Transductive Inference for Text Classification using Support Vector Machines

By : Thorsten Joachims

Speaker : Sameer Apte

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Introduction

- Text classification one of the key techniques for organizing online information (organize document databases, filter spam for email etc.)
- Learn classifiers from examples as hand coding is impractical
- Crucial for learner to generalize well using little training data
- Take transductive approach to tackle problem of learning from small training samples

Introduction

- Examples of transductive text classification task :
  - Relevance Feedback
  - Netnews Filtering
  - Reorganizing a document collection
- Use Transductive Support Vector Machines (TSVMs) .They substantially improve already excellent performance of SVMs for text classification
- New algorithm for efficiently training TSVMs with 10,000 examples and more

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Transductive Inference

- Problem of estimating the values of a function at given points of interest (Introduced by Vapnik)
- In inductive inference, one uses given empirical data to find the approximation of a functional dependency (inductive step) and then uses this approximation to evaluate values of a function at points of interest (deductive step)
- In transductive inference, we try to estimate the values of a function at the points of interest in one step

Transductive Inference

- For example : problem of learning from small training samples
- Inductive approach : Learner tries to induce a decision function which has a low error rate on the whole distribution of examples for the particular learning task
- In many situations we do not care about the particular decision function, but rather we classify a given set of examples (*test set*) with as few errors as possible [Transductive Inference]

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Text Classification

- Supervised learning problem
- Goal is the automatic assignment of documents to a fixed no. of semantic categories
- Learn classifiers from examples which assign categories automatically
- Documents (strings of characters), have to be transformed into a representation suitable for the learning algorithm and the classification task

Text Classification

- IR research suggests that word stems work well as representations
  - Example : "computes", "computing", and "computer" are all mapped to same word stem "comput"
- Attribute value representation of text
- Each word w_i corresponds to a feature with TF(w_i,x), the no. of times word w_i occurs in document x , as its value

Text Classification

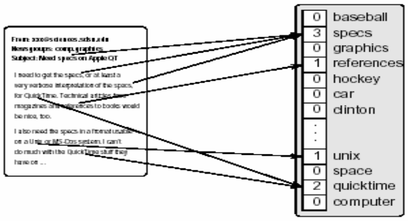

From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subject: Need specs on Apple QT
I need to get the specs, or at least a very verbose interpretation of the specs, for QuickTime. Technical articles from magazines and references to books would be nice, too.
I also need the specs in a format usable on a Unix or MS-Dos system. I can't do much with the QuickTime stuff they have on ...
0 baseball
3 specs
0 graphics
1 references
0 hockey
0 car
0 clinton
1 unix
0 space
2 quicktime
0 computer

Figure 1: Representing text as a feature vector.

Text Classification

- For improved performance , we refine the basic representation by scaling the dimensions of the feature vector with their *inverse document frequency* IDF(w_i)
- IDF(w_i) = log(n / DF(w_i))
  - where n =total no. of documents

    DF(w_i) = no. of documents the word w_i occurs in
- IDF of a word is low if it occurs in many documents and is highest if the word occurs in only one

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Transductive Support Vector Machines (TSVMs)

- SVMs basic idea :
  - Choose a separating plane based on maximizing the notion of a margin
  - How to pick best separating plane
    - Define a set of inequalities we want to satisfy
    - Use advanced optimization methods (eg., linear programming) to find satisfying solutions
- SVMs have mechanisms for
  - Noise
  - Non-linear separating surfaces

TSVMs

- Take into account a particular test set and try to minimize misclassifications of just those particular examples
- Learner L is given a hypothesis space H of functions h : X- > { 1,1} and sample S_{train} of n training examples :
  - $(\vec{x}_1,y_1),(\vec{x}_2,y_2),\ldots\ldots.,(\vec{x}_n,y_n)$
- Also given sample S_{test} of k test examples :
  - $\vec{X}_1{}^*, \vec{X}_2{}^*,\ldots\ldots\ldots, \vec{X}_k{}^*$ from same distribution

TSVMs

- Transductive learner L aims to select a function h_L = L(S_{train},S_{test}) from H using S_{train} and S_{test} so that the expected no. of erroneous predictions R(L)

  $$R(L) = \int \frac{1}{k} \sum_{i=1}^{k} \Theta(h_L(\vec{x}_i^*), y_i^*)\, dP(\vec{x}_1, y_1) \cdots dP(\vec{x}_k^*, y_k^*)$$

  on the test examples is minimized. Ə(a,b) is zero if a=b, otherwise it is one.
- What information we get from studying test sample and how can we use it ?

TSVMs

- Training and test sample split the hypothesis space H into a finite no. of equivalence classes H′
- Two functions from H belong to the same equivalence class if they both classify the training and test sample in the same way
- This reduces the learning problem from finding a function in the possibly infinite set H to finding one of finitely many equivalence classes H′
- These equivalence classes can be used to build a structure of increasing VC-Dimension for *structural risk minimization*[Vapnik]

$$H'_1 \subset H'_2 \subset \cdots \subset H'$$

TSVMs

- Using prior knowledge about the nature of the learning task we can build a more appropriate structure and learn more quickly.
- We can build the structure based on the margin of separating hyperplanes on both the training and the test data
- With the size of the margin we can control the maximum no. of equivalence classes (i.e. the VC dimension) [Theorem by Vapnik]

TSVMs [Optimization Problem]

- Minimize over $(y_1^*, y_2^*, \ldots\ldots\ldots, y_n^*, \vec{w}, b)$:

  $$\tfrac{1}{2}||\vec{w}||^2$$

  subject to $\forall^{n}_{i=1}: y_i[\vec{w}.\ \vec{x_i} + b] >= 1$

  $\forall^{k}_{j=1}: y_j^*[\vec{w}.\ \vec{x_j^*} + b] >= 1$
- Solving this problem means finding a labelling $y_1^*, \ldots\ldots\ldots, y_k^*$ of the test data and a hyperplane $<\vec{w}, b>$, so that this hyperplane separates both training and test data with maximum margin.
- Figure 2 illustrates this :

TSVMs

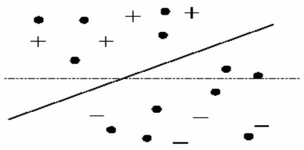


Figure 2: The maximum margin hyperplanes. Positive/negative examples are marked as +/−, test examples as dots. The dashed line is the solution of the inductive SVM. The solid line shows the transductive classification.

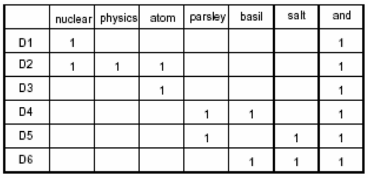
Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

TSVMs for Text Classification

- Properties of text classification task
  - High dimensional input space
  - Document vectors are sparse
  - Few irrelevant features
- SVMs do well on this setting
- Why TSVMs better ?
  - Words occur in strong co-occurrence patterns
  - Many IR approaches try to exploit this cluster structure
  - TSVMs exploit co-occurrence information as prior knowledge about the learning task

TSVMs for Text Classification(Example)



	nuclear	physics	atom	parsley	basil	salt	and
D1	1						1
D2	1	1	1				1
D3			1				1
D4				1	1		1
D5				1		1	1
D6					1	1	1

Figure 3: Example of a text classification problem with co-occurrence pattern. Rows correspond to documents, columns to words. A table entry of 1 denotes the occurrence of a word in a document.

TSVMs for Text Classification(Example)

- D1 given class A, D6 given class B
- How to classify D2,D3 and D4
- D2 and D3 :class A   D3 and D4 :class B
- For the TSVM, the co-occurrence information in the test data was analyzed and two clusters were found {D1,D2,D3} and {D4,D5,D6}
- TSVM gives same classification as above. It gives the maximum margin solution
- Maximum margin bias reflects our prior knowledge about text classification well.

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Experiments

- 3 test collections :
  - Reuters-21578 collected from Reuters newswire in 1987
    - 9603 training documents and 3299 test documents
  - WebKB collection of www pages [CMU text-learning group]
    - Only the classes course,faculty,project and student are used
    - 4183 examples, pages from Cornell University used for training , all other pages for testing

Experiments

- Test collections(Contnd.)
  - Ohsumed corpus
    - 10000 training docs and 10000 test docs
    - Assign documents to one or multiple categories of the 5 most frequent "diseases" categories
    - Doc belongs to a category if it is indexed with at least one indexing term from that category
- Performance Measures
  - Precision/Recall- Breakeven Point

Experiments (Performance Measures)

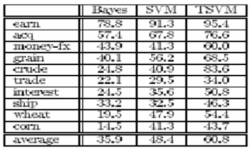
- Precision/Recall- Breakeven Point
  - Precision : Probability that a document predicted to be in class "+" truly belongs to this class
  - Recall : Probability that a document belonging to class "+" truly is classified into this class
- P/R breakeven point is the value for which precision and recall are equal
- Transductive SVM uses the breakeven point for which the no. of false positives equals the no. of false negatives

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Results

- Comparison of different classifiers :
  - 17 training docs and 3299 test docs



	Bayes	SVM	TSVM
earn	78.8	91.3	95.4
acq	57.4	67.8	76.6
money-fx	43.9	41.3	60.0
grain	40.1	56.2	68.5
crude	24.8	40.9	83.6
trade	22.1	29.5	34.0
interest	24.5	35.6	50.8
ship	33.2	32.5	46.3
wheat	19.5	47.9	54.4
corn	14.5	41.3	43.7
average	35.9	48.4	60.8

Figure 5: P/R-breakeven point for the ten most frequent Reuters categories using 17 training and 3,299 test examples. Naive Bayes uses feature selection by empirical mutual information with local dictionaries of size 1,000. No feature selection was done for SVM and TSVM.

Results

- Effect of varying the size of the training set

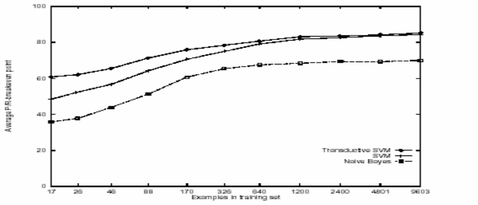


Figure 6: Average P/R-breakeven point on the Reuters dataset for different training set sizes and a test set size of 3,299.

Results

- Influence of size of the test set

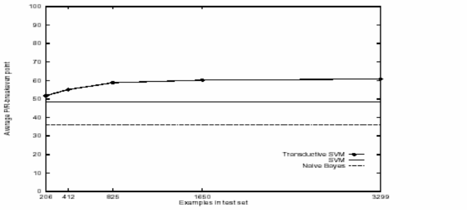
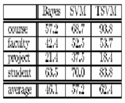


Figure 7: Average P/R-breakeven point on the Reuters dataset for 17 training documents and varying test set size for the TSVM.

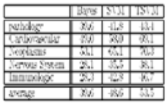
Results

- Comparison of different classifiers (WebKB and Ohsumed):



	Bayes	SVM	TSVM
course	57.2	68.7	93.8
faculty	42.4	52.5	53.7
project	21.4	37.5	18.4
student	63.5	70.0	83.8
average	46.1	57.2	62.4

Figure 8: Average P/R-breakeven points for the WebKB categories using 9 training and 3957 test examples. Naive Bayes uses a global dictionary with the 2,000 highest mutual information words. No feature selection was done for the SVM. Due to the large number of words, the TSVM used only those words which occur at least 5 times in the whole sample.



	Bayes	SVM	TSVM
pathology	39.6	41.8	43.4
Cardiovascular	49.0	58.0	69.1
Neoplasms	53.1	65.1	70.3
Nervous System	28.1	35.5	38.1
Immunologic	28.3	42.8	46.7
average	39.6	48.6	53.5

Figure 9: Average P/R-breakeven points for the Ohsumed categories using 120 training and 10,000 test examples. Here, Naive Bayes uses local dictionaries of 1,000 words selected by mutual information. No feature selection was done for the SVM. The TSVM again uses all words that occur at least 5 times in the whole sample.

Results

- Effect of varying the size of the training set for category *course*

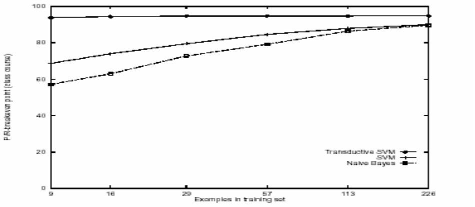


Figure 10: Average P/R-breakeven point on the WebKB category course for different training set sizes.

Results

- Effect of varying the size of the training set for category *project*

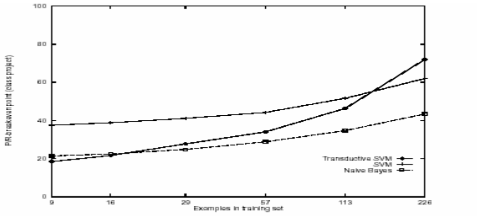


Figure 11: Average P/R-breakeven point on the WebKB category project for different training set sizes.

Outline

- Introduction
- Transductive Inference
- Text Classification
- Transductive Support Vector Machines(TSVMs)
- TSVMs for Text Classification
- Experiments
- Results
- Related Work and Conclusions

Related Work

- Naïve Bayes classifier [Nigam] using EM algorithm
  - Independence assumption is violated for text
- Co training [Blum and Mitchell] uses unlabeled data and describes the problem by multiple representations
  - Boosting scheme that exploits a conditional independence between these representations

Conclusions

- Margin of separating hyperplanes is a natural way to encode prior knowledge for learning text classifiers
- Test set can be used as additional source of information about margins by taking transductive approach
- Trasductive approach shows improvements over the currently best performing method
  - Most substantially for small training samples and large test sets

References

- Transductive Inference for Text Classification using Support Vector Machines [Joachims]
- Statistical Learning Theory [Vapnik]

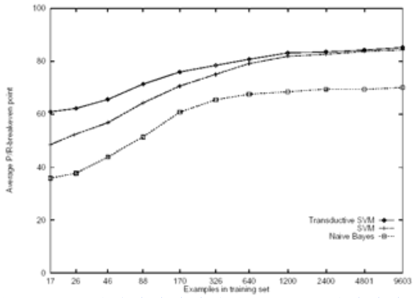
Transductive Inference for Text Classification using Support Vector Machines (Joachims, 1999)

Comments by Alex Kosolapov

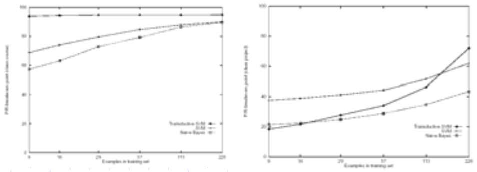
Performance

- Goal of transductive inference:
  - "classify a given set of examples (i.e., a test set) with as few errors as possible"

Large Training Sets



Smaller Training Sets



Question

Relationship between the time required for classification on test set and accuracy in a TSVM?

- Scaling up for large test sets?
- Joachims does not report the training/test times for TSVM in the paper

- Joachims, T. 2002 Learning to Classify Text using Support Vector Machines. Methods, Theory, and Algorithms. Kluwer Academic Publishers, ISBN 0-7923-7679-X

Reference

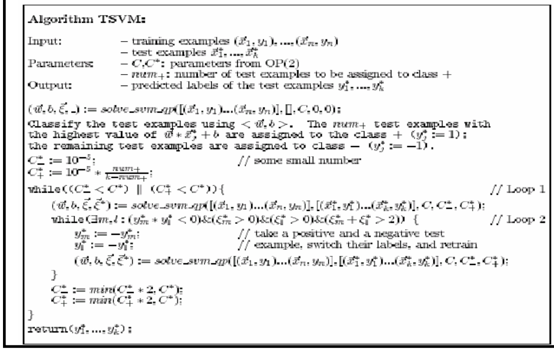
Joachims, T. (1999) Transductive inference for text classification using support vector machines. ICML-99

Transductive Inference for Text Classification using Support Vector Machines

By
Thorsten Joachims

Presented - Sameer Apte
Comments – Kiran Vuppla

Algorithm TSVM



```
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(($C_-^* < C^*$) || ($C_+^* < C^*$)){                                    // 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 TSVM

Objective : solve combinatorial optimization problem OP2.

Algorithm is designed to handle large datasets for classification

Key Idea: Labeling the test data based on the classification of an Inductive SVM to improve solution by decreasing objective function

Algorithm TSVM

- Input : - training examples $(\vec{x}_1, y_1),....(\vec{x}_n, y_n)$
  - test data $\vec{x}_1^*,... \vec{x}_n^*$
- Parameters : - C, C*: parameters from OP2
  - num$_+$: # of test examples to be assigned +
- Output: - predicted labels of the test examples $y_1^*,... y_n^*$

Algorithm TSVM

- Training an inductive SVM on training data and classifying test data
- Increasing the influence of test examples by incrementing C_-^* and C_+^*
- Changing the labels of the examples decreases the objective function

solve_svm_qp

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

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

subject to :

$$\forall_{i=1}^{n} : y_i[\vec{w}.\vec{x}_i + b] \geq 1 - \xi_i$$

$$\forall_{j=1}^{n} : y_i[\vec{w}.\vec{x}_j + b] \geq 1 - \xi_j^*$$

SVM^{light} [Joachims] is used to solve the above problem