

Transductive Inference for Text Classification using Support Vector Machines

Thorsten Joachims

International Conference on Machine Learning, 1999

Presented by Joe Drish

CSE 254: Seminar on Learning Algorithms, 2001

Department of Computer Science and Engineering

University of California, San Diego

Introduction

Main Goals

- Introduce a new method for text classification - Transductive Support Vector Machines (TSVMs)
- Analyze why TSVMs are well-suited for text classification
- Describe a novel algorithm for training TSVMs
- Experimentally demonstrate classification improvements using TSVMs compared to standard inductive learning methods

Talk Outline

- I. **Text classification**
- II. Transductive inference
- III. TSVMs for text classification
- IV. TSVM algorithm
- V. Experimental results
- VI. Conclusions and future work

Text Classification

Problem

- Classify documents into multiple, exactly one, or no semantic categories
- Learn a classifier to assign categories automatically

Applications

- Netnews Filtering - find interesting news articles
- Reorganizing a document collection - automatically classify document databases after new categories are introduced

Document Preprocessing

Information Extraction

- Documents are strings of characters
- Words are represented as word stems
- Example: “computes”, “computing”, and “computer” are all mapped to the word stem “comput”
- Information retrieval research suggests that word stems work well without information loss

Documents as Feature Vectors

Feature Vectors (see Figure 1)

- Each document has one feature vector, indexed by word stems
- Each vector entry is $TF(w_i, x)$, the number of times word stem w_i occurs in document x

Scaling by Inverse Document Frequency (IDF)

- Each feature vector entry is multiplied by

$$IDF(w_i) = \log \left(\frac{n}{DF(w_i)} \right)$$

where n is the total number of documents, and $DF(w_i)$ is the number of documents the word w_i occurs in

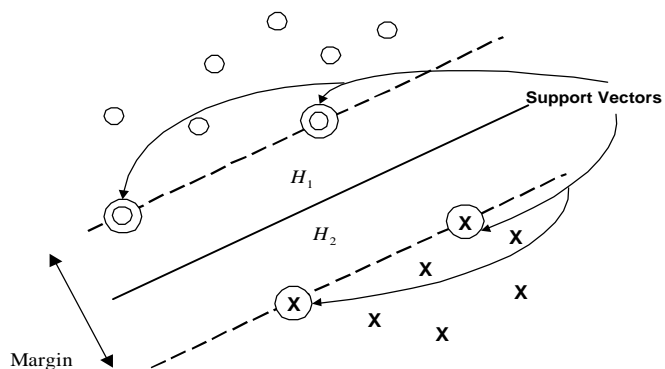
- IDF scaling assigns greater weight to word stems that are infrequent across all documents, and lesser weight to frequent word stems

Talk Outline

- I. Text classification
- II. **Transductive inference**
- III. TSVMs for text classification
- IV. TSVM algorithm
- V. Experimental results
- VI. Conclusions and future work

Inductive Support Vector Machines

- Input vectors are separated into two regions: H_1 and H_2
- Margin is maximized given minimal separation error
- Data points that lie on the margin are “support vectors”



Inductive versus Transductive Learning

Objectives of Inductive and Transductive Inference

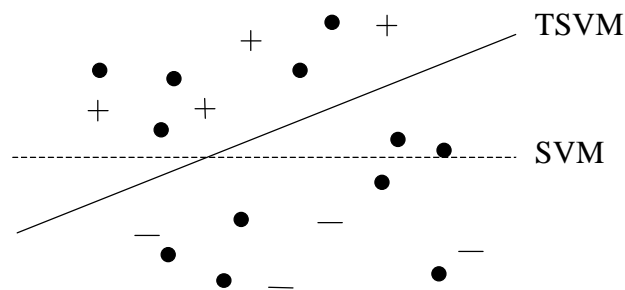
- Inductive learning: generalize for any future test set
- Transductive learning: predict the classes for a specific test set
- In transduction we use information from the given test set

Transduction using Support Vector Machines

- Inductive Support Vector Machines (SVMs) learn a decision boundary between two classes to predict labels for future test sets
- Transductive Support Vector Machines (TSVMs) attempt to minimize the number of erroneous predictions on a specific test set
- A variation of supervised and unsupervised learning

Transductive Support Vector Machines

- Positive/negative training examples are marked +/-
- Test examples are dots
- The solid line gives the TSVM separating hyperplane



Redrawn from figure 2 in [Joachims, 1999]

Talk Outline

- I. Text classification
- II. Transductive inference
- III. **TSVMs for text classification**
- IV. TSVM algorithm
- V. Experimental results
- VI. Conclusions and future work

TSVMs and Text Classification

Text Classification Task Features

- High dimensional input space (10,000 features)
- Document feature vectors are sparse
- Every feature is important, since most words are relevant

What makes TSVMs good for this task?

- TSVMs inherit properties of SVMs, which work well
- TSVMs exploit co-occurring patterns of text

Alta Vista Search Example (number of hits in year 2001)

```
pepper, salt: 181,827  
pepper, physics: 19,425  
salt: 1.9 million  
physics: 4.2 million
```

TSVMs Using the Test Set: An 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 in [Joachims, 1999]

- Documents **D1** and **D6** are the training feature vectors
- Documents D2 through D5 are the test feature vectors
- D1, D2, and D3 are classified into class A
- D4, D5, and D6 are classified into class B
- ✓ This is possible since the test vectors D2 and D3 share a common word (atom), as do D4 and D5 (parsley)

Talk Outline

- I. Text classification
- II. Transductive inference
- III. TSVMs for text classification
- IV. **TSVM algorithm**
- V. Experimental results
- VI. Conclusions and future work

TSVM Training Algorithm Overview

Algorithm Overview

- Input:
- labeled training examples $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$
 - unlabeled test examples $\vec{x}_1^*, \dots, \vec{x}_k^*$
 - C, C^* from OP(2) in [Joachims, 1999]
 - num+: anticipated number of positive test examples
- Output:
- predicted labels of the test examples y_1^*, \dots, y_k^*

User Parameters

- C and C^* specify the SVM margin size
- num+ allows the tradeoff of recall versus precision
 - recall: proportion of items in the category that are actually placed in the category
 - precision: proportion of items placed in the category that are really in the category

TSVM Training Algorithm Description

Algorithm Idea

- Refer to Figure 4 in the paper, [Joachims, 1999]
- First label the test data based on inductive SVM classification
- Set the cost factors C_-^* and C_+^* to a small number

Outer loop (loop 1)

- Increment the cost factors up to the user defined value of C^*

Inner loop (loop 2)

- Locate two test examples for which changing the class labels leads to a decrease in the current objective function OP(2)
- If these two examples exist, switch them

Algorithm Notes

- SVM^{light} (Joachims) is web software for the inductive SVM

TSVM Inner Loop

Motivation

- Goal is to minimize objective function $OP(2)$
- Algorithm will switch two examples that further minimize $OP(2)$, if two such examples exist
- Same example can have its label switched repeatedly
- $OP(2)$ decreases with every iteration
- Converges in a finite number of steps (proof given in paper)

Issues

- Why is it reasonable to switch to examples - randomness?

Talk Outline

- I. Text classification
- II. Transductive inference
- III. TSVMs for text classification
- IV. TSVM algorithm
- V. **Experimental results**
- VI. Conclusions and future work

Test Set Collections

Reuters-21578

- Consists of Reuters news data collected in 1987.
- ModApte split: 9,603 (75%) training and 3,299 (25%) test documents
- Can be in one or more of 10 classes (e.g., earn, grain, crude, etc.)

WebKB collection

- A collection of World Wide Web pages
- 4,183 examples: Cornell University for training, others for testing
- Can be in only one of 4 classes: course, faculty, project, student

Ohsumed corpus

- Medical documents compiled in 1991
- 10,000 training examples; 10,000 testing examples
- Can be in one or more of 5 classes (e.g., pathology, neoplasms, etc.)

Performance Metrics

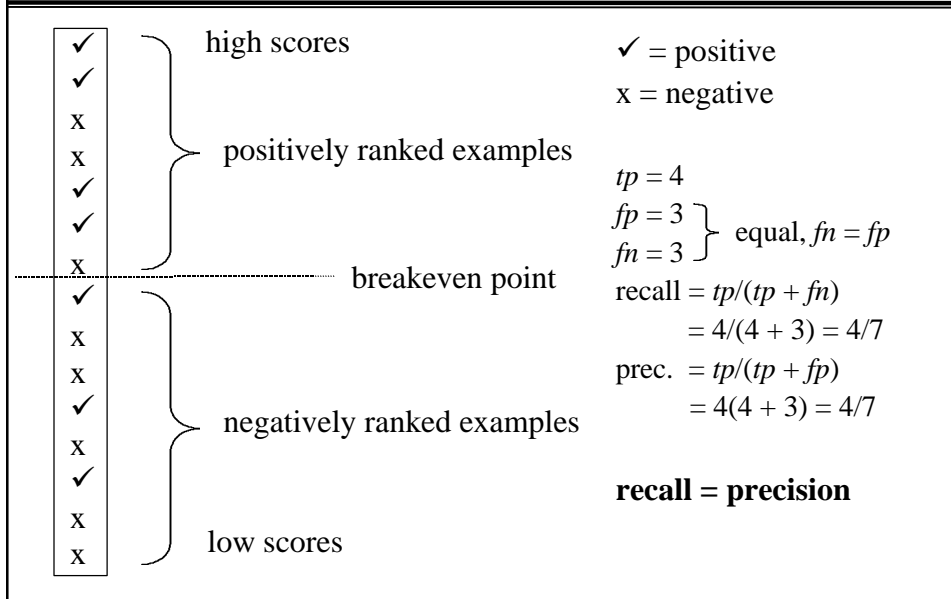
Recall and Precision (defined intuitively before)

- recall: $tp/(tp + fn)$, where tp is true positives, and fn is false negatives
- precision: $tp/(tp + fp)$, where fp is false positives

Precision/Recall (P/R) Breakeven Point

- Standard measure of performance in text classification
- Defined as the value for which precision and recall are equal
- Number of false positives equals number of false negatives

Breakeven point: Recall = Precision



Reuters Experiment

| | 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 in [Joachims, 1999]

61.3?

Results

- 17 training and 3,299 test examples
- The TSVM gives better performance on all classes
- TSVMs are better for small training sets (Figure 6)
- TSVMs are less superior for larger training sets (Figure 7)

WebKB Experiment

| | 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 in [Joachims, 1999]

Results

⇒ 9 training and 3,957 test examples

⇒ **course** is especially good, **project** is especially bad. Why?

- **course** pages at Cornell do not give topic information
- With more training examples SVM catches up to TSVM (figure 10)
- **project** is smallest class (1/9), and pages give topic information
- With more training examples TSVM overcomes SVM (figure 11)

Ohsumed Experiment

| | 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 |

Redrawn from figure 9 in [Joachims, 1999]

Results

- 120 training and 10,000 test examples
- The TSVM gives better performance on all classes

Talk Outline

- I. Text classification
- II. Transductive inference
- III. TSVMs for text classification
- IV. TSVM algorithm
- V. Experimental results
- VI. **Conclusions and future work**

Conclusions and Future Work

TSVMs combine powerful tools

- use prior knowledge about the test set
- exploit co-occurrence properties of text
- use separating hyperplane margin (SVM)
- ✓ TSVMs are well-motivated for text classification
- ✓ Improved performance verified experimentally using three challenging datasets

Open questions

- ? type of concepts that benefit best from transductive learning
- ? better way to represent text and documents
- ? further exploration of better training algorithms
- ? extend transductive classifiers to be inductive classifiers

Questions?