A Joint Clustering and Classification Approach for Healthcare Predictive Analytics *

Wuyang Dai[†], Theodora Brisimi[†], Tingting Xu[†], Taiyao Wang[†],
Venkatesh Saligrama[†] and Ioannis Ch. Paschalidis [‡]

Abstract

We introduce a new method for a classification problem motivated by healthcare predictive analytics. In our setting, the positive class is a mixture of multiple clusters and the negative class is drawn from a single cluster. We employ an alternating optimization approach, which jointly discovers the clusters in the positive class and optimizes the classifiers that separate each positive cluster from the negative samples. The classifiers are Support Vector Machines (SVM) with double, sparsity-inducing regularizations. Classification discovers a different lower-dimensional feature subspace for each cluster and we show that the sample complexity is proportional to the dimension of this subspace. We establish the convergence of the joint clustering/classification process and characterize its VC dimension. We show comparisons to several existing methods. Experimental results on actual medical data demonstrate better prediction accuracy as well as successful cluster detection, thereby, providing a way to interpret the classification results.

1 Introduction

This paper is motivated by the important problem of predicting future hospitalizations based on patients' Electronic Health Records (EHRs). Accurate predictions are valuable as they lead to improved health outcomes and reduction in health care costs through prevention. The potential for cost savings is huge; in the U.S. more than $ 30 billion are spent each year on hospitalizations deemed preventable [1]. Heart diseases, on which we focus later in our experimental results, are responsible for about 31% of that amount.

We can treat this problem as a classification problem, distinguishing between patients likely to be hospitalized or not. Intuitively, however, patients belong to different clusters depending on demographics and their ailments that are likely to cause a future hospitalization. Common supervised learning methods can certainly make classifications without considering these *hidden clusters*, yet, identifying the clusters can potentially improve classification performance. An additional key benefit of cluster identification is that results become more *interpretable*. Patients in the same cluster, especially if the cluster is identified based on a low-dimensional subspace of "diagnostic" features, share key characteristics and their cluster membership offers an explanation as to why they have been flagged for a future hospitalization. In the medical setting and elsewhere, interpretability has an essential role in persuading domain experts outside the machine learning community to trust the learning outputs and rely on them for their decision making.

EHRs exhibit interesting special structure in that for each patient only a very low-dimensional subset of features are important in predicting a future hospitalization. This subset is different for each cluster and, typically, there is no universal set of irrelevant features that can be eliminated. This suggests that it is useful to consider sparse classifiers for each cluster and, as we show, this positively impacts sample complexity.

The remainder of this paper is organized as follows. Sec. 2, briefly reviews related work, followed by a formal definition of our particular problem in Sec. 3. In Sec. 4, we present our approach, provide convergence guarantees, and characterize its VC dimension. We also establish sample complexity bounds for the classifiers. Experimental results from actual medical data are shown in Sec. 5. Conclusions are in Sec. 6.

2 Related work

In the literature, there are generally two types of assumptions about hidden clusters in a classification problem, implicit or explicit. The implicit approach is more prevalent, e.g., in piecewise linear techniques [2]. A more obvious assumption of hidden clusters (even though still implicit) is in feature space partitioning

*Research partially supported by the NSF under grants CNS-1239021 and IIS-1237022, by the NIH under R01-GM093147, by the ARO under W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under N00014-10-1-0952.

[†] Dept. of Electrical and Computer Eng., Boston Univ., {wydai, tbrisimi, tingxu, wty, srv}@bu.edu.

[‡] Dept. of Electrical and Computer Eng. and Dept. of Biomedical Eng., Boston Univ., yannisp@bu.edu.

methods such as in [3]. Yet another approach is to learn a mixture model of different SVMs applied to the data [4]. An explicit accounting of clusters within a classification problem is proposed in [5], where training samples are first put into clusters and then separate classifiers are trained.

A special feature of our problem is that the two classes are asymmetric in the sense that only the positive samples are assumed to have hidden clusters. A concrete example can be drawn again from medical diagnosis, where the positive class represents the unhealthy people and the negative class represents the opposite. It is intuitive that people get sick for various reasons (viewed as different clusters) while healthy people should be healthy in every aspect (thus, forming only one cluster). A similar asymmetric setting is also proposed in [6] where the data are assumed to be imbalanced and the larger class contains hidden clusters. Their solution is to solely cluster the larger class and train classifiers with copies of the samples from the other class. We design two methods along this direction which serve as a baseline for comparison.

Closer to our work is [7], also with a medical application. There, they try to maximize the margin between hidden clusters which is generally suitable for cases with only two hidden clusters. A simpler Discriminative Sub-Categorization (DSC) approach was proposed in [8] where joint clustering and SVM-based classification is performed. In the latter work, standard (vs. sparse) linear SVMs are used, all features are used for clustering, and no supporting theory is developed (e.g., VC-dimension bound or sample complexity results). We will directly compare our approach with DSC in our experimental results.

3 Problem definition

We consider a classification problem that has multiple hidden clusters in the positive class, while the negative class is assumed to be drawn from a single distribution. For different clusters in the positive class, we assume that the discriminative dimensions, with respect to the negative class, are different and sparse. We could think of these clusters as "local opponents" to the whole negative set (see Fig. 1) and therefore, the "local boundary" (classifier) could naturally be assumed to be different and lying in a lower-dimensional subspace of the feature vector. In summary, the classification problem satisfies the following assumptions: (i) The negative class samples are assumed to be i.i.d. and drawn from a single cluster with distribution P_0. (ii) The positive class samples belong to L clusters, with distributions $P_1^1, \dots, P_1^L$. (iii) Different positive clusters have different features that separate them from the negative samples.

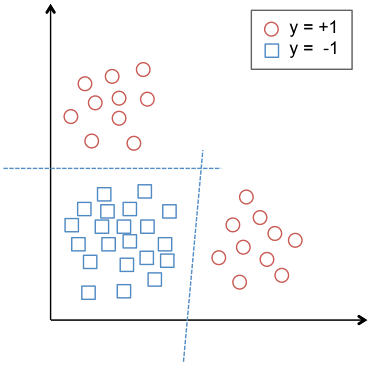

Figure 1: An example with two positive clusters as "local opponents."

We propose joint cluster detection and classification in an SVM framework. Let $(\mathbf{x}_i^+, y_i^+)$, $i = 1, \dots, N^+$, denote the $D+1$ dimensional positive samples, where $\mathbf{x}_i^+$ is the feature vector and $y_i^+ = 1$ the class label. Similarly, $(\mathbf{x}_j^-, y_j^-)$, $j = 1, \dots, N^-$, denotes the negative samples with $y_j^- = -1$. Assuming L hidden clusters in the positive class, we try to discover the L hidden clusters (denoted by a mapping function $l(i)$) and L sparse linear SVM classifiers, one for each cluster. Let $(\boldsymbol{\beta}^l, \beta_0^l)$ be the vector orthogonal to the SVM hyperplane for cluster l. Let also T^l be a parameter controlling per-cluster sparsity. The joint problem is:

$$\min_{\boldsymbol{\beta}^l,\beta_0^l,l(i)} \sum_{l=1}^{L}\left(\frac{1}{2}||\boldsymbol{\beta}^l||^2 + \lambda^+ \sum_{\{i:l(i)=l\}} \xi_i^l + \lambda^- \sum_{j=1}^{N^-} \zeta_j^l\right) \tag{1}$$

$$\text{s.t. } \textstyle\sum_{d=1}^{D} |\boldsymbol{\beta}_d^l| \leq T^l, \ \forall l,$$

$$\xi_i^l \geq 1 - y_i^+ \beta_0^l - \sum_{d=1}^{D} y_i^+ \beta_d^l x_{i,d}^+, \ \forall l, i : l(i) = l,$$

$$\zeta_j^l \geq 1 - y_j^- \beta_0^l - \sum_{d=1}^D y_j^- \beta_d^l x_{j,d}^-,\ \forall l, j = 1, \ldots, N^-,$$
$$\xi_i^l, \zeta_j^l \geq 0,\ \forall l, i : l(i) = l, j = 1, \ldots, N^-.$$

As mentioned earlier, the negative samples are not clustered but simply copied into each cluster. So their empirical costs are counted L times as shown in (1). The relative weight of costs from negative samples compared to that of the positive samples is controlled by λ^- and λ^+. The constraint $\sum_{d=1}^D |\beta_d^l| \leq T^l$ is an ℓ_1-relaxation of the sparsity requirement for the local classifiers.

4 Alternating clustering and classification

Problem (1) involves two sets of decision variables: $(\boldsymbol{\beta}^l, \beta_0^l)$ for the classifiers and $l(i)$ for cluster assignments. Altogether, the problem is a mixed integer programming problem, but given $l(i)$ it reduces to L quadratic optimization problems. This motivates our alternating optimization approach with two major modules: (i) training a classifier for each cluster and (ii) re-clustering positive samples given all the estimated classifiers.

Problem (1) uses the SVM framework. The formulation in (1) employs "soft margins" and by introducing slack variables it allows for some misclassification errors. A linear SVM and an RBF SVM (using a Radial Basis Function kernel) will serve as baseline methods in our experiments.

The classifier for each cluster is a linear SVM with an extra ℓ_1-constraint, which we call *Sparse Linear SVM (SLSVM)* (see also [9]). Specifically, the lth problem involves only objective terms for the lth cluster, constraints for all i such that $l(i) = l$, denoted by $i = 1, \ldots, N_l^+$, and all $j = 1, \ldots, N^-$ corresponding to all negative samples. We will use O^l to denote its optimal value. Note that $\sum_{l=1}^L N_l^+ = N^+$.

Next we present a result that suggests a favorable sample complexity for SLSVM compared to the standard linear SVM. Suppose that SLSVM for the lth cluster yields $Q^l < D$ non-zero elements of $\boldsymbol{\beta}^l$, thus, identifying a Q^l-dimensional feature subspace used for classification. The value of Q^l is controlled by T^l. Assume we draw a training set with N^- negative samples from P_0 and N_l^+ positive samples from P_1^l, where $N^l = N_l^+ + N^-$. Let R_N^l denote the expected training error rate and R^l the expected test error under these distributions.

Theorem 4.1 *For any $\epsilon > 0$ and $\delta \in (0, 1)$, if the sample size N^l satisfies*

$$N^l \geq \frac{8}{\epsilon^2}\left((Q^l + 1)\log\frac{2eN^l}{Q^l + 1} + Q^l \log\frac{eD}{Q^l} - \log\frac{2}{\delta}\right),$$

then with probability no smaller than $1 - \delta$, $R^l - R_N^l \leq \epsilon$.

Thm. 4.1 shows that the required sample size N^l is a linear function of Q^l and since Q^l could be much smaller than D this can lead to a much reduced need for training data and computational effort.

Initially, the positive samples can be randomly assigned into clusters and negative samples are copied into every cluster. After that, classifiers for each cluster could be estimated. We next describe how to re-cluster the positive samples given all estimated classifiers.

In this re-clustering, we allow additional flexibility concerning the features that determine the clusters. Specifically, the re-clustering algorithm does not have to use all of the features but can concentrate on just a subset. As discussed in the Introduction, this subset enables us to add prior knowledge about the clusters so that the identified clusters bear more intuitive explanations. We name the set of features used for re-clustering $\mathcal{C} \subseteq \{1, 2, \ldots, D\}$. The re-clustering algorithm is shown in Algorithm 1; we use prime for transpose. After re-clustering, positive samples are assigned to the cluster that has the maximum projection $\mathbf{x}_{i,\mathcal{C}}' \boldsymbol{\beta}_{\mathcal{C}}^l$. In this re-clustering module, we impose an important extra constraint (2) to guarantee the global convergence of the alternating process.

Different from typical clustering methods, such as k-means clustering [10], our re-clustering method does not assume any cluster centers. The reason is that we have label information for our samples and the goal of clustering is to assist classification. Therefore, our re-clustering intends to put samples into the right cluster such that the samples lie as far away as possible from the classification boundaries.

The whole process of *Alternating Clustering and Classification (ACC)* is shown in Algorithm 2. Once training has been performed with Alg. 2, we can classify a newly presented sample $\mathbf{x}$ not seen during training simply by assigning it to to cluster $l^* = \arg\max_l \mathbf{x}_{\mathcal{C}}' \boldsymbol{\beta}_{\mathcal{C}}^l$ and using classifier $(\boldsymbol{\beta}^{l^*}, \beta_0^{l^*})$.

Theorem 4.2 establishes the convergence of our ACC procedure. We note that tuning λ^+ and λ^- in ACC should be done globally, i.e., λ^+ and λ^- should be fixed across all clusters to guarantee convergence.

Algorithm 1 Re-clustering procedure given classifiers.

```
Input: positive samples $\mathbf{x}_i^+$, classifiers $(\boldsymbol{\beta}^l, \beta_0^l)$, current clusters assigning $i$ to cluster $l(i)$.
for all $i \in \{1, \dots, N^+\}$ do
  for all $l \in \{1, \dots, L\}$ do
    calculate projection $a_i^l = \mathbf{x}_{i,\mathcal{C}}^{+'} \boldsymbol{\beta}_{\mathcal{C}}^l$ of positive sample $i$ onto the classifier for cluster $l$ in the feature subspace corresponding to $\mathcal{C}$.
  end for
  update cluster assignment of sample $i$ from $l(i)$ to $l^*(i) = \arg\max_l a_i^l$, subject to
      $$\mathbf{x}_i^{+'} \boldsymbol{\beta}^{l^*(i)} + \beta_0^{l^*(i)} \geq \mathbf{x}_i^{+'} \boldsymbol{\beta}^{l(i)} + \beta_0^{l(i)}. \quad (2)$$
end for
```

Algorithm 2 Alternating Clustering and Classification Training.

```
Initialization:
For all $i = 1, \dots, N^+$ assign positive class sample $i$ to cluster $l(i) \in \{1, \dots, L\}$ (e.g., randomly).
repeat
  Classification Step:
  Train an SLSVM classifier for each cluster. Each classifier is the outcome of a quadratic optimization (cf. (1)) providing $(\boldsymbol{\beta}^l, \beta_0^l)$ and an optimal objective value $O^l$.
  Re-clustering Step:
  Re-cluster the positive samples by Alg. 1 and update the $l(i)$'s.
until no $l(i)$ is changed or $\sum_l O^l$ does not decrease.
```

Theorem 4.2 *For any $\mathcal{C}$, the ACC process converges.*

ACC yields L functions for clustering on a subset $\mathcal{C}$ of features and an up to D-dimensional classifier for each of the resulting clusters. Let the dimensionality of $\mathcal{C}$ be $D_{\mathcal{C}}$ (obviously, $D_{\mathcal{C}} \leq D$). Let $\mathcal{H}$ denote the family of clustering/classification functions produced by ACC. We have the following theorem bounding the VC-dimension of $\mathcal{H}$. The proof is based on [11] and is omitted due to space limitations.

Theorem 4.3 *The VC-dimension of the class $\mathcal{H}$ is bounded by $(L+1)L(D+1)\log\left(e^{\frac{(L+1)L}{2}}\right)$.*

Thm. 4.3 states that the VC-dimension of ACC grows linearly with the number of features D and polynomially with the number of clusters. Since the local classifiers are sparse, they likely have lower dimension than D. Moreover, the clustering function also lies in a lower dimensional space $\mathcal{C}$. Thus, the bound in Thm. 4.3 could be tighter in practice.

To demonstrate the superiority of our new ACC algorithm, we compare it with a conventional linear SVM and an RBF SVM. We also introduce two new hierarchical methods which naturally arise from our assumptions regarding the data. Finally, we compare ACC with DSC [8].

Since we assume that only the positive class contains clusters, during the model training phase we could first cluster the positive samples (still based on the feature set $\mathcal{C}$), then copy negative samples into each cluster, and finally optimize classifiers (linear SVMs) for each cluster. For clustering we adopt the widely used k-means method [10]. To classify new (test) samples we can use an approach just like the ACC method. We name this algorithm *Cluster Then Linear SVM (CT-LSVM).* The second hierarchical method we introduce is very similar to CT-LSVM but instead of training a linear SVM, we train a sparse linear SVM calling the method *Cluster Then Sparse Linear SVM (CT-SLSVM).*

Notice that an important difference between CT-LSVM, CT-SLSVM and ACC is that ACC has an alternating procedure while the other two do not. With only one-time clustering, CT-LSVM and CT-SLSVM create unsupervised clusters without making use of the negative samples, whereas ACC is taking class information and classifiers under consideration so that the clusters also help the classification. DSC, on the other hand, uses a joint clustering and classification approach but does not use sparse SVMs for each cluster and clusters using the full feature set. The latter, may lead to less interpretable clusters, which is an important consideration in our targeted medical data setting.

5 Experimental results

We tested ACC on the medical application we outlined in the Introduction. The data used for the experiments come from the Boston Medical Center (BMC). [1] In summary, we collect 10-year (2001-2010) records for a set of patients with at least one heart-related diagnosis or procedure within 01/2005–12/2010. The medical factors we extract include demographics, diagnoses, procedures, vitals, lab tests, tobacco use, emergency room visits, and admissions. Overall, the data set contains $45,579$ patients, of which 40% are randomly selected for training and the remaining 60% are used for testing. This random splitting is repeated 10 times. The objective is to predict whether a patient is hospitalized during a target year or not, given her/his past medical history. Hospitalized patients correspond to the *positive class* while non-hospitalized make up the *negative class*.

We compare our new algorithm to SVMs (linear and RBF), DSC, CT-LSVM and CT-SLSVM. Parameter tuning was done by 3-fold cross-validation with only training data. Some preliminary experiments led us to set $T^l = 6$. L explicitly varies in $(2, 3, 4, 5, 6)$ for all methods involving clustering. In Table 1, only the best results for CT-SLSVM and CT-SLSVM are presented (obtained under $L = 2$). In DSC, the parameter C which describes the penalty of the constraint violation is set to 10^3 (the default setting in the DSC code). We use the *Area Under the ROC Curve (AUC)* as the performance criterion, because it captures the trade-off between false positives and false negatives.

One important point for this experiment is that clustering with ACC uses a subset of "diagnostic" features, since these are the features that better delineate across different types of heart disease. The experimental results confirm that this intuition leads us to meaningful clusters. Table 1 shows the comparison between ACC and the alternative methods. For DSC we run two versions: one with an ℓ_2-based normalization of the input data (described in [8] and default in the DSC code) and one without normalization whose results are shown in parentheses in Table 1. In DSC, the parameter C is selected from $(1, 10, 10^2, 10^3, 10^4)$ by cross-validation. Initial clusters for both DSC and ACC are selected using a heuristic suggested in [8]. The "%" columns of Table 1 count the number of times (out of 10) that each method's AUC outperforms RBF SVM. It can be seen from the table that ACC outperforms the alternatives by anywhere between 1.85% and 9% in average AUC. Under $L = 3$, ACC outperforms RBF SVM in 10 our of 10 repetitions and outperforms DSC in 9 out of 10 repetitions.

Table 1: Average Prediction Accuracy (AUC) on Experimental Data.

Settings	avg AUC	std AUC	%	Settings	avg AUC	std AUC	%
ACC, $L = 2$	76.83%	0.87%	10	DSC, $L = 2$	68.02% (75.21%)	0.63% (0.76%)	0 (9)
ACC, $L = 3$	77.06%	1.04%	10	DSC, $L = 3$	68.69% (74.92%)	0.95% (1.67%)	0 (8)
ACC, $L = 4$	75.14%	0.92%	10	DSC, $L = 4$	69.05% (74.91%)	0.80% (0.79%)	0 (9)
ACC, $L = 5$	75.14%	1.00%	9	DSC, $L = 5$	69.76% (74.65%)	0.74% (1.09%)	0 (9)
ACC, $L = 6$	74.32%	0.87%	6	DSC, $L = 6$	70.24% (74.97%)	0.77% (0.69%)	0 (10)
Lin. SVM	72.83%	0.51%	3	RBF SVM	73.35%	1.07%	-
CT-LSVM ($L = 2$)	71.31%	0.76%	0	CT-SLSVM ($L = 2$)	71.97%	0.73%	1

In an attempt to interpret the ACC clusters we plot in Fig. 2 (Left) the mean value over each cluster of each element in the feature vector ($\mathbf{x}_\mathcal{C}$). This is done for a single repetition of the experiment and $L = 3$, which yields the best performance in average AUC. From Fig. 2 it is evident that the 3 clusters are well separated. Cluster 2 contains patients with other forms of chronic ischemic disease (mainly coronary atherosclerosis) and old myocardial infarction. Cluster 3 contains patients with dysrhythmias and heart failure. Cardiologists would agree that these clusters contain patients with very different types of heart disease. Finally, Cluster 1 contains all other cases with some peaks corresponding to endocardium/pericardium disease. An alternative view of cluster separation is shown in Fig. 2 (Right), where we project patient feature vectors on two features: dysrhythmias and other forms of chronic ischemic disease. To avoid overlap of different patients on the graph (natural as features take discrete values) we add a small uniform noise (in $[-0.1, 0.1]$) to each dimension of each sample. Cluster 1 patients are not visible as they are overwritten around $(0, 0)$.

6 Conclusions

We proposed a binary classification problem where the positive class consists of multiple "hidden" clusters. We developed an alternating minimization approach where at one stage local per-cluster classifiers are

[1] Proper IRB approval has been obtained.

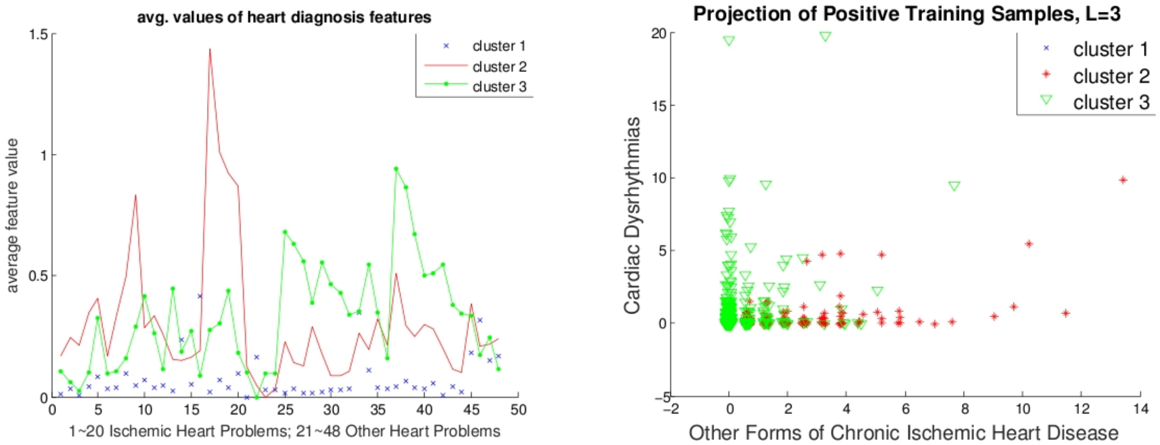

Figure 2: (Left): Average feature values in each cluster ($L = 3$). (Right): An alternative view of cluster separation.

optimized given cluster assignments, and at the second stage samples are re-clustered while the classifiers are fixed. We showed convergence of the method and established a bound on the VC dimension of the class of functions it produces. For the local per-cluster classifiers we proposed a sparse version of SVM that classifies in a lower-dimensional features subspace, potentially different for each cluster. We established a bound on the sample size needed for classification that grows linearly with the dimension of this subspace.

We tested our method on a medical dataset from the Boston Medical Center (BMC). The classification objective with the BMC data is to predict hospitalizations, which can lead to preventive actions and drastic reductions in health care spending. We compared our method against standard SVMs and other alternative classifiers involving a clustering step. The experimental results demonstrate the superiority of our approach in terms of prediction accuracy. An important benefit of our algorithm is that it identifies intuitively meaningful clusters. More specifically, samples in the same cluster share key features and cluster membership can be used to "explain" the assignment of a sample in the positive class. Especially in the medical application, the latter feature is crucial as it can guide decision making and help physicians "trust" the results.

Future work will test our method on diffent datasets and explore alternative classifiers (different from SVM), a larger number of clusters, and the sensitivity of the results to different proportions of positive vs. negative training examples.

References

[1] Jiang HJ, Russo CA, Barrett ML. Nationwide frequency and costs of potentially preventable hospitalizations, 2006. 2009;Available from: `http://www.hcup-us.ahrq.gov/reports/statbriefs/sb72.jsp`.

[2] Pele O, Taskar B, Globerson A, Werman M. The pairwise piecewise-linear embedding for efficient non-linear classification. In: Proceedings of The 30th International Conference on Machine Learning; 2013. p. 205–213.

[3] Breiman L, Friedman J, Olshen R, Stone C. Classification and Regression Trees. Wadsworth International Group; 1984.

[4] Fu Z, Robles-Kelly A, Zhou J. Mixing linear SVMs for nonlinear classification. Neural Networks, IEEE Transactions on. 2010;21(12):1963–1975.

[5] Gu Q, Han J. Clustered support vector machines. In: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics; 2013. p. 307–315.

[6] Zhao Y, Shrivastava A. Combating Sub-clusters Effect in Imbalanced Classification. In: IEEE 13th International Conference on Data Mining (ICDM); 2013. p. 1295–1300.

[7] Filipovych R, Resnick S, Davatzikos C. JointMMCC: Joint maximum-margin classification and clustering of imaging data. IEEE Transactions on Medical Imaging. 2012;31(5):1124–1140.

[8] Hoai M, Zisserman A. Discriminative sub-categorization. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE; 2013. p. 1666–1673.

[9] Gómez-Verdejo V, Martnez-Ramn M, Arenas-Garca J, Lzaro-Gredilla M, Molina-Bulla H. Support vector machines with constraints for sparsity in the primal parameters. IEEE Transactions on Neural Networks. 2011;22(8):1269–1283.

[10] Lloyd S. Least squares quantization in PCM. IEEE Transactions on Information Theory. 1982;28:129–137.

[11] Sontag ED. VC dimension of neural networks. In: Neural Networks and Machine Learning. Springer; 1998. p. 69–95.