Where's my Ferry?

Support Vector Machines for Modeling Ferry Tardiness

March 24, 2013

Contents

1 Introduction

Ferries (as in **Figure 1** present an interesting opportunity to examine a complex traffic system filled with data and affecting the lives of many individuals. This paper will discuss a new approach to predicting the timeliness of ferries using a support vector machine. An overview of how support vector machines work and their complications in real world use will introduce the discussion of how this powerful construct presents a powerful and intuitive model for tackling the massive data associated with ferries. A functioning model is developed to examine the complexities of this approach, the practicality, and as a means of exploring some of the data.

Figure 1: A prototypical WSDOT ferry on its route.

We will attempt to predict the tardiness of ferries, where tardy is defined as no less than three minutes later than the initial (at day's beginning) scheduled arrival or departure time. This definition of tardiness was chosen to allow for a non-trivial number of tardy events (more than 10restroom in three minutes time or otherwise make use of that information. The Washington State Department of Transportation's ferry system in the Pacific Northwest served as the source for all data. As will be discussed later on, this ferry system is notoriously on time with over 95% of trips being on schedule (within one minute). This system is of particular interest for its sheer volume of passengers: 22 million per year. The volume lends itself well to this project's goals of using data mining.

2 Preliminaries

Since the goal of the project was to use a large data set for machine learning and to develop a model for possibly predict future ferry tardiness, two approaches popular in available implementations and the literature were initially evaluated: artificial neural networks (ANNs) and support vector machines (SVMs). These two approaches fit into the category of **supervised learning**: labeled (known) data is fed into an algorithm that learns to make better predictions through comparing its own predictions to the labeled data set. ANN and SVM are often considered to be very similar in the problems they attempt to tackle, especially since each can be used as a linear classifier: a function taking in some object and determining a class to which it belongs. Rather than considering tardiness as a spectrum of degrees, we approach tardiness as a binary feature: late or not. This conceptually simplifies our problem. If artificial neural networks and support vector machines can both solve classification problems, however, which are we supposed to choose?

2.1 ANN and SVM

Byvatov et al examine the differences of the two approaches for classifying drugs [1]. While their model certainly isn't for a ferry system, their discussion suggests data sets with many features (an aspect of the ferry model we discuss later) are better suited for SVM, and they also find little difference in the predictions of ANN and SVM. Most importantly, their conclusion, and that of their references, is the two approaches are complementary. Each tool catches different cases, but the results are comparable in each approach, and SVM may even require less "tweaking". They concede that SVMs are generally easier to conceptualize, lending a sense of confidence to the work.

Support vector machines are more intuitively motivated. Consider mapping the features of a ferry trip (time of departure, weather, boat name, etc.) into a coordinate plane. Now, if we do this for all the points in our training set, we can also attach labels to the points (late or not, as in **Figure 2**). In two dimensions, we could try drawing a line through the late and not-late points to separate them. This line is simply a hyperplane in higher dimensions, and SVM has its theoretical underpinnings based on the idea of drawing this hyperplane optimally. Apart from the possible ease of use and understanding, there are also examples of SVM being used in similar traffic prediction systems, providing a baseline comparison and the final reason for choosing SVM over artificial neural networks.

Smith et al uses SVM to predict the timeliness of airline traffic in conjunction with weather patterns [3]. Their goal is to predict the likelihood of requiring a "ground delay program": shuffling traffic to account for tardiness in the schedule. The model they use involves traffic flow management programs to estimate the effects of capacity on an air

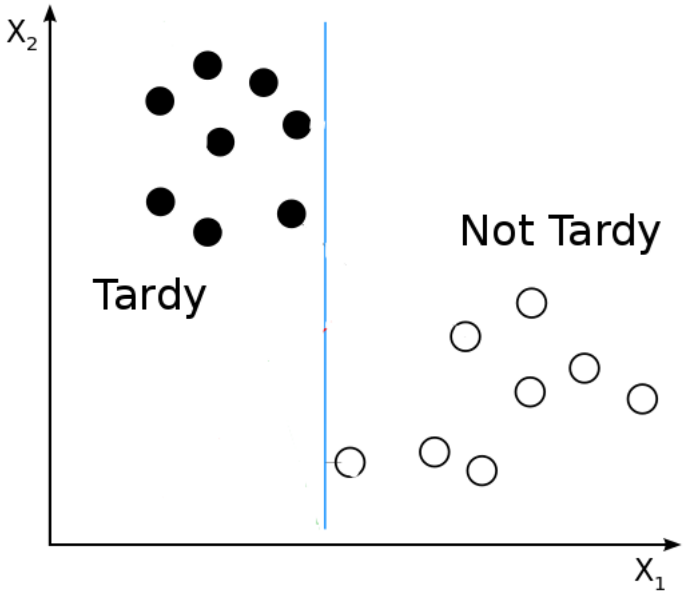

Figure 2: A simple example of a separating line for tardiness (i.e. hyperplane). Each data point (or instance) has features X_1 and X_2.

traffic system, and the SVM is trained on labeled data to produce a function predicting the need of a ground delay program and/or an actual delay. Their system was found to be 78% accurate in predicting need of a ground delay program and 83% accurate in predicting a delay. Air traffic is much like the ferry system given its sensitivity to weather, attempt to adhere to a strict schedule, and heavy use. Due to these similarities, Smith's access to resources like the AMPL supercomputer, and his group's expert knowledge, we use 80% as the target accuracy in for the ferry model.

It is also noteworthy that SVMs can be used for regression analysis in large data sets with many variables [2]. While this paper will not cover this use of SVM, the possibility of performing such an analysis with the same tool was one reason in choosing support vector machines.

2.2 Basics of SVM

A brief overview of the principles behind support vector machines and their use will help to lay a basis for the process of the project. As stated earlier, SVMs are used to find a linear classifier (some function) for a set of data. In this project, we seek a way to separate ferry trips that will be three minutes past their scheduled time (late) or less (on time). To find the linear classifier, an SVM uses a feature vector, F, to describe the trip. An example F may have entries for

$$F = \begin{bmatrix} EstimatedArrival \\ BoatName \\ Temperature \\ WindSpeed \end{bmatrix}.$$

This feature vector needs to be entirely numerical, so categorical variables like boat name must be encoded as numbers. It is possible to assign boat names to different values, but most often, it is best to expand the feature like a bit vector. If there were boat names *SS Minnow, Death Star,* and *Millennium Falcon*, then we would have

$$F = \begin{bmatrix} EstimatedArrival \\ SS\_Minnow \\ Death\_Star \\ Millennium\_Falcon \\ Temperature \\ WindSpeed \end{bmatrix}$$

where the entries for boat names are a 1 if this trip has that name and 0 otherwise. For reasons detailed in [2] this is considered best practice.

Along with feature vectors for every trip, an SVM uses a weight vector, w, that it is the backbone for the linear classifier. Thus, this w is what the SVM trains. To determine which class a feature vector corresponds to the dot product of F and w is taken, with negative values corresponding to late and positive values to on time. While training on the labeled data, however, the SVM adjusts w whenever it predicts incorrectly. The amount to adjust can vary based on the implementation, but over time w improves on its ability to predict the class of any instance in the training set.

An optimal solution to w would form a hyperplane separating the instances perfectly by class. In **Figure 3** is an example of how an SVM might train a weight vector (represented graphically as a line) over a set of training data. The final vector, H_3, is considered optimal because it maximizes the distance between itself and the nearest points of each class. In the example, it is possible to separate the data with a line, but in the real world, and our ferry problem, this is rarely the case.

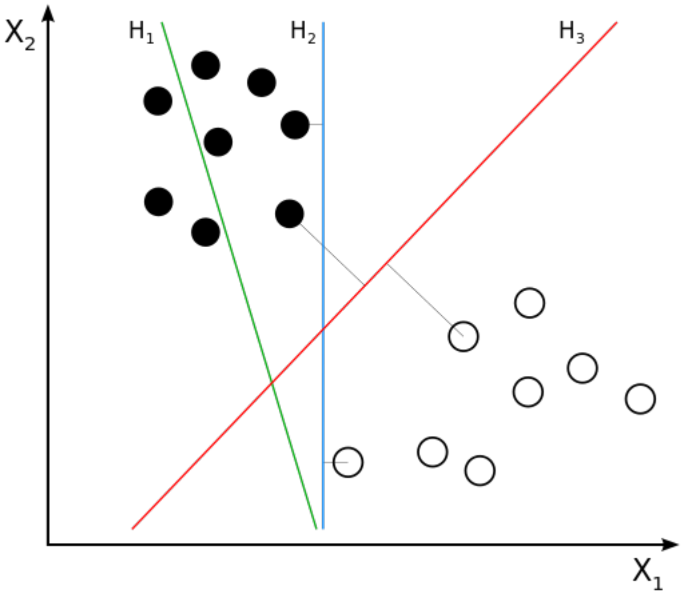

Figure 3: As in **Figure 2**, we are trying to separate labeled points with a line. An SVM does this iteratively, meaning line H_1 would be the first pass, H_2 the result of refitting, and H_3 the final classifier produced by the SVM.

2.3 Kernel mapping

2.4 LibSVM

3 The problem with ferries

3.1 Washington State's ferries

3.2 Where the data comes from

3.3 The first pass

3.4 General results

3.5 The problem with weather

4 Closing thoughts

4.1 Remaining questions

4.2 Extensions

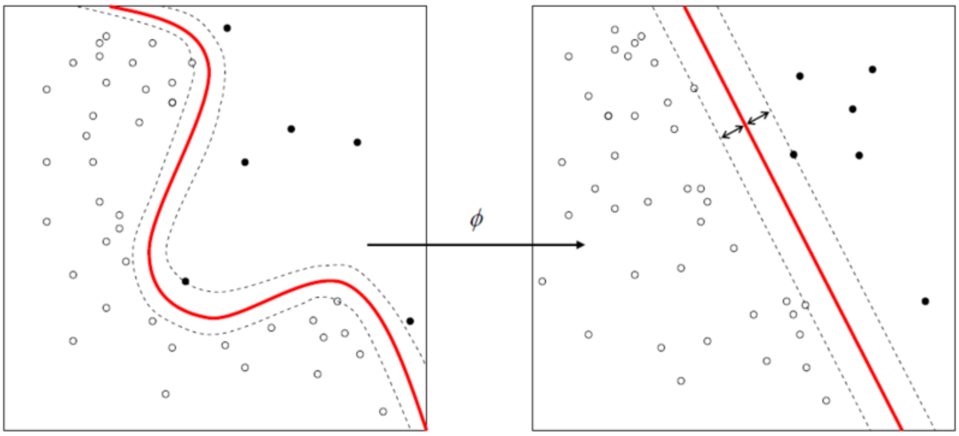

Figure 4: Separating the data in the first part requires a mapping into a higher (often much) dimension. This mapping makes it possible for a learning algorithm like SVM to find at least some separating hyperplane.

References

[1] Evgeny Byvatov, Uli Fechner, Jens Sadowski, and Gisbert Schneider, *Comparison of support vector machine and artificial neural network systems for drug/nondrug classification*, Journal of Chemical Information and Computer Sciences **43** (2003), no. 6, 1882–1889.

[2] Chih-Chung Chang and Chih-Jen Lin, *Libsvm: a library for support vector machines*, ACM Transactions on Intelligent Systems and Technology (TIST) **2** (2011), no. 3, 27.

[3] David A Smith, Lance Sherry, and G Donohue, *Decision support tool for predicting aircraft arrival rates, ground delay programs, and airport delays from weather forcasts*, International Conference on Research in Air Transportation (ICRAT-2008), 2008.