Incremental Classification Learning for Anomaly Detection in Medical Images

Balathasan Giritharan, Xiaohui Yuan, Jianguo Liu
Computer Vision and Intelligent Systems Lab
University of North Texas, Denton, TX

INTRODUCTION

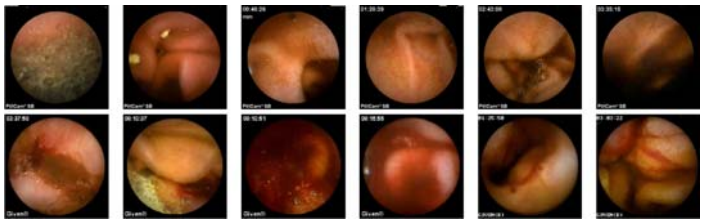
In diagnosis of new instances, disagreement occurs between a CAD system and physicians, which could suggest inaccurate classifiers. Intuitively, misclassified instances and the previously acquired data should be used to retrain the classifier. This, however, is very time consuming and, in some cases where dataset is too large, becomes infeasible.

Among the patient data, only a small percentile shows positive signs. Such a phenomenon is called imbalanced data. The minority class affects the performance of the classification methods, and the challenge lies in learning the distribution of the minority dataset.

The incremental learning strategy described in this article uses Support Vector Machines (SVM) to learn the distribution of the minority class and integrate the misclassified examples on-the-fly.

Incremental learning algorithms are usually applied to problems where the complete training set becomes available over time, and where the complete training sets are larger in size, and makes the learning process practical when limited learning power available.

Our main contribution of this work is using incremental SVM to learn the distribution of the imbalanced dataset, as new exemplars become available.



SUPPORT VECTOR MACHINE

SVMs use structural risk minimization principle to construct decision rules using hyperplanes. Suppose we have a training set

$$(x_i, y_i) \text{ and } i \in \{1, 2, \ldots, n\}, x \in R^m, y = \pm 1$$

The linear hyperplanes that represent the decision function are described as follows:

$$f(x) = \text{sgn}((w \cdot x) + b) \quad (1)$$

Derived from a separating hyperplane which generalizes well and can be found by minimizing the following error function,

$$\frac{1}{2}\|w\|^2 + \gamma \sum_{i=1}^{n} \xi_i \quad (2)$$

Subject to

$$\xi_i \geq 0, \quad y_i(w \cdot x_i) + b \geq 1 - \xi_i \quad \text{for } i = 1, \ldots, n \quad (3)$$

The solution to this problem can be shown to have an expansion $w = \sum_{i=1}^{n} \alpha_i x_i$

where only those nonzero coefficients α_i satisfy the constraint in Eq-(3). The corresponding x_i lie closest to the decision boundary and are called the support vectors. The coefficients α_i (the variables from the dual of the given minimization problem) are found by solving the quadratic programming problem defined by Eq-(2)&(3).

METHODOLOGY

Retraining an SVM with all the available examples is impractical in many real world.

Bootstrapping is a method for manipulating the sample distribution of an estimator by selecting instances from the original dataset with replacement. Bootstrap derives more robust estimates of standard errors and confidence intervals of a population parameter such as mean and standard deviation. It is often used as a robust alternative to perform inference using parametric assumptions when assumptions are in doubt or the parametric inference is impossible or requires very complicated formulas for the calculation of standard errors.

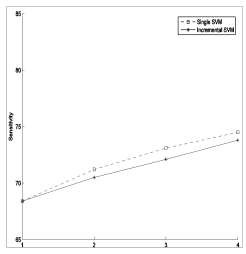
Here we use bootstrapping at incremental steps, to select potential candidates support vectors in future.

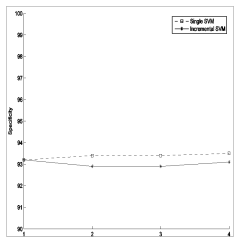
1. Let WS = Support Vectors and misclassified examples of training SVM using U_0
2. For each incremental step
   (a) Let U_n be the new exemplars available
   (b) for $j = 1 \cdots k$
      i. V_j = M Random samples selected from U_n
      ii. SV_j = Support Vectors of training SVM using V_j
      iii. $WS = WS \cup SV_j$
   (c) U_n^* be the misclassified samples, of a new SVM tranined using WS
   (d) The working set for the next iteration $WS = WS \cup U_n^*$

EXPERIMENTS

To evaluate the effectiveness of incremental learning, the performance of an incrementally trained SVM is compared with the SVM trained with all data available at each incremental step.

We report the average results of 5 repetitions of two randomly selected videos initially, incrementally training using the remaining three and testing was performed on two full length videos. SVMs were trained using the radial basis kernel function and bootstrapping was achieved by randomly creating *k* bootstraps.





	SVM	Incremental SVM
Starting	120 mins	120 mins
Incremental Step #1	210 mins	14 mins
Incremental Step #2	250 mins	17 mins
Incremental Step #3	320 mins	18 mins