Obtaining calibrated probability estimates from support vector classifiers: project proposal

Joseph Drish
Department of Computer Science and Engineering 0114
University of California, San Diego
La Jolla, California 92093-0114
jdrish@cs.ucsd.edu

1 Description

In many supervised learning tasks a learned classifier automatically induces a ranking of test examples, making it possible to determine the relative likelihood that a given test example belongs to a certain class. However, for many applications this ranking is not sufficient, particularly when the classification decision is cost-sensitive. In this case, it is necessary to convert the outputs of the classifier into well-calibrated posterior probabilities. A recent paper that addresses this problem is [7], which introduces new methods for estimating the probabilities from naive Bayes and decision tree classifiers. The goal of this project is to replicate that work using Support Vector Machines (SVMs).

Based on the theory of Structural Risk Minimization [5], SVMs learn a decision boundary between two classes by mapping the training examples onto a higher dimensional space and then determining the optimal separating hyperplane between that space. Given a test example x, the SVM outputs a score that provides the distance of x from the separating hyperplane. The sign of the score indicates to which class j example x belongs, where $j \in \{1, -1\}$. The problem of interest is how to calibrate that score into an accurate class conditional posterior probability, or $P(j|x)$.

Our initial solution is to use a histogram technique known as binning, which is recommended in [7] for naive Bayes classifiers. We selected this method because of the similarities between naive Bayes and support vector classifiers, and also because of its simplicity. The binning method proceeds by first ranking the training examples according to their scores, then dividing them into b subsets of equal size, called bins. The value of b is chosen experimentally such that the variance is reduced in the binned probability estimates. Given a test example x, it is placed in the bin according to the score produced by the SVM. The corresponding estimated probability $P(j|x)$ is the fraction of training examples that actually belong to the class that has been predicted for the test example.

The dataset that we will use to train and test the support vector classifiers is from the KDD'98 data mining competition. This dataset contains information about persons in the past who either did or did not make donations to a certain charity. It consists of 95,412 training examples, each corresponding to an individual, and 481 features. The test set consists of 96,367 examples and 479 features. The training set has two additional fields: one indicating whether or not the individual has donated, and another for the amount of the donation. The goal is to choose individuals to solicit a donation so that overall profit is maximized, assuming that the cost to mail a solicitation is $0.68.

A stress test that examines the behavior of SVMs using the KDD'98 dataset is an ancillary benefit of this project. It is not known whether support vector classifiers are able to scale to a dataset of this size. Therefore, it is extremely important to evaluate and choose SVM software with care. We evaluate the software using a number of criteria, including its efficiency, robustness, ease of use, adaptability, and documentation quality. There should also be evidence that the software is updated on a frequent basis.

The quality of the SVM probability estimates will be evaluated using the 4 metrics suggested in [7]. These are squared error, log-loss or cross entropy, lift charts, and the profit obtained when we use the estimates to choose individuals to solicit a donation. The last metric requires knowledge of expected donation amount for an individual that is predicted to make a donation. Since this project is only concerned with calibrating accurate probabilities, these amounts are fixed and based on a regression method described in [6]. We evaluate the success of SVM probability estimates by comparing our results with those obtained in [7] for naive Bayes and decision tree classifiers.

2 Timeline

The first step of the project is to evaluate available SVM software. The author has already elected to use a C or C++ implementation, since code written in Java or Matlab would be too slow for this application. The two packages currently under consideration are SVM^{light} [1] by Thorsten Joachims of the University of Dortmund, and LIBSVM by Chih-Chung Chang and Chih-Jen Lin of National Taiwan University. Both of these implementations are written in C.

SVM^{light} is fairly well documented, and it uses a modified version of Sequential Minimal Optimization (SMO) [3] for training, which is the most efficient algorithm to train SVMs currently known. An advantage of SVM^{light} is that it is used by many SVM researchers, and Joachims is accessible by email to provide installation and usage support. Although it is easy to use, SVM^{light} is difficult to modify. This is a hindrance since we expect to augment the code with various methods for probability estimation. The latest version of SVM^{light}, 3.50, was released on September 11, 2000.

Version 2.31 of LIBSVM was released on April 12, 2001. The software is accompanied by a detailed description of the implementation used for training, which appears to be very good. It also uses a modified version of SMO, and is known not to contain too many bugs. One drawback of this package is that many SVM researchers are not knowledgeable of it, leaving only the developers as a means of software support. However, an advantage is that the primary research interests of Chang and Lin is developing software for SVMs, whereas for Joachims SVM software development is only a secondary interest. Another benefit of this software is that it is very well organized, making it easy to modify and use.

Even though the current version of LIBSVM is more recent, there is probably not much difference between the two packages with respect to efficiency. They are also functionally comparable, as both of them have options to set a number of parameters for training support vector classifiers (e.g, polynomial and radial basis kernels). It is probably best to choose the package that is most likely to be updated as new algorithms emerge for training SVMs. We expect to have the software selected, installed, debugged, and tested by Monday, April 30th.

Once the SVM software is ready to use, the next step is to choose the optimal parameter settings for experimentation. A subset of the training data will be used to tune the SVM kernel function parameters, and to determine the optimal value for b, the number of bins. As mentioned in [7], it is not necessary to use separate training sets to learn the naive Bayes classifier and for the binning process. It is unclear if this assumption can be made for SVMs. Experiments to determine all these settings will be completed by Tuesday, May 8th.

We can then begin the official experiments using the binning method to obtain the class conditional probability estimates from the SVMs. We will gather the results and compare them to the results using naive Bayes and decision tree classifiers. We will repeat all experiments to ensure that we have not made any mistakes. This part of the project will be completed by Tuesday, May 15th.

If time permits, we will choose another method for SVM probability estimation motivated by a relevant research paper. Two such candidates are [2] and [4]. The former uses a Bayesian approach to moderate SVM outputs into calibrated probabilities, and the latter is a regression method that uses the SVM classifier to estimate a sigmoid function. The selection of the final experimental method will be made in consultation with the instructor for CSE 254. This last experiment is to be finished by Tuesday, May 22nd.

The draft of the project report will be completed by Wednesday, May 30th, and the final version will be completed and submitted by Monday, June 11th.

References

[1] T. Joachims “Making large-Scale SVM Learning Practical”. *Advances in Kernel Methods - Support Vector Learning*, MIT-Press, 1999.

[2] J. T. Kwok. “Moderating the Outputs of Support Vector Machine Classifiers”. *IEEE-NN*, 1999.

[3] J. Platt. “Fast Training of Support Vector Machines using Sequential Minimal Optimization”. *Advances in Kernel Methods - Support Vector Learning*, 1999.

[4] J. Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. *Advances in Large Margin Classifiers*, 1999.

[5] V. Vapnick. Statistical Learning Theory. John Wiley & Sons, 1998.

[6] B. Zadrozny and C. Elkan. “Learning and making decisions when costs and probabilities are both unknown”. (Technical Report CS2001-0664). University of California, San Diego.

[7] B. Zadrozny and C. Elkan. “Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers”. To appear, *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.