Domain Kernels for Word Sense Disambiguation

Alfio Gliozzo, Claudio Giuliano and Carlo Strapparava, ACL 2005

Discussed by: Mahesh Joshi
University of Minnesota, Duluth
joshi031@d.umn.edu

14th October 2005

Overview

- Background about Kernel Methods
- Domain Models
- Kernel Methods for WSD
- Results and Discussion

Support Vector Machines

- Machine Learning algorithms based on the principle of Structural Risk Minimization from Statistical Learning Theory
- Maximal Margin classifiers

The Classic SVM Diagram

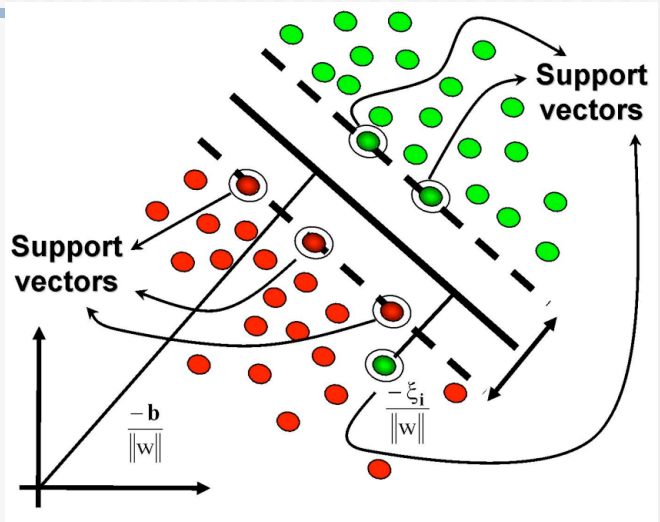

Image Source: http://www.cac.science.ru.nl/people/ustun/SVM.JPG

Basic SVM Equation

- Hyperplane equation is $<\mathbf{w},\mathbf{x}> + b = 0$
- If the input space is N dimensional, i.e. there are N features, then the hyperplane is N-1 dimensional
- Requires that the data be linearly separable (maybe with just a few mis-classified examples)

Duality

- **w** has dimensions equal to the number of features
- Turns out that **w** can be formulated as a linear combination of the training examples
  - $\mathbf{w} = \sum \alpha_i \mathbf{x}_i$ for i = 1 to *k* where *k* is the number of examples

Duality

- This fact, put together with the original equation for SVM, yields the dual form:

$$\sum_{i=1}^{l} \alpha_i < x_i, x > + b$$

- Notice the Dot Product or the Inner Product

Duality

- Now the algorithm needs to find the *dual variables* α_i
- In this case i ranges from 1 through *k* where *k* is the number of traning examples
- This can help efficiency if $k < N$, i.e. if the number of examples is less than the number of features

Why the Hallelujah About Kernels?

- Many problems are not linearly separable
- So linear hyperplanes cannot help, since some non-linear surface separates the data

Kernels at Work

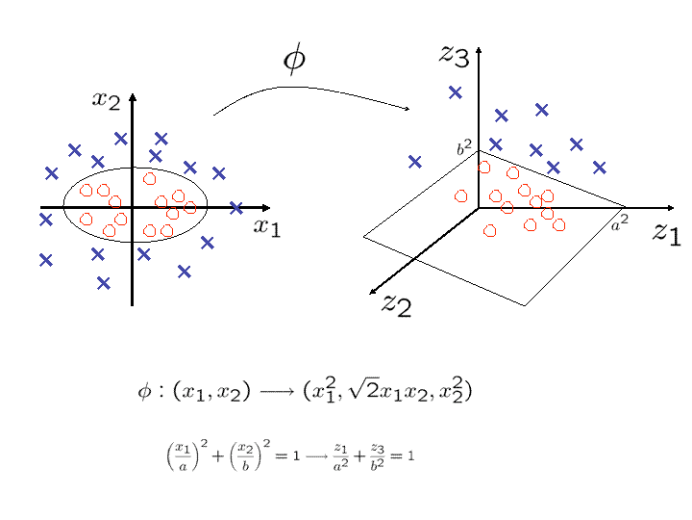

Image Source: http://omega.albany.edu:8008/machine-learning-dir/notes-dir/ker1/phiplot.gif

Input Space to Feature Space

- Need a function (ϕ) that operates upon the input space and yields a *feature space*
- Should to do this in such a way that the data which was not linearly separable in input space, becomes linearly separable in the feature space

Dual formulation in Feature Space

$$\sum_{i=1}^{l} \alpha_i < \phi(x_i), \phi(x) > + b$$

Increase in the dimensionality due to transformation will not affect efficiency of learning, assuming constant time for transformation.

Kernels

- Instead of

$$\sum_{i=1}^{l} \alpha_i < \phi(x_i), \phi(x) > + b$$

- Make use of

$$\sum_{i=1}^{l} \alpha_i \, k(x_i, x) + b$$

Kernels

- *k* is the *kernel function* which gives you the *inner product* of the input examples *in feature space*
- For *k* to be a *valid* kernel function, it needs to satisfy certain conditions
- In very simplistic terms, it should be such a function that gives you a *valid inner product* of the examples when they are transformed into the feature space

Kernel Computation

- With respect to the transformation ϕ on slide 10:

$$
\begin{aligned}
< \phi(x), \phi(z) > &= < (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) > \\
&= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\
&= (x_1z_1 + x_2z_2)^2 = < x, z >^2
\end{aligned}
$$

What does a Kernel Function Evaluate?

- Since the output of the kernel function is an inner product, it evaluates the *similarity* between 2 examples
- Put another way, the kernel is a measure of the distance between 2 examples
- So formulating a new kernel involves finding a meaningful way to represent distance among the training examples, subject to certain validity conditions

Domain Kernels for WSD

- Addresses the knowledge acquisition bottleneck
- A form of semi-supervised learning

Key Ideas

- Use of unlabeled corpora for acquiring *external* knowledge - specifically domain knowledge (domain is used in a more general sense)
- Reducing the required amount of training data (for obtaining a given accuracy) by means of above augmentation

Vector Space Model

- Classical *term-by-document* matrix representation, words along rows and documents along columns
- Cell values are frequencies of the *i th* word in the *j th* document
- A given document represented by the corresponding column vector
- Similarity among documents found using cosine measure

VSM Drawbacks

- Lack of understanding *variability* - two documents may be similar even though they do not share any *lexical terms*
- Lack of resolving *ambiguity* - two documents might appear similar due to an ambiguous terms in both of them

Domain Models

- Enhancement over Vector Space Model
- Represents the domain relevance of the terms / words, by making use of a domain matrix D
- Words along rows and *domains* along columns
- *Again*, think of a domain as an abstract group of related concepts and terms

Advantages of Domain Matrix

- Capture variability in columns, the same domain or similar concept can have multiple words associated with it
- Capture ambiguity in rows, the same word can belong to multiple domains

Transformation

- Transform document vectors from VSM to Domain Model
- Make use of Inverse Document Frequency for each term, in addition to the domain matrix D

Learning a Domain Matrix

- Using Latent Semantic Analysis
- The reduced number of dimensions represent the different *domains*
- This domain matrix is plugged into the Equation (1) from the paper
- Similarity is then calculated among these *domain vectors*

Kernel Methods

- The WSD kernel is a combination of several other kernels
  - Bag-of-Words kernel
  - Part-of-Speech kernel
  - Collocation kernel
  - Domain kernel

Domain Kernels

- Aim to use the domain similarity among the context of ambiguous words
- Require a domain matrix which is learned from unlabeled data
- Bab-of-Words kernel is a special case of the domain kernel

Syntagmatic Kernels

- Dealing with *sub-sequences* of strings
- e.g. *n-grams, part of speech tags*
- The strings are the contexts
- Sub-sequence length can vary

Salient Results

- The combination containing domain kernels yielded best results
- Reduced the required training data to approximately 50%
- In case of English and Spanish tasks, results exceeded human annotator agreement

References

- A. Gliozzo, C. Giuliano, C. Strapparava: Domain Kernels for Word Sense Disambiguation. *Proceedings of the 43rd Annual Meeting of the ACL*, pages 403--410, Ann Arbor, June 2005.
- J. Shawe-Taylor and N. Cristianini. 2004. Kernel Methods for Pattern Analysis. Cambridge University Press.