Batch Classification with Applications in Computer Aided Diagnosis

Volkan Vural[1], Glenn Fung[2], Balaji Krishnapuram[2], Jennifer Dy[1], Bharat Rao[2]

[1] Department of Electrical and Computer Engineering, Northeastern University ,
[2] Computer Aided Diagnosis and Therapy, Siemens Medical Solutions, USA

Abstract. Most classification methods assume that the samples are drawn independently and identically from an unknown data generating distribution, yet this assumption is violated in several real life problems. In order to relax this assumption, we consider the case where batches or groups of samples may have internal correlations, whereas the samples from different batches may be considered to be uncorrelated. Two algorithms are developed to classify all the samples in a batch jointly, one based on a probabilistic analysis and another based on a mathematical programming approach. Experiments on three real-life *computer aided diagnosis* (CAD) problems demonstrate that the proposed algorithms are significantly more accurate than a naive SVM which ignores the correlations among the samples.

1 Introduction

Most classification systems assume that the data used to train and test the classifier is independently and identically distributed. For example, samples are classified one at a time in a *support vector machine* (SVM), thus the classification of a particular test sample does not depend on the features from any other test sample. Nevertheless, this assumption is commonly violated in many real-life problems where sub-groups of samples have a high degree of correlation amongst both their features and their labels.

Good examples of the problem described above are *computer aided diagnosis* (CAD) applications where the goal is to detect structures of interest to physicians in medical images: e.g., to identify potentially malignant tumors in computed tomography (CT) scans, X-ray images, etc. In an almost universal paradigm for CAD algorithms, this problem is addressed by a three-stage system: (1) identification of potentially unhealthy candidates *regions of interest* (ROI) from a medical image, (2) computation of descriptive features for each candidate, and (3) classification of each candidate (e.g. normal or diseased) based on its features. CAD applications were the main motivation for the work presented in this paper, although the algorithms presented here can be applied to any problem where the data is provided in batches of samples.

As an illustrative example, consider Figure 1, a CT image of a lung showing circular marks that point to potential diseased candidate regions that are detected by a CAD algorithm. There are five candidates on the left and six candidates on the right (marked by circles) in Figure 1. Descriptive features are extracted for each candidate and each candidate region is classified as healthy or unhealthy.

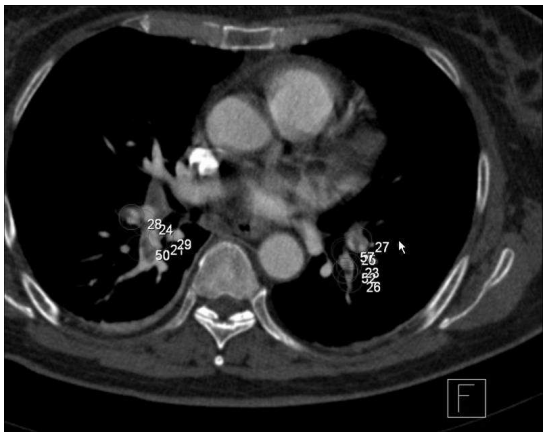


Fig. 1. *Two emboli as they are detected by the Candidate Generation algorithm in a CT image. The candidates are shown as five circles for the left embolus & six circles for the right embolus. The disease status of spatially overlapping or proximate candidates is highly correlated.*

In this setting, correlations exist among both the features and the labels of candidates belonging to the same (batch) image both in the training data-set and in the unseen testing data. Further, the level of correlation is a function of the pairwise-distance between candidates: the disease status (class-label) of a candidate is highly correlated with the status of other spatially proximate candidates, but the correlations decrease as the distance is increased. Most conventional CAD algorithms classify one candidate at a time, ignoring the correlations amongst the candidates in an image. By explicitly accounting for the correlation structure between the labels of the test samples, the algorithms proposed in this paper improve the classification accuracy significantly.

Beyond the domain of CAD applications, our algorithms are quite general and may be used for batch-wise classification problems in many other contexts. In general, the proposed classifiers can be used whenever data samples are presented in independent batches. In the CAD example, the batch corresponds to the candidate ROIs from an image, but in other contexts a batch may correspond to data from the same hospital, the patients treated by the same doctor or nurse, etc.

1.1 Related work

In natural language processing (NLP), *conditional random fields* (CRF) [4] and recently *maximum margin Markov* (MMM) networks [7] are used to identify part-of-speech information about words by using the context of nearby words. CRF are also used in similar applications in spoken word recognition. We are not aware of previous work on CAD algorithms that exploit internal correlations among the samples. However, while CRF and MMM are also fairly general algorithms, they are both computationally very demanding and it is also not very easy to implement them for problems where the relationship structure between the samples is in any form other than a linear chain (as in the text and speech processing applications). Certainly their application would be difficult

in many large-scale medical applications where run time requirements would be quite severe. For example, in the CAD applications shown in our experiments, the run-time of the testing phase usually has to be less than a second in order that the end user's (radiologist's) time would not be wasted.

Our algorithm is also related to the *multiple instance learning* (MIL) problem, where one is given bags (batches) of samples; class labels are provided only for the bags, not for the individual samples. A bag is labeled positive if we know that at least one sample from it is positive, and a negative bag is known to not contain any positive sample. In this manner, the MIL problem also encodes a form of prior knowledge about correlations between the labels of the training instances.

There are two differences between our algorithm and MIL. First, we want to classify each instance (candidate) in our algorithm; unlike MIL, we are not only trying to label a bag of related instances. Second, unlike the MIL problem which treats all the instances in a bag as equally related to each other, we account for more fine grained differences in the level of correlation between samples (via the covariance matrix Σ).

1.2 Organization of the paper

Section 2 presents the clinical motivation behind our work and describes the training and testing data that are used in these applications. In Section 3, we build a probabilistic model for batch classification of samples. Although dramatically faster than CRFs and their other cousins, the probabilistic algorithm is still too slow to be practical on several CAD problems, hence we propose another faster algorithm in Section 4. Unlike the previous methods such as CRF and MMM, both the proposed algorithms are easy to implement for arbitrary correlation relationships between samples, and further we are able to run these fast enough to be viable in commercial CAD products. In Section 5, we provide experimental evidence from three different CAD problems to show that the proposed algorithm is more accurate in terms of the metrics appropriate to CAD as compared to a naive SVM which is routinely used for these problems as the state-of-the-art in the current literature and commercial products. We conclude with a review of our contributions in Section 6.

Throughout this paper, we will utilize the following notations. The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix, A' will denote the transpose of A and A_i will denote the i-th row of A. All vectors will be column vectors. A vector of ones in a real space of arbitrary dimension will be denoted by e. Thus, for $e \in R^m$ and $y \in R^m$, $e'y$ is the sum of the components of y. A vector of zeros in a real space of arbitrary dimension will be denoted by 0.

2 Data in the medical domain

Data collection process for training CAD classifiers. Medical images (such as, CT scans, MRI, X-ray etc.) are collected from the archives of hospitals that routinely screen patients for cancer. Depending upon the disease, ground truth is determined for each patient based either on a more expensive, potentially invasive, test (e.g., biopsy of breast lesions, or colonoscopy for colon polyps), or via consensus opinion of a panel of expert radiologists for organs when a definitive test (lung biopsy) is deemed too dangerous.

In all cases, expert radiologist opinion is also required to mark the location, size, and extent of all "positive" regions within the images. A CAD system is then designed from the database of training images. Considerable human intervention and domain knowledge engineering is employed on the first two stages of a CAD system: (a) candidate generation: identify all potentially suspicious regions in a candidate generation stage with very high sensitivity, and (b) feature-extraction: to describe each such region quantitatively using a set of medically relevant features. For example, quantitative measurements based on texture, shape, intensity, contrast and other such characteristics may be used to characterize any region of interest (ROI). Finally, the candidate ROIs are assigned class labels based upon the overlap or spatial proximity to any radiologist-marked (diseased) region.

From the above description it is clear that the samples (candidates) are naturally collected in batches. While there are no correlations between the candidate ROIs in different images, the labels of all the regions identified from the same patient's medical images are likely to be at least somewhat correlated. This is true both because metastasis is an important possibility in cancer, and because the patient's general health and preparation for imaging are important factors in diagnostic classification (e.g., how thoroughly was the cleaning of stool undertaken before a colonoscopy). Further, in order to identify suspicious regions with high sensitivity, most candidate generation algorithms tend to produce several candidates that are spatially close to each other, often referring to the same underlying structure in the image. Since they often refer to regions that are physically adjacent in an image, both features and class labels for these candidates are also highly correlated.

Shortcomings in standard classification algorithms. Most of the classification algorithms such as *neural networks* and *support vector machines* (SVM) assume that the training samples or instances are drawn identically and *independently* from an underlying distribution. However, as mentioned in the introduction and in the previous subsection, due to spatial adjacency of the regions identified by a candidate generator, both the features and the class labels of several adjacent candidates are highly correlated. This is true both in the training and testing data. The proposed batch-classification algorithms account for these correlations explicitly.

3 A probabilistic batch classification model

Let $x_i^j \in R^n$ represent the n features for the i^{th} candidate in the j^{th} image, and let $w \in R^n$ be the parameters of some hyperplane classifier. Traditionally, linear classifiers label samples one at a time (i.e., independently) based on:

$$z_i^j = w' x_i^j = (x_i^j)' w \ , z \in R^1 \tag{1}$$

For example, in logistic regression, the posterior probability of the sample x_i^j belonging to class $+1$ is obtained using the sigmoid function $P(y_i^j = 1 | x_i^j) = \frac{1}{1+\exp(-w' x_i^j)}$.

By contrast, in our model, we claim z_i^j is only a noisy observation of the underlying, unobserved variable $u_i^j \in R^1$ that actually influences classification (as opposed to the traditional classification approach, where classification directly depends on z_i^j).

We have an a-priori guess or intuition about u_i^j even before we observe any x_i^j (therefore before z_i^j), which is purely based on the proximity of the spatial locations of candidates in the j^{th} image. Indeed this spatial adjacency is what induces the correlation in the predictions for the labels; we model this as a Gaussian prior on u_i^j.

$$P(u^j \in R^{n_j}) = N(u^j|0, \Sigma^j) \tag{2}$$

where n_j is the number of the candidates in the j^{th} image, and the covariance matrix Σ^j (which encodes the spatial proximity based correlations) can be defined in terms of S, the matrix of Euclidean distances between candidates inside a medical image (from a patient) as $\Sigma^j = \exp(-\alpha S)$.

Having defined a prior, next we define the likelihood as follows:

$$P(z_i^j|u_i^j) = N(z_i^j|u_i^j, \sigma^2) \tag{3}$$

After observing x_i^j and therefore z_i^j, we can modify our prior intuition about u^j in (2), based on our observations from (3) to obtain the Bayesian posterior:

$$P(u^j|z^j) = N\left(u^j|({\Sigma^j}^{-1}\sigma^2 + I)^{-1}z^j; ({\Sigma^j}^{-1} + \tfrac{1}{\sigma^2}I)^{-1}\right) \tag{4}$$

The class-membership prediction for the i^{th} candidate in the j^{th} image is controlled exclusively by u_i^j. The prediction probability for class labels, y^j is then determined as:

$$P(y^j = 1|B^j, w, \alpha, \sigma^2) = 1/\left(1 + \exp\left(-[{\Sigma^j}^{-1}\sigma^2 + I]^{-1}[B^j w]\right)\right). \tag{5}$$

Where $B^j \in R^{m_j \times n}$ represents the m_j training points that belong to the j^{th} batch. Note however, that this approach to batchwise prediction is potentially slow due to the matrix inversion, if the test data arrives in large batches.

3.1 Learning in this model

For batch-wise prediction using (5), w, α and σ^2 can be learned from a set of N training images via *maximum-a-posteriori* (MAP) estimation as follows:

$$[\hat{w}, \hat{\alpha}, \hat{\sigma^2}] = \arg\max_{w,\alpha,\sigma^2} P(w) \prod_{j=1}^{N} P(y^j|B^j, w, \alpha, \sigma^2) \tag{6}$$

where, $P(y^j|B^j, w, \alpha, \sigma^2)$ is defined as in (5) and $P(w)$ may be assumed to be Gaussian $N(w|0, \lambda)$. The regularization parameter λ is typically chosen by cross-validation.

3.2 Intuition about batch classification

Equations (4) and (5) imply that $\mathbb{E}[u^j|z^j] = ({\Sigma^j}^{-1}\sigma^2 + I)^{-1}z^j$. In other words, the class membership prediction for any single sample is a weighted average of the noisy prediction quantity z^j (distance to the hyperplane), where the weighting coefficients depend on the pairwise Euclidean distances between samples. Hence, the intuition presented above is that we predict the classes for all the n_j candidates in the j^{th} image together, as a function of the features for all the candidates in the batch (here a batch corresponds to an image from a patient). In every test image, *each* of the candidates is classified using the features from *all* the samples in the image.

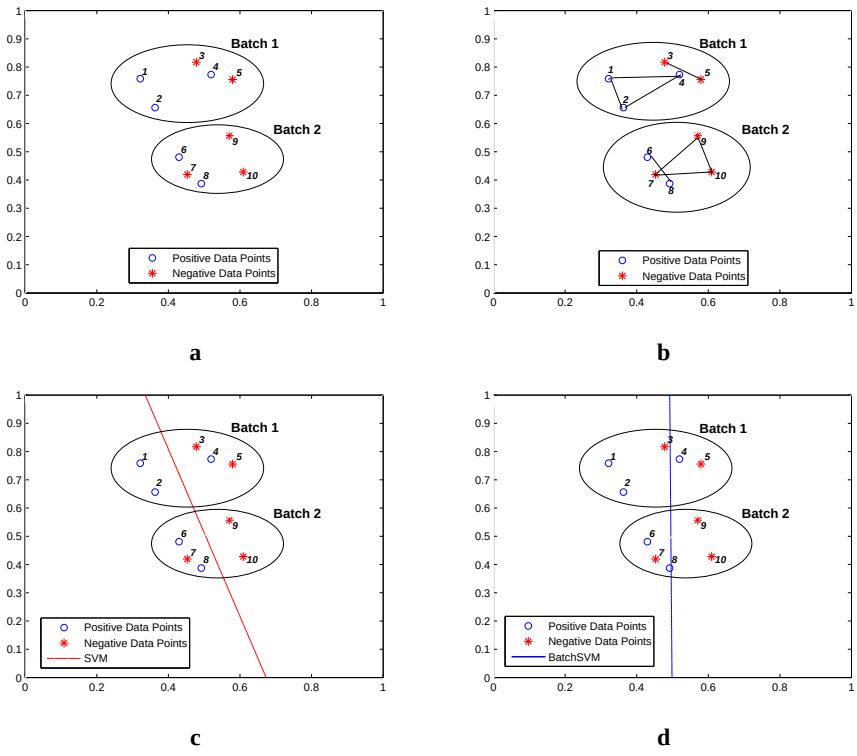


Fig. 2. An illustrative example for batch learning. **a**) Training data points are displayed in batches. **b**) Relations within training points are displayed as a linked graph. **c**) Classifier produced by SVM. **d**) Pre-classifier produced by BatchSVM. Unlike standard SVMs, the hyperplane, $f(x)$, produced by BatchSVM (preclassifier) is not the decision function. Instead, the decision of each test sample x_i, is based on a weighted average of the $f(x)$ values for the points linked to x_i.

4 A mathematical programming approach

Motivated by equations (5) and (6), we now re-formulate the problem of learning for batch-wise prediction as an SVM-like mathematical program.

In a standard SVM a hyperplane classifier, $f(x) = x'w - \gamma$ is learned from the training instances individually, ignoring the correlations among them. Consider the problem of classifying m points in the n-dimensional real space R^n, represented by the $m \times n$ matrix A, according to class membership of each point x_i (ith row of A) in the classes $A+$, $A-$ as specified by a given $m \times m$ diagonal matrix D with $+1$ or -1 along its diagonal, this is, $D = diag(y)$. The standard 1-norm support vector machine with a

linear kernel [8, 2] is given by the following linear program with parameter $\nu > 0$:

$$\begin{aligned} \min_{(w,\gamma,\xi,v)\in R^{n+1+m+n}} & \nu e'\xi + e'v \\ \text{s.t.} D(Aw - e\gamma) + \xi &\geq e \\ v \geq w &\geq -v \\ \xi &\geq 0 \end{aligned} \tag{7}$$

where, ν is the cost parameter and at a solution, $v = |w|$ is the absolute value of w.

While $A \in R^{m\times n}$ represents the entire traning data, $B^j \in R^{m_j \times n}$ represents the m_j training points that belong to the j^{th} batch and the labels of these training points are represented by the $m_j \times m_j$ diagonal matrix $D^j = diag(y^j)$ with positive or negative ones along its diagonal. Then, the standard SVM set of constraints: $D(Aw-e\gamma)+\xi \geq e$ can be modified in order to take into account the correlations among samples in the same batches, using the idea in equation 5 as:

$$D^j \left[\left({\Sigma^j}^{-1}\sigma^2 + I \right)^{-1} (B^j w - e\gamma) \right] + \xi^j \geq e, \quad \text{for } j = 1, \ldots, k \tag{8}$$

In a naive implementation, for each batch j, the probabilistic method requires calculating two matrix inversions to compute $\left(\Sigma^{-1}\sigma^2 + I\right)^{-1}$. Hence, training and testing using this method can be time consuming for large batch sizes. In order to avoid this problem while retaining the intuition presented in subsection 3.2, we modify equation (8). In particular, we replace the expression $\left(\Sigma^{-1}\sigma^2 + I\right)^{-1}$ by a much simpler expression: $(\Sigma\theta + I)$. As a result, the correlation among samples belonging to the same batch can be enforced by replacing the standard set of SVM constraints by:

$$D^j \left[\left(\theta\Sigma^j + I\right) (B^j w - e\gamma) \right] + \xi^j \geq e, \quad \text{for } j = 1, \ldots, k \tag{9}$$

As in equation (8), the class membership prediction for any single sample in batch j is a weighted average of the batch members prediction vector $B^j w$, and again the weighting coefficients depend on the pairwise Euclidean distances between samples. Using this constraint in the SVM equations (7), we obtain the optimization problem for learning BatchSVM with parameters ν and θ:

$$\begin{aligned} \min_{(w,\gamma,\xi,v)\in R^{n+1+m+n}} & \nu e'\xi + e'v \\ \text{s.t.} D^j \left[\left(\theta\Sigma^j + I\right) (B^j w - e\gamma) \right] + \xi^j &\geq e, \quad \text{for } j = 1, \ldots, k \\ v \geq w &\geq -v \\ \xi &\geq 0 \end{aligned} \tag{10}$$

Unlike standard SVMs, the hyperplane ($f(x) = w^{'} x - \gamma$) produced by BatchSVM is *not* the final decision function. We refer to $f(x)$ as a pre-classifier that will be used in the next stage to make the final decision on a batch of instances. While testing an arbitrary datapoint x_i^j in batch B^j, the BatchSVM algorithm accounts for the pre-classifier prediction $w' x_p^j$ for every member in the batch. The final prediction $\hat{f}(x_i^j)$ is given by:

$$sign(\hat{f}(x_i^j)) = sign(w^{'} x_i^j - \gamma + \theta\Sigma_i^j \left[B^j w - \gamma\right]) \tag{11}$$

Point	Batch	Label	SVM	Pre-classifier	Final classifier
1	1	+	0.2826	0.1723	0.1918
2	1	+	0.2621	0.1315	0.2122
3	1	-	-0.2398	**0.0153**	-0.0781
4	1	+	**-0.3188**	**-0.0259**	0.2909
5	1	-	-0.4787	-0.0857	-0.0276
6	2	+	0.2397	0.0659	0.0372
7	2	-	**0.2329**	**0.0432**	-0.0888
8	2	+	0.1490	0.0042	0.0680
9	2	-	-0.2525	-0.0752	-0.1079
10	2	-	-0.2399	-0.1135	-0.1671

Table 1. Outputs of the classifier produced by SVM, pre-classifier and the final classifier produced by BatchSVM. The outputs are calculated for the data points presented in Figure 2. The first column of the table indicates the order of the data points as they are presented in Figure 2a and the second column specifies the corresponding labels. Misclassified points are displayed in bold. Notice that the combination of the pre-classifier outputs at the final stage corrects the mistakes.

To obtain a more general nonlinear algorithm, we can "kernelize" equations (10,11) by making a transformation of the variable w as: $w = A'v$, where v can be interpreted as an arbitrary variable in $\Re^m$. This transformation can be motivated by duality theory [5]. Employing this idea will result in a term $B^j A'v$ instead of $B^j w$ in our formulations. If we now replace the linear kernels, $B^j A'$, by more general kernels, $K(B^j, A')$, we obtain a "kernelized" version of equations (10,11).

Consider the two dimensional example in Figure 2, showing batches of training points. The data points that belong to the same batch are indicated by the elliptical boundaries in the figure. Figure 2b displays the correlations amongst the training points given in Figure 2a using an edge. In Figure 2c, the hyperplane $f_{svm}(x)$ is the final decision function for standard SVM and gives the results displayed in Table 1, where we observe that the fourth and the seventh instances are misclassified. In Figure 2d, the pre-classifier produced by BatchSVM, $f_{batch}(x)$ gives the results displayed in the fifth column of Table 1 for the training data. If this pre-classifier were to be considered as the decision function, then three training points would be misclassified. However, during batch-testing (eq 11), the predictions of those points are corrected as seen in the sixth column of Table 1.

5 Experiments

5.1 The similarity function

As mentioned earlier, the matrix Σ^j represents the level of correlation between all pairs of candidates from a batch (an image in our case) and it is a function of the pairwise-similarity between them. In CAD applications, the covariance matrix Σ^j can be defined in terms of the matrix of Euclidean distances between candidates inside a medical image. Let r_p and r_q represent the coordinates of two candidates, B_p^j and B_q^j on the j^{th}

image. For our experiments, we used the Euclidean distance between r_p and r_q to define the pairwise-similarity, $s(p,q)$, between B_p^j and B_q^j as: $s(p,q) = \exp\left(-\alpha \|r_p - r_q\|^2\right)$.

Experimentally, we found it useful to discretize the continuous similarity function, $s(p,q)$ to the binary similarity function, $s^*(p,q)$ by applying a threshold as following:

$$s^*(p,q) = \begin{cases} 0\ , s(p,q) < e^{-4} \\ 1\ , s(p,q) \geq e^{-4} \end{cases} \tag{12}$$

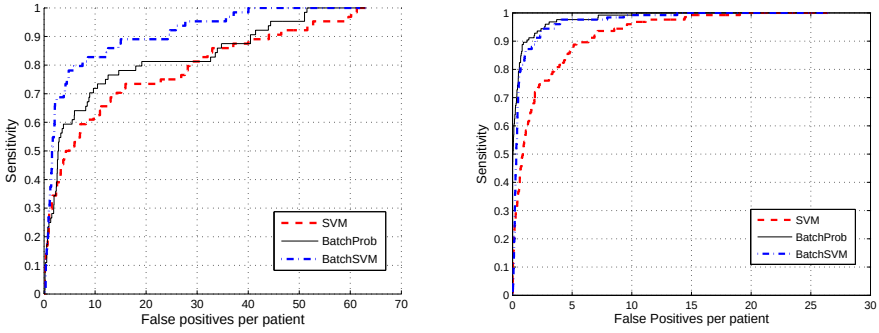
In all experiments, we set the threshold at e^{-4} to provide us with a similarity of one if the neighbor is at a 95% confidence level of belonging to the same density as the candidate assuming that the neighborhood is a Gaussian distribution with mean equal to candidate and variance $\varsigma^2 = \frac{1}{\alpha}$. Each element of Σ is given by: $\Sigma_{pq} = s^*(p,q)$.

5.2 Comparisons

In this section, we compare three techniques: regular SVM, probabilistic batch learning ($BatchProb$), and BatchSVM. Receiver Operating Characteristic (ROC) plots are used to study the classification accuracy of these techniques on three CAD applications for detecting pulmonary embolism, colon cancer, and lung cancer. In clinical practice, CAD systems are evaluated on the basis of a somewhat domain-specific metric: to maximize the fraction of *positives* that are correctly identified by the system while displaying at most a clinically acceptable number of false-marks per image. We report this domain-specific metric in an ROC plot, where the y-axis is a measure of sensitivity and the x-axis is the number of false-marks per patient (in our case, per image is also per patient). Sensitivity is the number of patients diagnosed as having the disease divided by the number of patients that has the disease. High sensitivity and low false-marks are desired. All our parameters in these experiments are tuned by 10-fold patient cross-validation on the training data (i.e., the training data is split into ten folds). During cross-validation, a range of parameters $(\theta, \sigma, \varsigma)$ were evaluated for the proposed methods: for θ in $BatchSVM$ and σ in $BatchProb$, we considered $-1, -0.9, ..., 0.9, 1$ and for ς that is necessary for Σ matrix, we used a logarithmically spaced range from 10^{-3} through 10^{1}. All classification algorithms are trained on the training dataset and evaluated on the sequestered (held-out) test set.

5.3 Data Sources and Domain Description

Example: Pulmonary Embolism Pulmonary embolism (PE), a potentially life threatening condition, is a result of underlying venous thromboembolic disease. An early and accurate diagnosis is the key to survival. Computed tomography angiography (CTA) has emerged as an accurate diagnostic tool for PE. There are hundreds of CT slices in each CTA study, thus manual reading is laborious, time consuming and complicated by various PE look-alikes (false positives). Several CAD systems are developed to assist radiologists in this process by helping them detect and characterize emboli in an accurate, efficient and reproducible way [6], [9]. We have collected 72 cases with 242 PEs marked by expert chest radiologists at four different institutions (two North American sites and two European sites). For our experiments, they were randomly divided into



a. PE **b. Colon**

Fig. 3. SVM, BatchProb and BatchSVM ROC curves comparisons for (a) the PE data and (b) the Colon Cancer data

two sets: a training and a testing set. The training set was used to train and validate the classifiers and consists of 48 cases with 173 PEs and a total of 3655 candidates. The testing set consists of 24 cases with 69 true PEs out of a total of 1857 candidates. This set was only used to evaluate the performance of the final system. A combined total of 70 features were extracted for each candidate.

Example: Colon Cancer Detection Colorectal cancer is the third most common cancer in both men and women. It is estimated that in 2004, nearly $147,000$ cases of colon and rectal cancer will be diagnosed in the US, and more than $56,730$ people would die from colon cancer [3]. In over 90% of the colon cancer cases that progressed rapidly is from local (polyp adenomas) to advanced stages (colorectal cancer), which has very poor survival rates. However, identifying (and removing) lesions (polyp) when still in a local stage of the disease, has very high survival rates, thus illustrating the critical need for early diagnosis. Most polyps in the training data are inherently represented by multiple candidates. The database of high-resolution CT images used in this study were obtained from seven different sites across US, Europe and Asia. The 188 patients were randomly partitioned into a training and a test set. The training set consists of 65 cases containing 127 volumes. Fifty polyps were identified in this set out of a total of 6748 candidates. The testing set consists of 123 cases containing 237 volumes. There are 103 polyps in this set from a total of 12984 candidates. A total of 75 features were extracted for each candidate.

Example: Lung Cancer LungCAD is a computer aided detection system for detecting potentially cancerous pulmonary nodules from thin slice multi-detector computed tomography (CT) scans. The final output of LungCAD is provided by a classifier that classifies a set of candidates as positive or negative. This is a very hard classification problem: most patient lung CTs contain a few thousand structures (candidates), and

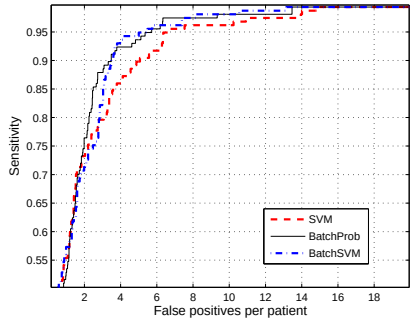


Fig. 4. SVM, BatchProb and BatchSVM ROC curves comparisons for the Lung Cancer data

only a few (≤ 5 on average) of which are potential nodules that should be identified as positive by LungCAD, all within the run-time requirements of completing the classification on-line during the time the physician completes their manual review. The training set consists of 60 patients. The number of candidates labeled as nodules in the training set are 157 and the total number of candidates is 9987. The testing set consists of 26 patients. In this testing set, there are 79 candidates labeled as nodules out of 6159 generated candidates. The number of features extracted for this dataset were 15.

5.4 Results:

Figures 3a, 3b, and 4 show the ROC curves for pulmonary embolism, colon cancer, and lung cancer data respectively. In our medical applications high-sensitivity is critical as early detection of lung and colon cancer is believed to greatly improve the chances of successful treatment [1]. Furthermore, high specificity is also critical, as a large number of false positives will vastly increase physician load and lead (ultimately) to loss of physician confidence.

In Figure 3a, corresponding to the comparison of the ROC curves on the PE dataset, we observe that standard SVM can only achieve 53% sensitivity for six false positives. However, BatchSVM achieves 80% with a remarkable improvement (27%). BatchProb also outperforms SVM with a 64% sensitivity. As seen from the figure, the two proposed methods are substantially more accurate than standard SVMs at any specificity level.

Colon cancer data is a relatively easier data set than pulmonary embolism since standard SVM can achieve 54.5% sensitivity at one false positive level as illustrated in Figure 3b. However, BatchSVM improved SVM's performance to 84% sensitivity for the same number of false positives. Note that BatchProb improved the sensitivity further, giving 89.6% for the same specifity. In one to ten false positives region which constitutes the region of interest in our applications, our proposed methods outperform standard SVM significantly.

Although SVM is very accurate for lung cancer application, Figure 4 shows that BatchProb and BatchSVM could still improve SVM's performance further. BatchProb

method is superior to the other methods at two and three false positives per image. Both BatchProb and BatchSVM outperform SVM in the 2-6 false positives per image region, which is the region of interest for commercial clinical lung CAD systems. All three of the methods are comparable at other specificity levels.

6 Conclusions

Two related algorithms have been proposed for classifying batches of correlated data samples. Although primarily motivated by real-life CAD applications, the problem occurs commonly in many situations; our algorithms are sufficiently general to be applied in other contexts. Experimental results indicate that the proposed method can substantially improve the diagnosis of (a) early stage cancer in the Lung & Colon, and (b) pulmonary embolisms (which may result in strokes). With the increasing adoption of these systems in routine clinical practice, these experimental results demonstrate the potential of our methods to impact a large cross-section of the population.

7 Acknowledgments

This research is supported by the Computer Aided Diagnosis Group, Siemens Medical Solutions, USA and NSF CAREER IIS-0347532.

References

1. L. Bogoni, P. Cathier, M. Dundar, A. Jerebko, S. Lakare, J. Liang, S. Periaswamy, M. Baker, and M. Macari. CAD for colonography: A tool to address a growing need. *British Journal of Radiology*, 78:57–62, 2005.
2. P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proc. 15th International Conf. on Machine Learning*, pages 82–90, San Francisco, CA, 1998. Morgan Kaufmann.
3. D. Jemal, R. Tiwari, T. Murray, A. Ghafoor, A. Saumuels, E. Ward, E. Feuer, and M. Thun. Cancer statistics, 2004.
4. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289, San Francisco, CA, 2001. Morgan Kaufmann.
5. O. L. Mangasarian. Generalized support vector machines. In *Advances in Large Margin Classifiers*, pages 135–146, 2000.
6. M. Quist, H. Bouma, C. V. Kuijk, O. V. Delden, and F. Gerritsen. Computer aided detection of pulmonary embolism on multi-detector CT, 2004.
7. B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
8. V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
9. C. Zhou, L. M. Hadjiiski, B. Sahiner, H.-P. Chan, S. Patel, P. Cascade, E. A. Kazerooni, and J. Wei. Computerized detection of pulmonary embolism in 3D computed tomographic (CT) images: vessel tracking and segmentation techniques. In *Medical Imaging 2003: Image Processing. Proceedings of the SPIE, Volume 5032, pp. 1613-1620 (2003).*, pages 1613–1620, May 2003.