Support Vector Machine Failure in Imbalanced Datasets

I. A. Illan(✉), J. M. Gorriz, J. Ramirez, F. J. Martinez-Murcia,
D. Castillo-Barnes, F. Segovia, and D. Salas-Gonzalez

Departamento de Teoria de la señal y Comunicaciones,
Universidad de Granada, Granada, Spain
illan@ugr.es

Abstract. Imbalanced datasets often pose challenges in classification problems. In this work we study and quantify the problem of imbalanced classification using support vector machines (SVM). We identify the conditions under which a SVM failure occur, both theoretically and experimentally, and show that it can be relevant even in cases of very weakly imbalanced data. The guidelines for exploratory data analysis are presented to avoid the SVM failure.

Keywords: Support vector machines · Imbalanced data · SVM · Data analysis · SVM failure

1 Introduction

Often in statistical learning, the available training data set has few samples in a high dimensional space, allowing very poor estimations on probability distribution functions. Non-parametric approaches based on statistical learning theory, such as neural networks or SVMs, have been proven to be very succesful solving classification problems. However, in the case of unbalanced training datasets, some difficulties arise if the learning algorithms are straightforwardly applied. For example, the soft margin solution in SVM [4,9] for non-separable classes, includes a term in the lagrangian that accounts for the classification error rate together with the structural risk minimization. If the learning algorithm is optimized to minimize the risk of misclassifying samples, some additional constraints must be imposed to avoid the trivial solution in imbalanced datasets. The trivial solution is achieved when all samples are classified as the dominant class. In that undesirable case, the misclassification error can be very small if the dominant class outnumbers the scarce class in several orders of magnitude, thus masking the problem. It is however possible that the trivial solution is achieved in cases of weakly imbalanced data.

A common practice in SVM imbalanced classification is to apply penalties to the classification errors on the scarce class, so that the risk of classification errors is weighted. Usually, no other method or theoretical ground for class weight estimation but trial-and-error is proposed. Moreover, it has been shown that

J. M. Ferrández Vicente et al. (Eds.): IWINAC 2019, LNCS 11486, pp. 412–419, 2019.
https://doi.org/10.1007/978-3-030-19591-5_42

applying class weights is equivalent to use fuzzy-SVMs [6]. The importance of the data properties has been studied in imbalanced data [7], although the use of weights in SVM is usually reserved to heavily imbalanced data.

In this work we study the relevant properties of the data for SVM imbalanced classification and its theoretical relation to the trivial solution.

2 Methods

The methodology followed in this study is as following: first a review on SVM is given, fixing the notation. Secondly, a definition of SVM failure is given, together with a theoretical derivation of the conditions that cause it. Lastly, a experimental set is proposed to illustrate the effects of the SVM failure and the circumstances around it.

2.1 Support Vector Machines

SVM is a machine learning algorithm that separates a given set of binary labeled training data with a hyper-plane that is maximally distant from the two classes (known as the maximal margin hyper-plane). In the C-SVM formulation, the problem of finding the maximal margin hyperplane is solved by quadratic programming algorithms that try to minimize the dual of the cost function J:

$$J(\boldsymbol{w}, w_0, \xi) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{l}\xi_i, \tag{1}$$

subject to the inequatity constraints:

$$y_i[\boldsymbol{w} \cdot \mathbf{x}_i + w_0] \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad i = 1, 2, ..., l. \tag{2}$$

where the slack variables ξ_i make the margin "soft", by incorporating to the optimization those feature vectors that are not separable, leading to the soft margin solution (details can be found in [10] and [9]).

By applying Lagrange duality and introducing kernel methods, the following dual optimization problem is obtained:

$$\min_{\alpha} \frac{1}{2}\sum_{j=1}^{l}\sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l}\alpha_i \tag{3}$$

subject to the KKT dual conditions:

$$\sum_{i=1}^{l} y_i \alpha_i = 0, \quad \text{and} \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., l \tag{4}$$

where $K(.,.)$ is the kernel function and α_i are the Lagrange multipliers that need to be solved. The dual conditions will be related to the primal problem as:

$$\alpha_i = 0 \quad \rightarrow \quad y_i[\mathbf{w} \cdot \mathbf{x}_i + w_0] \geq 1 \tag{5}$$

$$\alpha_i = C \quad \rightarrow \quad y_i[\mathbf{w} \cdot \mathbf{x}_i + w_0] \leq 1 \tag{6}$$

$$0 < \alpha_i < C \quad \rightarrow \quad y_i[\mathbf{w} \cdot \mathbf{x}_i + w_0] = 1 \tag{7}$$

Common kernels that are used by SVM practitioners for the nonlinear feature mapping are:

- Polynomial

$$K(\boldsymbol{x}, \boldsymbol{y}) = [\gamma(\boldsymbol{x} \cdot \boldsymbol{y}) + c]^d. \tag{8}$$

- Radial basis function (RBF)

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{y}||^2). \tag{9}$$

as well as the linear kernel.

The solution to that problem can be expressed by a linear combination of a subset of vectors, called support vectors:

$$d(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + w_0 \tag{10}$$

where $\mathbf{s}_i$ are the N_S support vectors; those vectors with $\alpha_i > 0$. Taking the sign of the function $d(\mathbf{x})$ leads to the binary classification solution [10]. The solution may also be expressed as:

$$y(\mathbf{x}) = \text{sign}(\varphi(\mathbf{x}) \cdot \mathbf{w} + w_0) \tag{11}$$

where:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \varphi(\mathbf{x}_i) \tag{12}$$

with $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ being the kernel mapping that will be the identity in the linear case.

3 SVM Failure

The solution given in 13 can be split into its positive and negative class fractions as:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i^+ \varphi(\mathbf{x}_i^+) - \sum_{i=1}^{l} \alpha_i^- \varphi(\mathbf{x}_i^-) \tag{13}$$

where $\mathbf{x}_i^-$ and $\mathbf{x}_i^+$ are the negative and positive training examples. The sign of the vector $\mathbf{w}$ will determine how the positive and negative labels are assigned in reference to the hyperplane. Ideally, the vector $\mathbf{w}$ will point *from* the negative class *to* the positive class. However, there will be some special cases where the vector $\mathbf{w}$ will point in the wrong direction, that is, from the positive class towards the negative class. In those cases, the training of the SVM will fail, and the only possible adjustment is to set w_0 so that all the training examples are classified as positive (or negative). We will call that situation *SVM failure.* There are several properties of the training data involved in a SVM failure, namely: the proportion between training samples and the overlap between them. We will show here how this undesirable situation occurs when the difference in support vector density between classes reaches a threshold inside the margin.

Let us first consider the simpler case of linear SVM. In accordance with the dual KKT conditions, $\alpha_i = C$ for all the training examples inside the margin, including those in the wrong side of the margin. The constraints imposed in Eq. 4 to the solution of Eq. 10 make it possible to express the vector $\mathbf{w}$ as:

$$\mathbf{w} = C(\bar{\mathbf{x}}_s^+ - \bar{\mathbf{x}}_s^-) \tag{14}$$

where $\bar{\mathbf{x}}_s^+$ and $\bar{\mathbf{x}}_s^+$ are the average positive and negative support vectors respectively, and where we have neglected those support vectors with $0 < \alpha_i < C$ for reasons that will become clear later. Intuitively, $\mathbf{w}$ can be thought as the difference vector between the average positive support vector and the average negative support vector, up to a factor. However, to guarantee the optimal performance of the SVM, the sign of the vector $\mathbf{w}$ must be the same as the sign of the vector $\mathbf{v}$ defined as:

$$\mathbf{v} = \sum_{i=1}^{n^+} \mathbf{x}_i^+ - \sum_{i=1}^{n^-} \mathbf{x}_i^- \tag{15}$$

where n^+ is the total number of samples in the positive class and n^- is the total number of samples in the negative class. In oder words, we expect the classifier to be somewhere between the average positive class and the average negative class, thus dividing both point clouds. To understand when this condition is not met, it is useful to analyze the different scenarios in which the Eq. 14 vanishes. For Eq. 14 to vanish, the classes are required to be non-separable. It is easier to analyze first the simpler case of imbalanced data in which all the samples of the scarce class are support vectors, and then discuss the more general case in which Eq. 14 can vanish but there are a non-negligible number of samples that are not support vectors.

3.1 All Support Vectors

In the case of a complete overlap between classes, it is a consequence of the constraints 2 that all the samples of at least one class must be support vectors. Take the negative class to be the scarce one. In that case, one of the terms of Eq. 14 can be calculated explicitly from the data only, the $\bar{\mathbf{x}}_s^-$ term. Therefore, if it was possible to calculate the $\bar{\mathbf{x}}_s^+$ term, it would be possible to predict if the classifier will fall into a SVM failure. The smallest value of the second term can be achieved only for no support vectors outside the margin, or all $\alpha_i = C$, neglecting smaller values of α_i as mentioned earlier. If we define the subset of one class samples that are in the region of the feature space delimited by the hyperplane located at a distance C of the furthest negative sample in the $\mathbf{v}$ direction as $\mathcal{L}$, then the SVM failure condition is:

$$\sum_{i \in \mathcal{L}} \mathbf{x}_i^{+} = \sum_{i=1}^{n^-} \mathbf{x}_i^{-} \tag{16}$$

4 Experiments

We performed simulations of real case scenarios in which the different properties of the data are varied, as the imbalance proportion between classes or their overlap. To model the data we used multidimensional Gaussian distributions, that allowed us to control the aforementioned characteristics. Two classes w_1 and w_2 were modeled with Gaussian distributions with different mean and covariance, and described by:

$$p(w_1 \,|\, \mathbf{x}) = \frac{1}{2\pi|\varSigma_1|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_1)^T \varSigma_1^{-1}(\mathbf{x} - \mu_1)\right) \tag{17}$$

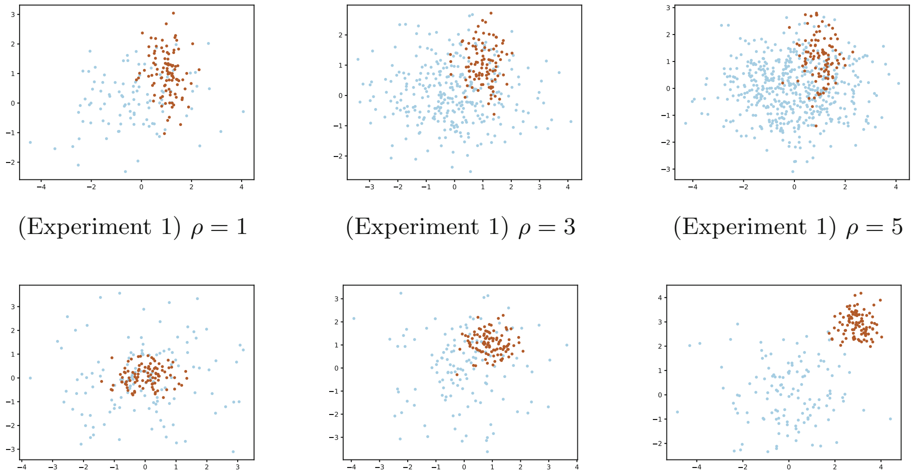
$$p(w_2 \,|\, \mathbf{x}) = \frac{1}{2\pi|\varSigma_2|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_2)^T \varSigma_2^{-1}(\mathbf{x} - \mu_2)\right) \tag{18}$$

Experiment 1: $\mu_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\varSigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ $\varSigma_2 = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.5 \end{pmatrix}$. The training set is built by joining a varying number b of w_1 samples and a fixed number $m = 100$ of w_2 samples. 400 different trained SVM are built by modifying the proportion $\rho = b/m$ ranging form $\rho = 1$ to $\rho = 5$ in 0.01 increments. (see Fig. 1)

Experiment 2: $\varSigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ $\varSigma_2 = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.5 \end{pmatrix}$, and $m = 100$ $b = 125$. The training set is built by varying the value of μ_2 while keeping $\mu_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. 300 different trained SVM are built by modifying the value of μ_2 according to $\mu_2 = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$, with a ranging from 0 to 3 in 0.01 increments. (see Fig. 1)

For each variation of the parameters a and ρ, a SVM was trained, and the accuracy, sensitivity and specificity of each classifier on the trained data was acquired. The results are shown in Fig. 2. To perform the experiments we used the SVC implementation of scikit-learn [8] based on libsvm [3].

Although the data is synthetic, these kind of datasets can represent real data. Such an example can be realized in dynamic-contrast-enchancing magnetic-resonance-imaging (DCE-MRI) for breast cancer diagnosis [5]. Consider a DCE-MRI patient image with N voxels. In such case, the classification problem reduces



(Experiment 1) $\rho = 1$

(Experiment 1) $\rho = 3$

(Experiment 1) $\rho = 5$

(Experiment 2) $a = 0$

(Experiment 2) $a = 1$

(Experiment 2) $a = 3$

Fig. 1. Illustration of simulated data for different parameter settings and different experiments.

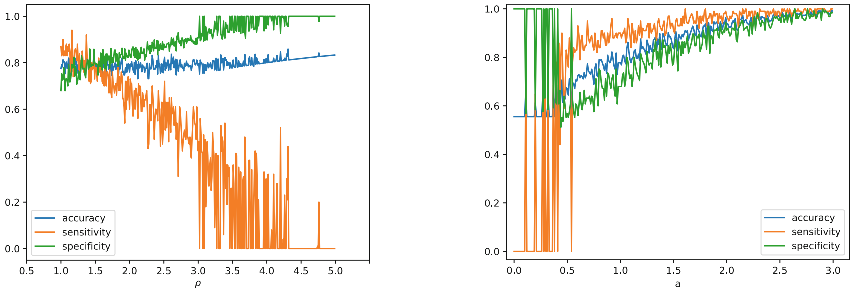
to separate healthy tissues from malignant ones. A set of voxels m belonging to a malignant region constitutes a tumor. It is expected that the number of voxels b belonging to benign regions outnumber the voxels in malignant regions, so that $n/b << 1$ in either the training set or the unseen case. However, the size of the tumor is unknown in unseen cases, and do not have to match necessarily the size of tumors in the training set. Therefore, the proportion n/b is variable as well as its overlapping.

5 Discussion

Results showed in Fig. 2 illustrate the effect of a SVM failure when the conditions are met. For this particular datasets, SVM failure occurs for $\rho \approx 4.5$ in Experiment 1 and when $a \approx 0.4$ in Experiment 2.

For $\rho \approx 4.5$ in Experiment 1 Eq. 16 is fulfilled. By varying the imbalance proportion while keeping the overlap between classes fixed, there is a limit in which there are not enough support vectors in the scarce class to balance the density of support vectors of the dominant class inside the margin, and the sign of $\mathbf{w}$ gets reversed producing the failure. This situation is easier to predict and quantify, since the number of samples in the scarce class establishes a fixed limit.

In the circumstances of Experiment 2, it is harder to predict the SVM failure since it depends on the particular realization of the data, adding to it the limitation to Gaussian distributions of this study. It is however showed that Eq. 14 can vanish in some particular configurations of the data, independently from its imbalanced proportion between classes. It also shows that SVM failure is less



Experiment 1

Experiment 2

Fig. 2. Performance of the SVM on the training set by varying the parameters ρ and a.

consistent, and more random in nature, suggesting that the small changes in the particular realization of the probability distribution function can affect significantly in the SVM failure. This fact can be very relevant in cross-validated studies, where subsampling the data into cross validation folds can produce SVM failure.

A common practice in SVM imbalanced classification is the use of class-weights. This solution is equivalent to the FSVM problem [6] or more precisely, the latter problem is more general and includes the former as an special case. In the light of this interpretation, applying different weights to class-slack variables is the same as decreasing the membership level of one class, say the w_1 class, misclassified samples, while keeping the w_0 membership not fuzzy.

The FSVM-CIL method proposed in [2] explicitly makes use of this correspondence, and propose three different functions to estimate the membership function s_i; distance to class center, distance to estimated hyperplane and distance to actual hyperplane. To tackle the imbalanced problem, they propose to use different upper bound values in the membership function for each class so that the ratio corresponds with the ratio on the priors, in concordance with [1].

However, the solutions to the imbalanced SVM classification problem do not usually study the conditions under which the solutions should be applied. Here we show that even weakly imbalanced data can require of a imbalanced solution as FSVM or weighted-SVM if certain conditions are met, suggesting that balanced classes in SVM are important in guaranteeing its performance.

Here, we have limited the analysis to the simpler linear case, but all the arguments are extrapolable to the non-linear case by including the use of kernels, without modifying any fundamental idea.

6 Conclusion

We have established the theoretical conditions for SVM failure and showed experimentally the circumstances under which it may occur. We have shown that not only the imbalanced proportion between classes is relevant for predicting the

SVM failure, but also their overlap. We have shown that SVM failure can be produced even for weakly imbalanced data, suggesting that balancing or weighting the data is always recommendable as a default option in SVM classification.

References

1. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_7
2. Batuwita, R., Palade, V.: FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Trans. Fuzzy Syst. **18**(3), 558–571 (2010)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 27:1–27:27 (2011). https://doi.org/10.1145/1961189.1961199
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1023/A:1022627411411
5. Illan, I.A., et al.: Automated detection and segmentation of nonmass-enhancing breast tumors with dynamic contrast-enhanced magnetic resonance imaging (2018). https://www.hindawi.com/journals/cmmi/2018/5308517/
6. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Trans. Neural Netw. **13**(2), 464–471 (2002)
7. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. **250**, 113–141 (2013). http://www.sciencedirect.com/science/article/pii/S0020025513005124
8. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
9. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. **12**(5), 1207–1245 (2000). https://doi.org/10.1162/089976600300015565
10. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)