<td>Avg 20-CV Error</td>
<td>33.87%</td>
<td>34.41%</td>
</tr>
<tr>
<td>%error loss</td>
<td>0.00%</td>
<td>-1.60%</td>
</tr>
</tbody>
</table>

- Gain depends on value of K
- Results confirm an improved accuracy for the transductive setting with respect to the inductive one.
Munich Census Data

Concern the level of monthly rent per square meter for flats in Munich expressed in German Marks

The vectorized boundaries of subquarters, districts and zones as well as the map of public transport stops are available for this study. The objects included in these layers are stored in different relational tables.

The target attribute was represented by the “monthly rent per square meter”, whose values have been discretized into two values low = [2.0, 14.0] or high =]14.0, 35.0].
TRANSC vs. Mr-SBC on Munich Census Data: average misclassification error on the working sets. Nb=40

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TRANSC</th>
<th></th>
<th></th>
<th></th>
<th>Mr-SBC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>k=9</td>
<td>k=18</td>
<td>k=27</td>
<td>k=36</td>
</tr>
<tr>
<td>Avg 10-CV Error</td>
<td>28.99%</td>
<td>28.61%</td>
<td>28.36%</td>
<td>28.30%</td>
<td>28.15%</td>
</tr>
<tr>
<td>%error loss</td>
<td>7.17%</td>
<td>8.41%</td>
<td>9.19%</td>
<td>9.40%</td>
<td>9.86%</td>
</tr>
<tr>
<td>Avg 20-CV Error</td>
<td>37.25%</td>
<td>36.30%</td>
<td>36.73%</td>
<td>36.67%</td>
<td>36.78%</td>
</tr>
<tr>
<td>%error loss</td>
<td>1.44%</td>
<td>3.94%</td>
<td>2.81%</td>
<td>2.98%</td>
<td>2.68%</td>
</tr>
</tbody>
</table>

Again results confirm an improved accuracy for the transductive setting with respect to the inductive one.
Thanks to them for their valuable contribution to this research topic:

Donato Malerba
Annalisa Appice
Details


M. Ceci: Transductive Learning from Textual Data with Relevant Example Selection. *DEXA (2) 2010*: 470-484


A. Appice, M. Ceci, D. Malerba: Transductive learning for spatial regression with co-training. *SAC 2010*: 1065-1070