The Effects of Outliers on Support Vector Machines

Josh Hoak
jrhoak@gmail.com

Portland State University

Abstract. Many techniques have been developed for mitigating the effects of outliers on the results of SVM classification, including fuzzy-SVMs and weighted-SVMs. This paper examines the effect that outliers have on weighted-SVMs and standard SVMs on artificially generated data, using linear, radial basis function, and polynomial kernels. We find that SVMs are robust in the presence of noise – outliers caused by mislabeled data.

1 Introduction

In the field of data classification, data points are detected as outliers either because they are the points of interest (known as anomaly detection) or because they are a hinderance to the classification. This paper considers outliers of the latter category and how they affect support vector machines or (SVM), a machine learning model developed by Vapnik and others [8, 10]. SVMs have been quite successful in both their training time and their accuracy, and so are of particular interest to researchers doing practical pattern classification [1].

SVMs are not impervious to outliers [8], and so methods have been developed to mitigate the effects of outliers on SVMs. Some researchers have used schemes that identify possible outliers, assigning a confidence value that indicates how likely a point is believed to be an outlier. In implementing this, researchers have developed the ideas of the "fuzzy"-SVM and weighted-SVMs [4–7, 11]. However, some of these studies show only incremental improvements over standard SVM methods [5].

These studies raise some fundamental questions:

- To what degree do outliers affect SVM models?
- By how much can we improve our SVM models in ideal and real-world situations?
- What methods should we use to deal with outliers?

These are difficult questions to answer in general. SVMs have many different tunable parameters, and so it is unclear what parameters represent the 'most general' parameters. Also, to test the effects of outliers, we must create outliers according to some rule. However, a common the characteristic of outliers is that they are unpredictable, and so results may be particular to the generating scheme. Deciding which data to use is difficult as well. We can choose to modify data already in existence, or we can make artificial data, but in this latter case, we have to design ‘reasonable’ data, and it is unclear what this should look like. Structurally, SVMs are only designed to classify between two classes and so the results of a multi-class SVM might reflect our choice of ensemble classifier rather than the effects of the outliers on SVMs.

This paper takes some first steps at examining these questions by looking at a particular type of outlier called *noise*, in which the class has been incorrectly labeled. For data, we created artificial data using the statistical package *R* and we examined some standard data sets [3]. We created noise on the training set by flipping the class label from 0 to 1 or 1 to 0. Then, we observed how the introduction of noise affected the accuracy on the test set. Finally, we compared the performance of a standard SVM with a weighted-SVM, both of which are provided in the LIBSVM library [2].

2 Support Vector Machines

Support vector machines are a standard classification technique used in machine learning and are reviewed here briefly. For a longer exposition of the topic, see [8, 9]. We define the *training data* S to be a set consisting of tuples of feature vectors $\mathbf{x}_i \in \mathbb{R}^N$, along with a given label $y_i \in \{-1, 1\}$ called the *class*. In symbols,

$$S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_m, y_m)\}.$$

The task of the SVM is to find the optimal hyperplane separating the a set of training points into two categories. Practically, many problems are not linearly separable, and so features are mapped to a higher dimensional space – that is, we often map $\mathbf{z} = \phi(\mathbf{x})$, where $\phi : \mathbb{R}^N \to \mathbb{F}$ and where $\mathbb{F}$ is some feature space. Then, we are looking to find the optimal hyperplane such that

$$\mathbf{w} \cdot \mathbf{z} + b = 0,$$

with

$$y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \qquad \text{for } i = 1, \dots, m$$

By symmetry, we require that $\xi_i \geq 0$. Finding the optimal hyperplane is equivalent to minimizing

$$\frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i$$

subject to

$$y_i(\mathbf{w} \cdot \mathbf{z}_i + b) \geq 1 - \xi_i, \qquad \text{and } \xi_i \geq 0 \qquad \text{for } i = 1, \dots, m.$$

To solve the optimal hyperplane problem, we can construct a Lagrangian and transforming to the dual, defining the kernel $K(\cdot, \cdot)$ as the function where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(x_i) \cdot \phi(x_j)$. Then, we can equivalently maximize

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to

$$\sum_{i=1}^{m} y_i \alpha_i = 0, \qquad \text{and } 0 \le \alpha_i \le C, \qquad \text{for } i = 1, \ldots, m.$$

For a test example $\mathbf{x}$, we define the decision function D for some hyperplane H (established by training) as

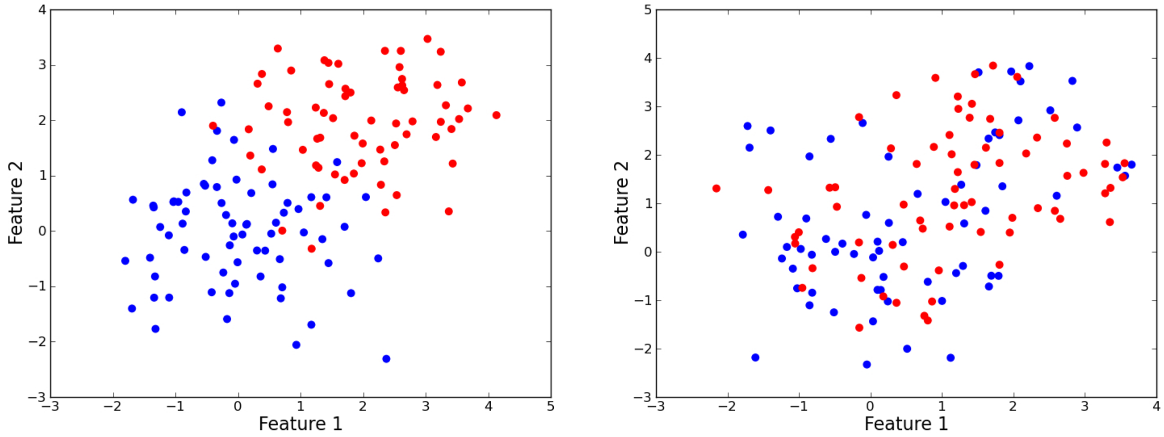
$$D_H(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right).$$

3 Methods

We generated the artificial data using the statistical package R. One hundred instances of data were created for each class $c \in \{1, 2\}$, with the features of each class normally distributed around $(2 \cdot (c - 1))$ with standard deviation 1 (see Figure 1). This means that for the artificial data used, the center of class 1 would be located at $(0, 0)$ while the center of class 2 would be located at $(2, 2)$. For the real-world data, we used the Breast Cancer, Heart, and Ionsphere data from UCI, Statlog, and UCI respectively [3].

We selected a fraction of the initial data set to be used for training and testing, and then we created outliers on the training set. For our experiments, 70% of the data went to the training set and 30% went to the testing set. Outliers were created by selecting a point and flipping the class for a fraction, between 0 and 1/2, of the training examples.

To perform the SVM training and testing, we used LIBSVM, with linear, polynomial (cubic) and radial-basis kernels [2]. To reduce variation for each noise-setting used, the SVM was run on multiple different sets of data and then accuracy over these sets was averaged. The performance of the standard SVM was then compared to weighted versions, we used LIBSVM's weighted tool, which allows one to give different weights for each training instance. We only did a simple weighting scheme in which we gave every non-outlier training example 1 and every outlier training example a fixed value (set between 0 and 1). A value of 0.0 for the weight indicates that the outlier is excluded from training.



(a) Initial Data Creation.

(b) Training Data with 30% Noise

Fig. 1. Data Creation. Data was generated in **R** using normal distributions centered around (0,0) and (2,2), each with standard deviation 1.

4 Results and Discussion

For the artificial data, both the linear and radial-basis kernels performed quite well in the presence of noise (see Figure 2). For these kernels, accuracy on the test sets was unaffected until outliers reached roughly 20% of the training examples. However, the polynomial kernel (over degree 3), showed worse performance: even with setting 5% of the training data set to outliers, we observed better performance in the weighted-SVM. The sigmoid kernel was also tested, but its performance was quite poor – the maximum accuracy was only around 60%.

For the real-world data (Figures 3-5), there was no perceivable pattern to which kernels did better. The breast cancer data performed very well in the presence of noise: for both the RBF and polynomial kernels, noise had no affect on the performance of a standard SVM until thirty percent of the training examples chosen as noise. Standard SVMs trained with the ionosphere and heart data were less resilient to noise. Still, noise in the training data only had a significant affect on the testing accuracies for these data sets when the percentage of noise examples in the training data rose above 10 percent.

From both the artificial and real-world data, this study indicates that significant results are difficult to attain using outlier-mitigating efforts such as fuzzy-SVMs over the standard SVMs. Our experiments indicate that one must have a high level of noise in a training set ($\geq$ 10%) before test-accuracies are significantly affected. If noise is present, we found that noisy examples do not have to be weighted severely to achieve significant improvements: setting the weights to 0.6 for the noise examples produced much higher accuracies on high noise settings.

Generally, this gives evidence that SVMs are more resilient to outliers than is sometimes supposed.

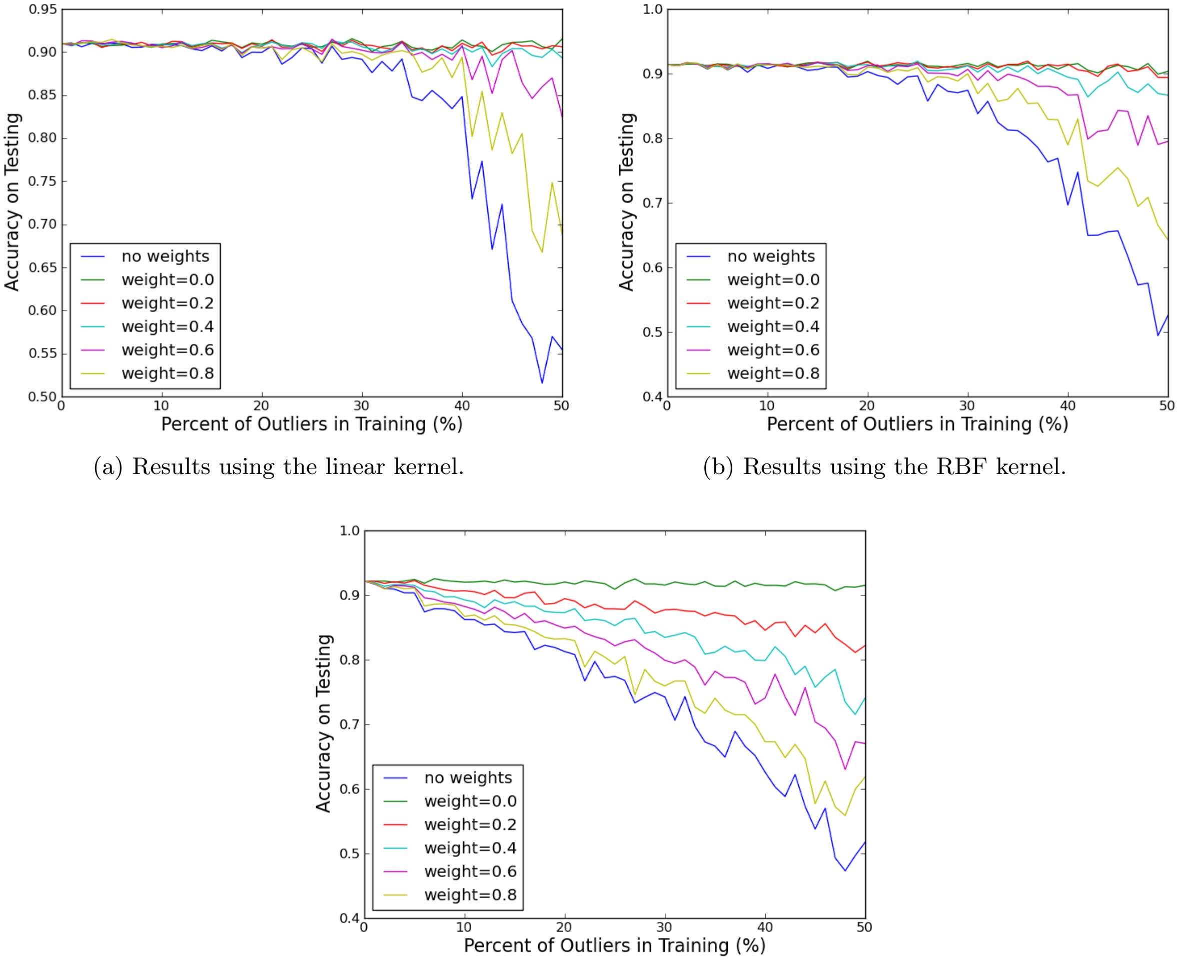
5 Future Work

There are still several questions and areas for further research on this topic. We only looked a the binary-class SVM; multi-class SVMs are more complicated and it would be valuable to see how they respond to noise and outliers. We only tested a very limited set of parameters and types of data. A more exhaustive search would give more convincing evidence that SVMs with particular kernels are more or less susceptible to outliers.

To support the hypothesis that SVMs are relatively robust to outliers, better methods need to be developed to generate outliers. Certainly, incorrectly labeled classes are one particular type of outlier, but probably the most common type of outlier is one in which a particular feature has unexpected or poorly sampled data. Also, we labeled all of our training examples as noise. It is quite possible that an outlier detection scheme would only be able to detect a fraction of the outliers, and so it might be valuable to study the effects on accuracy when only a fraction of outliers are weighted.

References

1. Bartlett, P., and Shawe-Taylor, J. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel Methods: Support Vector Learning* (1998), 43–54.
2. Chang, C.-C., and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
3. Frank, A., and Asuncion, A. UCI machine learning repository, 2010.
4. Lin, C.-f., and Wang, S.-d. Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters 25*, 14 (2004), 1647 – 1656.
5. Line, C.-F., and Wang, S.-D. Fuzzy support vector machines. *IEEE Transactions on Neural Networks 13*, 2 (March 2002).
6. Suykens, J. A. K., Brabanter, J. D., Lukas, L., and Vandewalle, J. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing 48*, 1-4 (2002), 85 – 105.
7. Tsujinishi, D., and Abe, S. Fuzzy least squares support vector machines for multiclass problems. *Neural Networks 16*, 5-6 (2003), 785 – 792. Advances in Neural Networks Research: IJCNN '03.
8. Vapnik, V. *The Nature of Statistical Learning Theory*, 2nd ed. Springer, November 1995.
9. Vapnik, V. *Statistical Learning Theory.* Wiley, 1998.
10. Vapnik, V., and Cortes, C. Support-vector networks. *Machine Learning 20*, 3 (September 1995).
11. Wang, T.-Y., and Chiang, H.-M. Fuzzy support vector machine for multi-class text categorization. *Inf. Process. Manage. 43*, 4 (2007), 914–929.

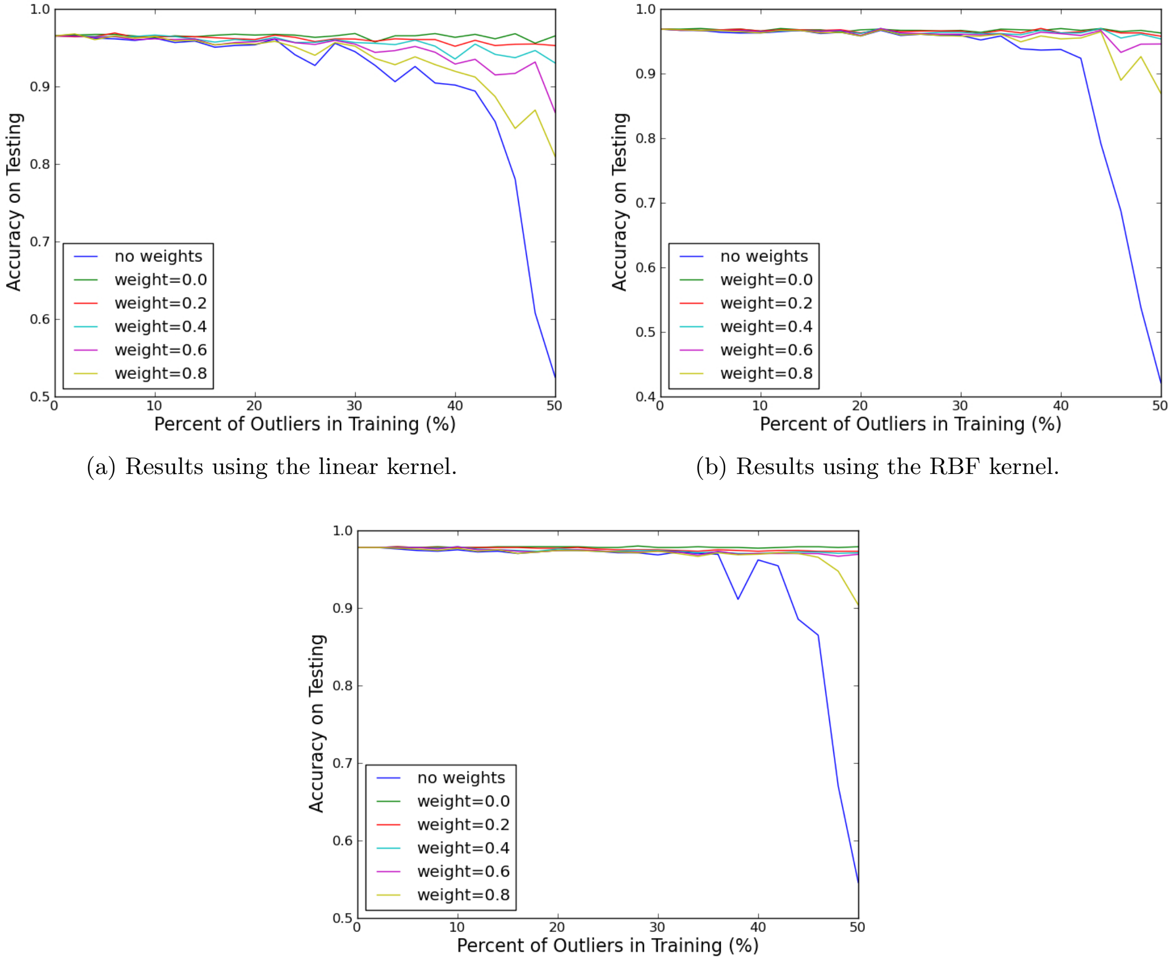


(a) Results using the linear kernel.

(b) Results using the RBF kernel.

(c) Results using the polynomial kernel with degree 3.

Fig. 2. **Results - Artificial Data**. Displayed are the results from both the weighted-SVM and standard SVM using linear, radial basis kernels, and polynomial kernels. The SVMs were run on 20 different data sets for each noise-setting (51 different settings, i.e., 0%, 1%, . . . , 50% of training) and then the accuracies on the test sets were averaged.

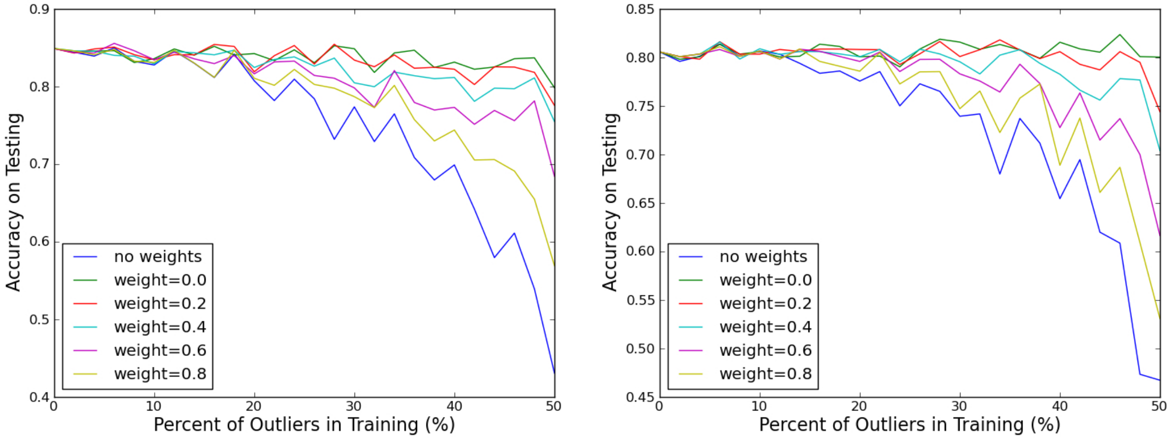


(a) Results using the linear kernel.

(b) Results using the RBF kernel.

(c) Results using the polynomial kernel with degree 3.

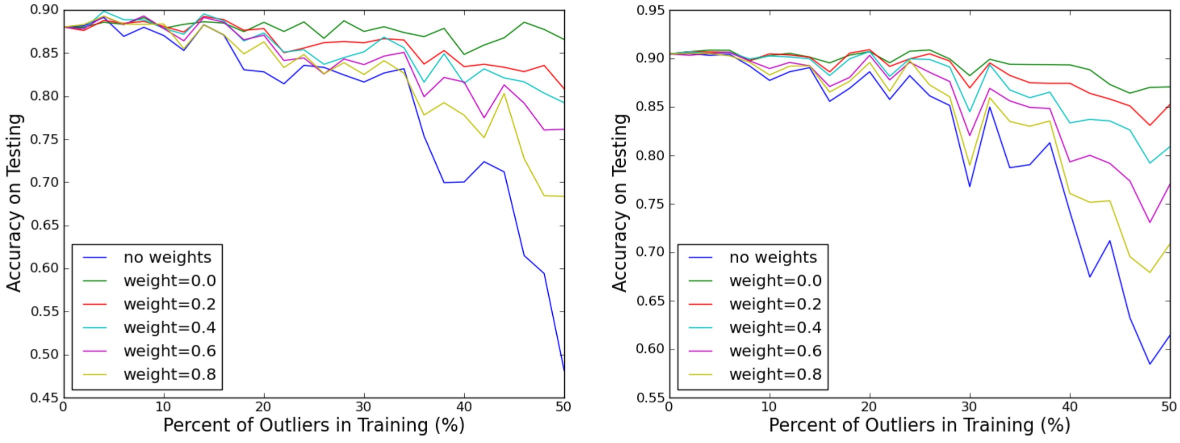
Fig. 3. **Results - UCI Breast Cancer Data**. Displayed are the results from both the weighted-SVM and standard SVM using linear, radial basis kernels, and polynomial kernels. The SVMs were run on 5 different data sets for each noise-setting (21 different settings) and then the accuracies on the test sets were averaged.



(a) Results using the linear kernel.

(b) Results using the RBF kernel.

Fig. 4. Results - Statlog Heart Data. Displayed are the results from both a weighted-SVM and a standard SVM using linear, radial basis kernels, and polynomial kernels. The SVMs were run on 5 different data sets for each noise-setting (21 different settings) and then the accuracies on the test sets were averaged.



(a) Results using the linear kernel.

(b) Results using the RBF kernel.

Fig. 5. Results - UCI Ionosphere Data. Displayed are the results from both a weighted-SVM and a standard SVM using linear, radial basis kernels, and polynomial kernels. The SVMs were run on 5 different data sets for each noise-setting (21 different percentages of the training) and then the accuracies on the test sets were averaged.