UNIVERSITÀ DEGLI STUDI FIRENZE

Transductive learning over networked text data

Comparison between transductive and inductive SVM

Daniele Mugnai

Firenzs,24/02/2020

INTRODUCTION

Project aims

- The implementation of Transductive Support Vector Machine as Joachims describes in his paper [1] ,[2]
- Evaluate TSVM in classication of semi labeled networked text data [3]

Networked Text Data

We can represent a networked text dataset as an indirect graph $G = \{V, E, X, Y\}$ where V is the set of vertices; E is the set of edges , $X = \{x_1, x_2, .. x_n\}$ is the **Content matrix** representing the content of each vertices; $Y = \{y_1, y_2, .. y_n\}$ are the **labels** of the vertices in V.

We also consider netwok **outliers** as the vertices whose attributes significantly deviate from vertices having same label and graph structure.

Semi-Supervised Learning

- In real life application it is common the case that collecting sufficient data is difficult as it requires huge amount of human efforts.
- For this reason we can speak about **Semi-Supervised learning,** which utilizes a small number of labeled data and a large number of unlabeled data to build a classifier.
- We can divide the vertices in a Networked dataset into **labeled** vertices and **unlabeled** vertices.

Learning Task

Given a set of l training examples:

$$X_{train} = (\mathbf{x}_{l_1}, \mathbf{x}_{l_2}, ..., \mathbf{x}_{l_l}) \qquad Y_{train} = (\mathbf{y}_{l_1}, \mathbf{y}_{l_2}, ..., \mathbf{y}_{l_l}),$$

$$\mathbf{x}_i \in \mathbb{R}^d. \qquad \mathbf{y}_i \in \{-1, +1\}.$$

Each training sample consists of a feature vector and a binary label.

In contrast to the inductive setting, it is also given a sample of u test examples from the same distribution of the training set.

$$X_{test} = (\mathbf{x}_{u_1}, \mathbf{x}_{u_2}, ..., \mathbf{x}_{u_l}).$$

Learning Task

The transductive learner uses $X_{train}, Y_{train}, \text{ and } X_{test}$ to produce predictions for the test examples:

$$Y^*_{test} = (\mathbf{y}^*_{u_1}, \mathbf{y}^*_{u_2}, \ldots, \mathbf{y}^*_{u_u}),$$

The aim is to minimize the fraction of erroneous predictions

$$Err_{test}(Y^*_{test}) = \frac{1}{u} \sum_{i \in S_{test}} \delta_{0/1}(\mathbf{y}^*_i, \mathbf{y}_i), \qquad \delta_{0/1}(a,b) \text{ is zero if } a = b,$$

TRANSDUCTIVE SVM

Transductive SVM

Linearly separable case

$$
\begin{array}{ll}
\text{minimize:} & V(\mathbf{y}^*_{u_1}, ..., \mathbf{y}^*_{u_u}, \mathbf{w}, b) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} \\
\text{subject to:} & \forall_{i=1}^{l} : \mathbf{y}_{l_i}[\vec{w} \cdot \mathbf{x}_{l_i} + b] \geq 1 \\
 & \forall_{j=1}^{u} : \mathbf{y}^*_{u_j}[\vec{w} \cdot \mathbf{x}^*_{u_j} + b] \geq 1 \\
 & \forall_{j=1}^{u} : \mathbf{y}^*_{u_j} \in \{-1, +1\}
\end{array}
$$

Solving this problem means find a labeling of the test data and a hyperplane so that this hyperplane separates both training and test data with maximum margin.
An inductive SVM also finds a large-margin hyperplane, but it considers only the training vectors while ignoring all test vectors.

UNIVERSITÀ DEGLI STUDI FIRENZE

Transductive svm

Non-separable case

To be able to handle non separable data, it can be introduce slack variable ξ_i similar to the way of inductive SVM

$$\text{min: } W(\mathbf{y}^*_{u_1}, \dots, \mathbf{y}^*_{u_u}, \mathbf{w}, b, \xi_1, \dots, \xi_l, \xi^*_1, \dots, \xi^*_u) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{l}\xi_i + C^*\sum_{j=1}^{u}\xi^*_j$$

$$\begin{aligned}
\text{s.t.: } & \forall_{i=1}^{l} : y_{l_i}[\mathbf{w}\cdot\mathbf{x}_{l_i} + b] \geq 1 - \xi_i \\
& \forall_{j=1}^{u} : \mathbf{y}^*_{u_j}[\mathbf{w}\cdot\mathbf{x}^*_{u_j} + b] \geq 1 - \xi^*_j \\
& \forall_{j=1}^{u} : \mathbf{y}^*_{u_j} \in \{-1, +1\} \\
& \forall_{i=1}^{l} : \xi_i \geq 0 \\
& \forall_{j=1}^{u} : \xi^*_j \geq 0
\end{aligned}$$

C and C* are parameters set by user.
Transductive SVMs can be extended to include kernels

Transductive svm

Non-separable case

The problem describes can be solved optimally for a small number of test examples tryning all possible assignments of y_i^* to two classes.

This approach become intractable for test set with more than 10 examples.

The algorithm proposed next is designed to handle larger test set with 10,000 test examples and more.
It finds an approximate solution to the problem using a form of local search. It starts with a initial instantiation of the variables and at each iteration the current instantiation is modified closer to the solution.

Algorithm

The function $solve\_svm\_qp$ is used in the algorithm as subprocedure. It takes as input the training set ,test set, temporary assigned label for test set and the parameters C, C_{-}^{*}, C_{+}^{*}
It refers to a quadratic problem of the following type:

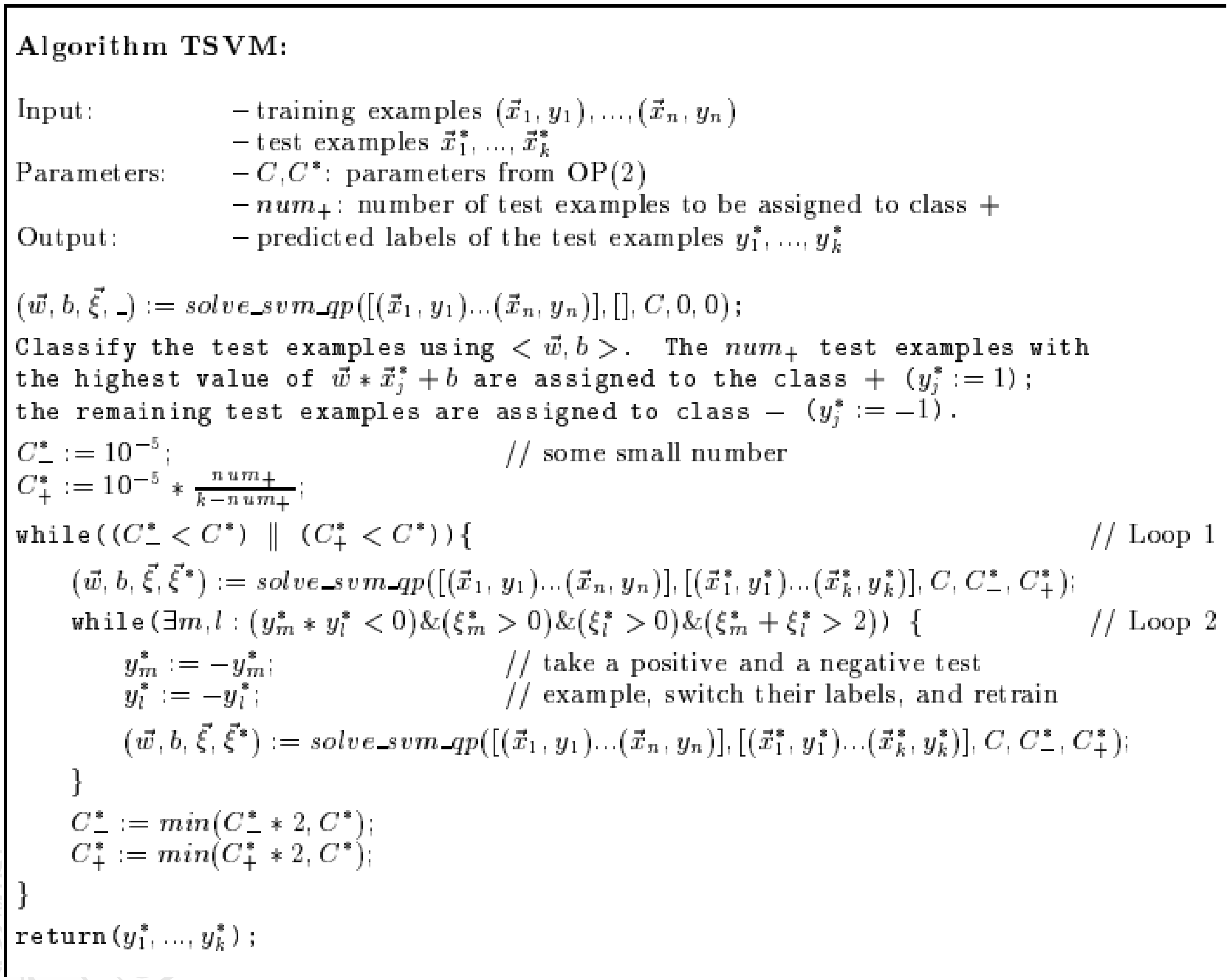
$$\text{Minimize over } (\vec{w}, b, \vec{\xi}, \vec{\xi}^{*}):$$

$$\frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n} \xi_i + C_{-}^{*}\sum_{j:y_j^*=-1} \xi_j^* + C_{+}^{*}\sum_{j:y_j^*=1} \xi_j^*$$

$$\text{subject to: } \quad \forall_{i=1}^{n} : y_i[\vec{w} \cdot \vec{x_i} + b] \geq 1 - \xi_i$$

$$\forall_{j=1}^{k} : y_j^*[\vec{w} \cdot \vec{x_j} + b] \geq 1 - \xi_j^*$$

This optimization problem is similar to an inductive SVM.



Algorithm TSVM:

Input:	– training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
	– test examples $\vec{x}^*_1, \ldots, \vec{x}^*_k$
Parameters:	– C, C^*: parameters from OP(2)
	– num_+: number of test examples to be assigned to class +
Output:	– predicted labels of the test examples $y^*_1, \ldots, y^*_k$

```
$(\vec{w}, b, \vec{\xi}, \_) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [], C, 0, 0)$;
Classify the test examples using $< \vec{w}, b >$.  The $num_+$ test examples with
the highest value of $\vec{w} * \vec{x}^*_j + b$ are assigned to the class + $(y^*_j := 1)$;
the remaining test examples are assigned to class – $(y^*_j := -1)$.
$C^*_- := 10^{-5}$;                                  // some small number
$C^*_+ := 10^{-5} * \frac{num_+}{k - num_+}$;
while($(C^*_- < C^*) \parallel (C^*_+ < C^*)$){                                  // Loop 1
    $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [(\vec{x}^*_1, y^*_1) \ldots (\vec{x}^*_k, y^*_k)], C, C^*_-, C^*_+)$;
    while($\exists m, l : (y^*_m * y^*_l < 0) \& (\xi^*_m > 0) \& (\xi^*_l > 0) \& (\xi^*_m + \xi^*_l > 2)$) {      // Loop 2
        $y^*_m := -y^*_m$;                       // take a positive and a negative test
        $y^*_l := -y^*_l$;                        // example, switch their labels, and retrain
        $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [(\vec{x}^*_1, y^*_1) \ldots (\vec{x}^*_k, y^*_k)], C, C^*_-, C^*_+)$;
    }
    $C^*_- := min(C^*_- * 2, C^*)$;
    $C^*_+ := min(C^*_+ * 2, C^*)$;
}
return($y^*_1, \ldots, y^*_k$);
```

ALGORITHM

Sottotitolo arial regular 18 pt

- First step: solving inductive SVM on the training data
- The ratio of test examples that are classified as positive is specified by the user or estimated from the ratio of positive to negative examples in the training set. This ratio is maintained throughout the optimization process to avoid degenerate solutions that assign all test examples to the same class
- Loop 1 increases the influence of the test example by incrementing the cost factor up to user-defined value C*
- Loop 2 improves the solution by switching the label of a pair of test examples. The criterion is to find two examples s.t. changing his labels leads to a decrease in the objective function.

EXPERIMENTS

Experiments

- For this work, it have been done two type of experiments: the first with an artificial dataset to validate the implementation of Transductive SVM and a second on public avaiable dataset.
- The transductive learning has the purpose to minimize the test error. For this reason we select the accuracy as metric to compare the methods.
- The accuracy is defined: $accuracy(Y,\hat{Y})=\frac{1}{N}\sum_0^N \delta_{0/1}(y_i,\hat{y}_i)$,

  where $Y=\{y_1,y_2,..y_N\}$ are the real labels , $\hat{Y}=\{\hat{y}_1,\hat{y}_2,..\hat{y}_N\}$ are the predicted labels

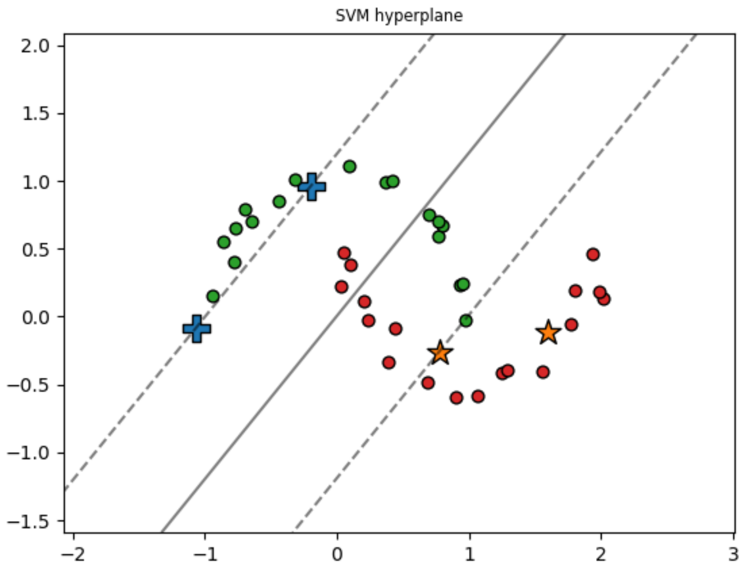
  and $\delta_{0/1}(a,b)=1\ if\ a=b\ else\ 0$

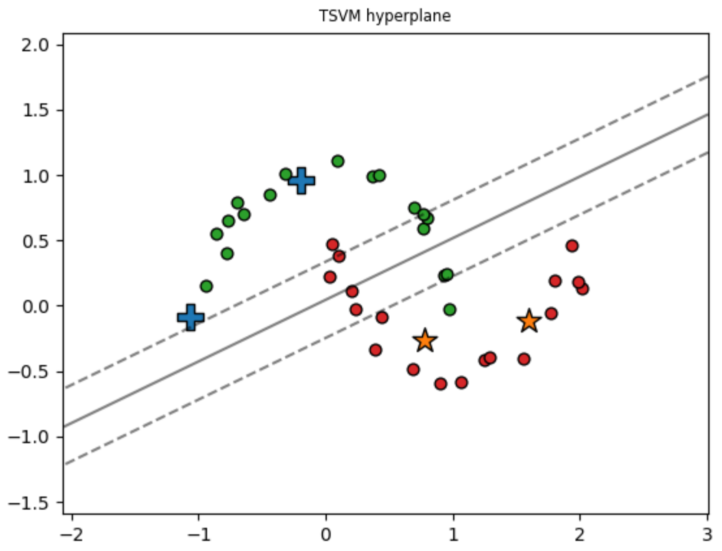
- TSVM is implemented using scikit learn packages and it's compare to C-SVM baselines(Scikit-learn)

Experiment on artificial dataset

- Firstly it is implemented experiment on one 2-dimensional artificial dataset
- Generating 40 points, 20 for each class
- For each class 2 points are taken as labelled and the rest as unlabeled
- For SVM and TSVM the parameters C=1 , C*=1 with linear kernel.

Experiment on artificial dataset





SVM	TSVM
72.2	83.1

✚ Labelled positive	● Unlabelled positive
★ Labelled negative	● Unlabelled negative

EXPERIMENTS ON BENCHMARK DATASET

Datasets

- The **Cora** dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.
- The **Citeseer** dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words
- The **Pubmed** Diabetes dataset consists of 19717 scientific publications from PubMed database pertaining to diabetes classified into one of three classes. The citation network consists of 44338 links. Each publication in the dataset is described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words

Parameter settings

- It was randomly selected 20 istances from each class and treated them as the labelled data for training.
- It was randomly sampled 1,000 of the remaining data as test set and treated them as unlabeled.
- For TSVM and SVM it was chosen Gaussian Kernel(«RBF») , C and C* choseen through grid search in the set {1,2,5,10} and gamma is set 0.1
- The classification problem is a multiclass problem, it was used One-vs-All strategy as implemented also in SVM-light
- The results obtained was compared with the ones of the paper: they used SVM-light packages for SVM and TSVM

Outliers Perturbation

To evaluete robustness of the proposed method to outliers, the datasets were perturbed using this algorithm.

- Select 5% of the vertices in the labeled and unlabeled data. Modify their attribute following this scheme:
  1. For a selected node i randomly pick another $m = \min(100, \frac{n}{4})$ vertices from the whole networked dataset
  2. Select the node j with the most different attribute from node i from the m vertices. To select the most different it was chosen the one that maximes $\left\|x_i - x_j\right\|_2$
  3. Replace the attribute of node i with the attribute of node j . $x_i := x_j$

RESULTS

Results

Classification accuracy in percentage using orginal datasets and noisy dataset(mark with *)

	Cora		Citeseer		Pubmed	
	Experiments	Article	Experiments	Article	Experiments	Article
SVM	**58.0**	58.3	**59.3**	58.0	**69.5**	73.5
TSVM	**59.5**	58.3	**61.5**	62.3	**72.1**	66.9

	Cora*		Citeseer*		Pubmed*	
	Experiments	Article	Experiments	Article	Experiments	Article
SVM	**54,2**	58.1	**56,6**	58	**68,2**	73.5
TSVM	**53.1**	55.6	**59.5**	61.4	**66.4**	66.5

Conclusion

- In the case of not noisy dataset the Transductive SVM performs better than the Inductive SVM
- With nosy dataset Transductive SVM performs worst: different distribution of test set respect training set
- Another problem of Tranductive SVM is the parameter num_+: it has to be specified at the beginning of the learning. It’s difficult to estimate the value: using the fraction of positive examples in the training set may lead a large estimating error in the case of small training set.

References

[1] Joachims, Thorsten. (2001**). Transductive Inference for Text Classification Using Support Vector Machines**. ICML.

[2] Joachims Thorsten. (2001). **Learning to classify text using Support Vector Machines. Cornell University.**

[3] Jiongqian Liang and Peter Jacobs and Jiankai Sun and Srinivasan Parthasarathy. (2017). **Semisupervised Embedding in Attributed Networks with Outliers**

UNIVERSITÀ
DEGLI STUDI
FIRENZE

Thank you for attention

Daniele Mugnai

Firenzs,24/02/2020