Learning from Large Data Set — Segmentation of Capsule Endoscopy Videos

Xiaohui Yuan *
University of North Texas,
Denton TX 76201
USA
E-mail: xiaohui.yuan@unt.edu * Corresponding author

Balathasan Giritharan
University of North Texas,
Denton TX 76201
USA
E-mail: giri@unt.edu

Sandeep Panchakarla
University of North Texas,
Denton TX 76201
USA
E-mail: sp0353@unt.edu

Abstract: Reviewing video of capsule endoscopy is a tedious work that takes hours. Hence, efficient and scalable approaches are needed to automate the process of large dataset and be able to refine the model given new examples. This paper presents an incremental SVM to learn from large dataset with dynamic patterns. Our method extends the reduced convex hull concept and defines the approximate skin segments of convex hulls. Experiments were conducted using synthetic data set, real-world data sets, and CE videos. Our results demonstrated highly competitive performance that requires much less resource, which cast new light on learning with limited resource.

Keywords: Classification; Video Segmentation; Incremental Learning.

Reference to this paper should be made as follows: Yuan, X., Giritharan, B., and Panchakarla S. (xxxx) 'Learning from Large Data Set — Segmentation of Capsule Endoscopy Videos', *Int. J. of Functional Informatics and Personalized Medicine*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Xiaohui Yuan received a B.S. degree in Electrical Engineering from Hefei University of Technology, China in 1996 and a Ph.D. degree in Computer Science from Tulane University, U.S. in 2004. After graduation, he worked at the National Institutes of Health for two years on medical image analysis. He is an Assistant Professor in the Computer Science and Engineering Department at the University of North Texas. His research interests include data mining, machine learning, image processing, and pattern recognition, and he has published more than 50 technical papers. He is a member of IEEE and SPIE.

Balathasan Giritharan received a B.S. degree in Computer Science from University of Peradeniya, Sri Lanka in 2002. From 2002 to 2005 he was a lecturer at University of Peradeniya. Now he is pursuing his Ph.D. in the Department of Computer Science and Engineering at the University of North Texas. His research interests include pattern recognition and image processing.

Sandeep Panchakarla received a B.S. Degree in the Computer Science and Engineering Department at the Acharya Nagarjuna University, India in 2008. Currently he is pursuing Masters Degree in the Department of Computer Science and Engineering at the University of North Texas, U.S.A. His research interests include computer vision and machine learning.

1 Introduction

Wireless capsule endoscopy (WCE) is a recently established technology that requires no wired device intrusion and can be used to examine the entire small intestine non-invasively. The imaging component of this system is a vitamin-sized capsule that is composed of a color CMOS camera, a battery, a light source and a wireless transmitter. It provides a 140-degree field of view and generates 256×256 images. Once the device is activated, it is ready to take pictures. The camera acquires two pictures every second for approximately eight hours, transmitting images to a recording device worn by the patient. By using a lens of short focal length, images are obtained as the capsule is propelled through the tract. Unlike conventional fiber-optic endoscopy, WCE requires little patient preparation and can potentially image any section in the digestive system. The ability of WCE to detect undersized lesions in the small intestine is ideally suited for this particular role. It enables physicians to examine the entire small intestine, a region that was previously difficult to view at all, and provide a new non-invasive gastrointestinal (GI) visualization technology. The diagnostic yield using WCE is much higher compared to other endoscopic imaging methods. Capsule endoscopy has the potential for use in a wide variety of illnesses.

Recognizing where a WCE frame is taken in the digestive tract is vital to diagnosis and treatment deployment. An important question that often arises at diagnosis is where the lesion is found. Reviewing WCE videos and estimating the anatomical locations of WCE frames are, however, very difficult, even for experienced readers. The primary reasons are inconsistent speed of WCE device and lack of physical landmarks. The current technology relies on wireless signal strength, which is used to calculate the distance of the device to the data receiver. This method provides very coarse anatomical trace. A common practice by physicians is reading WCE frames to identify some specific gastrointestinal (GI) images, known as GI landmarks, that indicate entrance to a GI section [1, 2].

With improved optical sensors the size of each image captured is expected to increase. In addition, to obtain better generalization performance and avoid curse of dimensionality, we expect to train classification algorithms using videos of multiple human subjects. The implementation of classification algorithms that require all training data to be present in memory make the learning task extremely challenging. Existing

classification algorithms are facing difficulties in handling large number of training samples.

Incremental learning has great potential to handle these challenges. At each incremental step, only the samples that are useful to build future classifiers along with model information is retained. When new data becomes available, they are integrated with the retained samples, and the model is updated and/or trained to build a new classifier. Support Vector Machines (SVM) [3] is a supervised learning algorithm that can be applied to classification or regression. The SVM algorithm is based on the statistical learning theory. This article focuses on extending SVM to multi-class classification with large training data set. In general it is utilized to solve binary classification problems and is more robust, less computationally expensive in comparison with linear classifiers. Multi-class SVM (MSVM) is used to classify samples belonging to more than two classes in a data set. A single MSVM can be considered as a group of binary SVMs, which are assumed to yield an output that assigns more weight to samples which belong to positive class and less weight to samples from negative class. The binary classifiers can be built as One-against-all (OvA) [3] and One-versus-one (OvO) [4].

The rest of this paper is organized as follows. Section 2 presents our methods and discusses two algorithms for identifying extreme points from a convex hull and our learning algorithms for learning large data set using the skin of reduced convex hulls. Section 3 presents our experimental results using synthetic data, real world benchmark data sets, and our capsule endoscopy videos. Section 4 concludes this paper with summary and our future extension.

2 Methodology

Geometric SVM represents two classes as convex hulls and solves the problem by finding the minimum distance between the two [5]. Given a set $X = \{x_1, x_2, \ldots, x_n\}$, the function ϕ maps each instance into the feature space, i.e., $\phi(x_i)$. For simplicity, we use ϕ_i to denote $\phi(x_i)$ and the mapped points form a feature set $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$. The convex hull, $C(\Phi)$, is hence a linear combination of all the instances in Φ:

$$C(\Phi) = \left\{ \sum_{i=1}^{k} \alpha_i \phi_i |\ \phi_i \in \Phi,\ 0 \leq \alpha_i \leq 1,\ \sum_{i=1}^{k} \alpha_i = 1 \right\} \tag{1}$$

To address linearly non-separable classes, soft convex hulls [6] (or Reduced Convex Hull (RCH) [7, 8]) was proposed. The RCH, $R(\Phi, \mu)$, is the set of convex combinations of instances in Φ with α_i bounded by a μ, $\mu \leq 1$. Following convex hull formula, an RCH is expressed as follows:

$$R(\Phi, \mu) = \left\{ \sum_{i=1}^{k} \alpha_i \phi_i |\ \phi_i \in \Phi,\ 0 \leq \alpha_i \leq \mu,\ \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{2}$$

By selecting appropriate μ for each class, a linearly non-separable problem can be transformed to a linearly-separable case [8]. The decision boundary is then perpendicular to the nearest points between the two RCHs derived from training samples. An example

is shown in Figure 1. Figure 1(a) and (b) illustrate convex hulls of two linearly non-separable classes and the separating hyper plane. Figure 1(c) depicts RCHs that retract for form the linearly separable case.

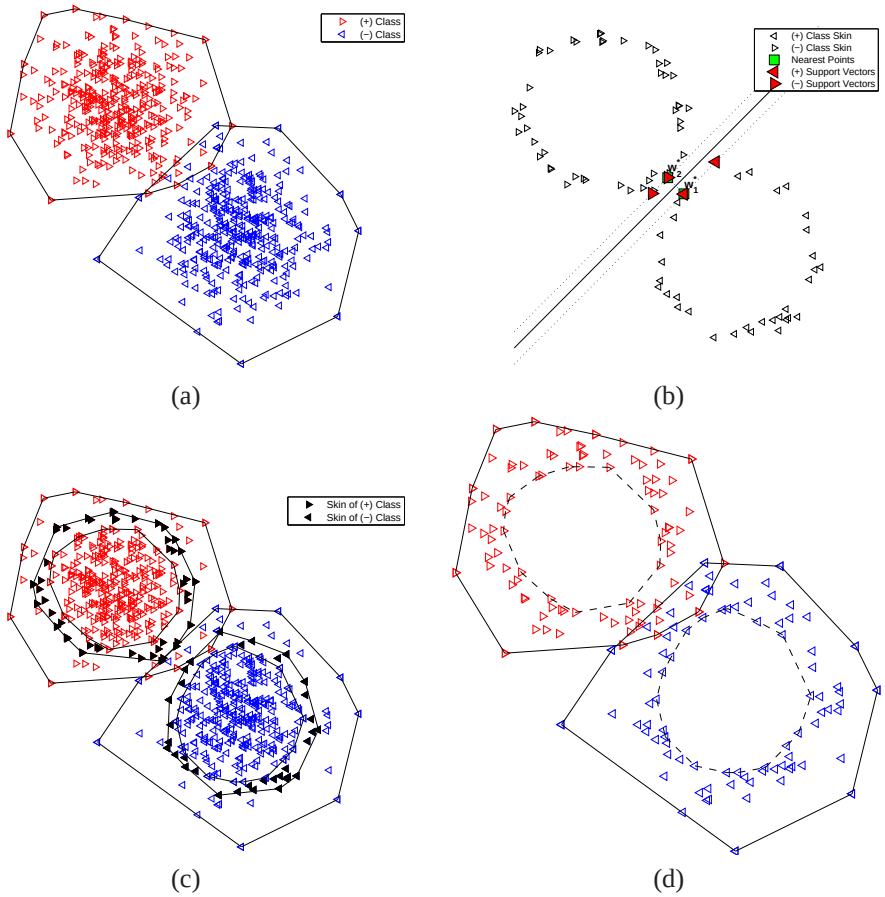


Figure 1 (a) Two linearly non-separable classes consisting of 700 data points. (b) Decision boundary is found by finding the nearest points between the $R(\mu = 0.5)$ of both classes. (c) The classes are linearly separable using $R(\mu = 0.5)$. (d) The $S(\Phi_i, 0.5, 1)$ is retained for future iterations.

To overcome high computational demands from large data set, we propose to identify and employ a subset of samples in the training process. Our method extends the RCHs and defines the approximate skin segments of convex hulls. The intuition is that only the samples within the skin are retained in training. When additional samples become available, they will be used together with the skin of the convex hull constructed from previous data set. Therefore, a much less number of instances is used in the training process. On the other hand, we need to avoid possible exclusion of future support vectors (SVs). Although no theoretical proof is given in this paper, with convex hulls constructed in the feature space, the skin is essentially a superset of the possible SVs.

Given an upper bound μ_u and a lower bound μ_l for α_i, $0 < \mu_l < \mu_u \leq 1$, the skin, $S(\Phi, \mu_l, \mu_u)$, of convex hull $C(\Phi)$ is the set of instances between two RCHs and can be expressed as follows:

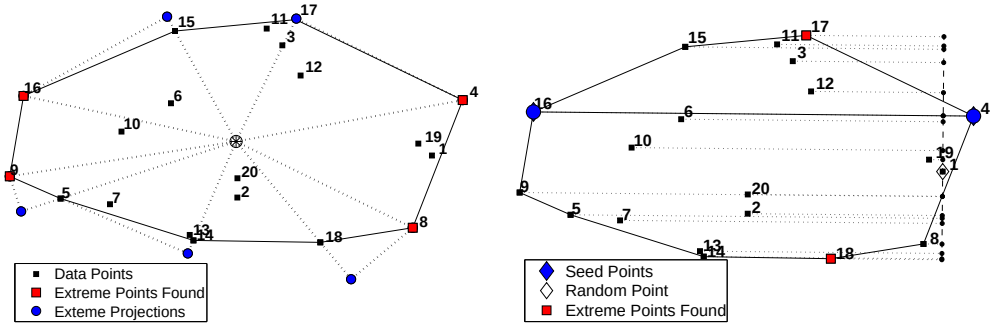
$$S(\Phi, \mu_l, \mu_u) = \{\phi_i | \phi_i \in \{R(\Phi, \mu_u) - R(\Phi, \mu_l)\}. \tag{3}$$

Finding the skin of convex hull, however, is challenging due to the lack of knowledge of the data distribution. We propose a recursive method that finds the vertices (i.e., extreme points) of a convex hull, which are used to represents the skin.

The projection of a vector ϕ_k to a direction $d = \phi_b - \phi_a$ is defined as the inner product of the two difference vectors with respect to ϕ_a (the reference vector):

$$P(\phi_k, d) = \langle \phi_k - \phi_a, d \rangle. \tag{4}$$

The explicit expression of feature vectors ϕ_i is not required to compute the extreme points. The projection $P(\phi_n, d(\phi_m))$ in the feature space can be achieved with the kernel operation.



(a) (b)

Figure 2 Finding extreme points using center of gravity.

Intuitively, evaluating the projection magnitude of all data points in all possible directions finds the complete set of extreme points. It is computationally infeasible given a large data set. An alternative approach is to evaluate the data point projection in finite number of directions. Our method finds the extreme points in two steps: 1) a set of seed points are identified based on the center of gravity; and 2) the complete set of extreme points are then found via recursively searching along the direction defined by a pair of extreme points.

A set of seed extreme points are found using the gravity center. Using gravity center could miss some less prominent extreme points following the above procedure. An example is illustrated in Figure 2(a). The solid squares denote the data samples and the gravity center is marked with a large circle. The projected vectors are marked with solid dots. Using our method, four extreme points are identified and highlighted with solid squares. For instance, point 16 is identified as an extreme point since it gives the

Algorithm 1 A recursive method for finding extreme points. Probing(Φ', d, ϕ_p, ϕ_q)

Input: $\Phi' \subseteq \Phi$, d, ϕ_p, and ϕ_q
Output: set of extreme points F

$F \leftarrow \emptyset$
Randomly select $\phi_m \in \Phi'$ and $m \neq p, m \neq q$
if $\Phi' \neq \emptyset$ **then**
  Identify probing direction d^* using Eq. (5)
  if $\langle d^*, d \rangle < 0$ **then**
    $d^* \leftarrow -d^*$
  end if
  $d^* \leftarrow \frac{d'^*}{||d^*||}$
  $F \leftarrow F \bigcup \{\phi_e | \phi_e = \arg\max_{\phi_k \in \Phi'} P(\phi_k, d^*)\}$
  for all $\phi_i \in \Phi'$ **do**
    if $P(\phi_i, d^*) > 0$ **then**
      $\Phi'' \leftarrow \Phi'' \bigcup x_i$
    end if
  end for

  $F \leftarrow F \bigcup$ Probing$(\Phi'', d, \phi_p, \phi_e)$
  $F \leftarrow F \bigcup$ Probing$(\Phi'', d, \phi_q, \phi_e)$
end if
return F

greatest projection to $d(x_{16}, \bar{X})$ (as well as $d(x_{15}, \bar{X})$). However, instances 14, 15, 17, and 18 are the extreme points but are missed by the process.

Provided with a set of seed extreme points, our algorithm recursively searches along the perpendicular directions of the convex hull boundaries, which is lists in Algorithm 1. The recursive steps start with randomly selected two seed extreme points ϕ_p, ϕ_q and another instance $\phi_m \in \Phi$. Search for extreme points is performed in the direction perpendicular to the difference vector of ϕ_p, ϕ_q, through ϕ_m. Vector $\phi_q - \phi_p$ splits the space into two halves. The perpendicular searching direction d^* can then be determined as follows:

$$d^* = \phi_m - \phi_p - P(\phi_m, d(\phi_p, \phi_q)) \frac{\phi_q - \phi_p}{||\phi_q - \phi_p||} \tag{5}$$

Searching in each half space is achieved recursively using a pair of identified extreme points, ϕ_p and ϕ_q. Let Φ' denote the instances in the half space. With a random instance ϕ_m in Φ', a probing direction, d^*. d^* points toward the outside of the convex hull; Otherwise, change its direction. Hence, an extreme point is identified in Φ'. ϕ_m is paired with ϕ_p and ϕ_q to split the feature space for further probing. The process stops when no additional points exist in Φ'.

Figure 2(b) illustrates an example of probing in a half space. The two extreme points are 9 and 14, which determines the probing direction. The dotted lines depict the projections of the instances. In two iterations, extreme points 1 and 12 are found.

The learning task is defined as $f : \mathbb{R}^N \rightarrow \{-1, +1\}$, using a training set $\{(x_1, y_1), \cdots (x_n, y_n)\}$, where $x_i \in \mathbb{R}^N$, $y_i \in \{-1, +1\}$. The two classes are denoted

with $\Phi_+ = \{\phi_i : y_i = 1\}$, $\Phi_- = \{\Phi_i : y_i = -1\}$. The quadratic and geometric solutions of SVM have been shown to be equivalent [9, 10, 6, 11, 5]. In geometric SVM, the two classes are represented by their convex hulls and finding solution is equivalent to identifying the nearest points of the two convex hulls. The optimal support vector machine is hence the hyperplane perpendicular to the connection of the pair of the nearest points of the two convex hulls.

The pair of nearest points from the two classes is denoted with (ϕ_+^*, ϕ_-^*) and satisfy the following criterion:

$$(\phi_+^*, \phi_-^*) = \arg \min_{\phi_+ \in \Phi_+, \phi_- \in \Phi_-} (||\phi_+ - \phi_-||)$$

where $\phi_+^* \in \Phi_+, \phi_-^* \in \Phi_-$, are found using the Gilbert's algorithm [12].

The steps at each incremental iteration to deal with linearly non-separable case is summarized in Algorithm 2. Note that due to unforeseen data distribution, skin of the same thickness for the distinct classes could enclose significantly different number of samples. Although the original data set is balanced, the selected ones could tilt the ratio and result in skewed training data set. The skewness of the training data set penalizes the minority class implicitly [13]. To avoid the sample size-induced bias, two separate skin parameters are determined by weighting using the ratio of number of samples from each class seen, i.e., μ_{u+}, μ_{u-}. A $\mu_l (< \mu_u)$ is selected and the skins $S(\Phi_i, \mu_{li}, \mu_{ui})$ $i = \{+, -\}$ (illustrated with filled triangles in Figure 1) are used to find the decision boundary.

Algorithm 2 Learning from large data set using RCHs.

1: Select μ_u
2: Construct linearly separable $R(\Phi_+, \mu_u)$ and $R(\Phi_-, \mu_u)$ using Algorithm 1
3: Select a μ_{l+}, $\mu_{l+} < \mu_{u+}$, for the positive class. Without loss of generality, we assume the positive class has less number of training examples.
4: Choose the μ_{l-} for the other class such that $|S_+| \approx |S_-|$
5: Construct inner RCHs $R(\Phi_+, \mu_{l+})$ and $R(\Phi_-, \mu_{l-})$
6: Retain samples in $S(\Phi_i, \mu_{l+}, \mu_{u+})$ and $S(\Phi_i, \mu_{l-}, \mu_{u-})$
7: Derive the classifier using Gilbert's algorithm [12]

3 Experiments

3.1 Data Preparation and Implementation

To analyze our method, we prepared a 2-D synthetic data set so that the final classifier boundary can be visualized. Examples of three classes are sampled from Gaussian functions with different means and variances.

Six sets of real world data are used, within which five sets are obtained from the UCI Machine Learning Repository [14] and a mammography data set [15] is included. Table 1 lists the properties of these real world data sets.

In our experiments, we use hierarchical SVM to achieve multi-class classification. In the first level one-against-all for all classes is performed to find the class which

Table 1 Properties of our experimental datasets.

Datasets	Dim.	Positive Class	Sample Size	
			+ Class	- Class
Iris	4	Setosa	50	100
SPECT	22	1	212	55
Pima	8	1	268	500
Yeast	8	CYT	463	1,021
Ionosphere	34	b	126	225
Mammography	6	2	260	10,923

discriminates from the rest. The first level classifier is used to classify the class identified. In the next level the one-against-all for remaining classes is repeated to find the class which discriminates the rest of the remaining classes. This is repeated until all classes can be classified.

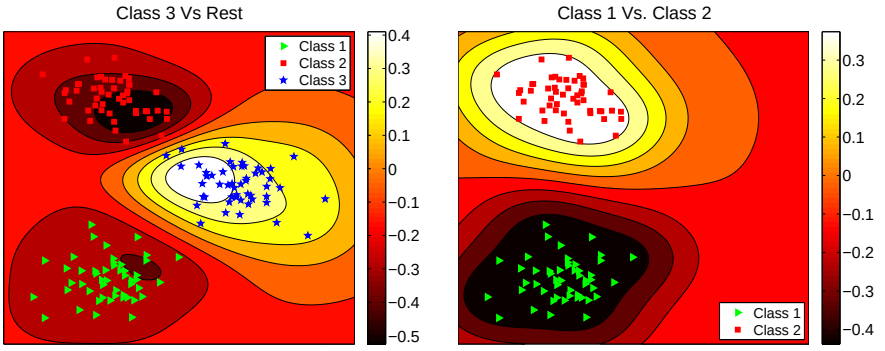
The performance of our methods was evaluated using sensitivity S, specificity P and accuracy A. Let TP, TN, FP, FN denote true positive, true negative, false positive, and false negative counts, respectively, in the testing results. The three metrics are computed as follows:

$$S = \frac{TP}{TP + FN} \tag{6}$$

$$P = \frac{TN}{TN + FP}$$

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

3.2 Experiments with Synthetic and Real World Data Sets



(a) (b)

Figure 3 Results of hierarchical multi class classification. Contours of decision boundaries of (a) Class 3 vs. Class 1 and 2 and (b) Class 1 vs. class 2.

Figure 3 illustrates the contour plots of decision boundary trained with synthetic data sets of three classes. The color depicts the distance to the center of each class.

Two classifiers were trained to partition the space into three parts that correspond to the class distribution. The decision boundary for this example is denoted with the zero level contour. Figure 3(a) illustrates the decision boundary of class 3 verses classes 1 and 2, whereas Figure 3(b) illustrates the decision boundary of classes 1 and 2. The 2D contour plots of the classifier demonstrate that our method predicts class labels successfully in the case of multi-class classification.

Experiments on UCI data sets were performed to evaluate the general applicabilit of our learning method in a binary classification setting. Each feature in a dataset was normalized with the mean value and the range of that feature. That is, each component of an instance is in the range of $[0, 1]$.

Figure 4 illustrates the classifier performance during the learning iterations. Ten (10) repetitions were conducted with random initial training set. For each dataset, 50% of the data were randomly selected and used for training. The remaining data were used as the test set. In each case, a SVM classifier was created using all the training data. The best parameters were selected based on their generalization performance with the testing data set. The results from these classifiers are used as reference and are depicted as horizontal lines in Figure 4.

In our learning process, 10 samples were randomly selected from each class of the training set and a SVM is trained. In each incremental step, 10 randomly selected samples from the remaining training dataset were used to update the classifier. The intermediate classifiers were evaluated with the test dataset. For each dataset, 10 repetitions were conducted and the average performance is plotted as a line in Figure 4, the vertical line segments depicts the variance of performance in each incremental step.

With more examples included in the training process, the classifier trained with our method improves its performance. It is evident in the cases of Yeast, SPECT, Pima, and Ionosphere. In the cases of Iris and Mammography, the performance at the very beginning is already superior and there is not much of space for improvement. Hence, the change of performance in the following iterations is very trivial. However, improvement in sensitivity can still be observed in the training using Mammography dataset and by the end of iterations, classifier outperformed the batch learning by a small margin.

Despite a slightly drop of specificity of the SPECT dataset, the SVMs trained with our method achieved the same performance as or even outperformed the batch learning method. It is interesting that in five cases (except Iris), the intermediate classifier had a degradation in early iterations, but the training process was able to recover to the benchmark performance asymptotically as additional data instances are included for training.

3.3 Experiments with Capsule Endoscopy Videos

The analysis tool provided by the manufacture of the Pillcam capsule endoscopy plots the trace of device throughout the digestive tract based on the wireless signal strength to the external image downloader carried by the patient. Despite its large error margin, it requires manually annotation by medical specialist the frame of pylorus, where the capsule leaves the stomach and enters the intestine, and the frame of ileocecal valve, between the intestine and colon. Our experiments on CE videos were performed to automate the classification of the frames in CE videos into four natural digestive organs, namely esophagus, stomach, small intestine, and colon.

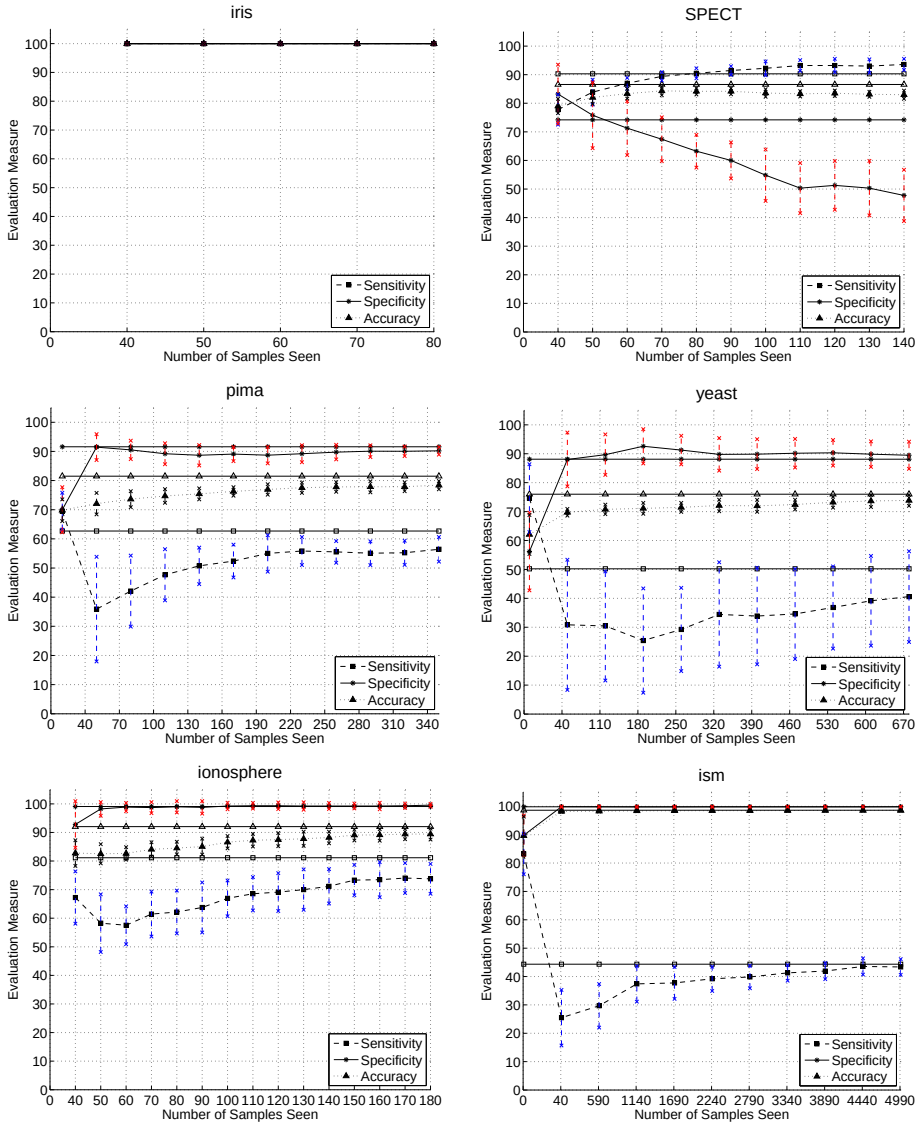


Figure 4 Results of learning error using real world data sets from the UCI repository. The vertical bars depict the variance around the mean value.

We have collected 6 CE videos that are manually annotated by a gastroenterologist. Each video consists of approximately 55,000 frames that are extracted and saved as JPEG images. Out of the 6 videos, we randomly selected one video to train, leaving the remaining 5 for testing and evaluation.

In our previous experiments with CE videos, we found that HSV color space gives better classification performance on average [16]. In addition, using histogram significantly reduces the dimensionality (Each frame is a 256$\times$ 256 color image. If pixel color is used, the dimensionality of each instance is up to 196,608.) Hence, we

adopted the color histogram in the HSV space as features. The color histogram is very large and sparse matrix as shown in Figure 5. With n bins used in each color component, there are n^3 features using HSV histogram for every video frame, most of which are zeros or close to zeros. To suppress sparseness and the number of values in features, only the hue and saturation (HS) components were used. As observed in our previous experiments [16, 17], an advantage of using HS components is improved robustness in handling lighting variations in the GI tract. As shown in Figure 5, the 2D space spanned by HS components is dominated with small values. Hence, a minimum bounding rectangle region of the HS space with non zero values was identified from the training images. Only the values within the rectangular region of HS histogram were used as features for our classification.

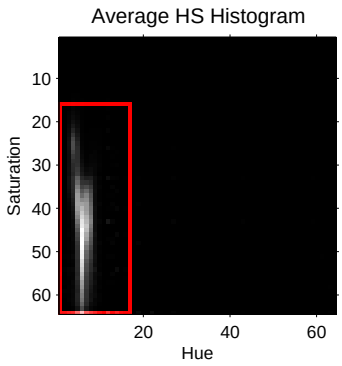


Figure 5 The average HS histogram of the CE video used in training. The rectangle denotes the color space used in our learning and classification processes.

The order of classification of multi-class SVM was determined based on the preliminary classification evaluation. In our experiments, identification of esophagus gives the best accuracy followed by the identification of small intestine. Hence, the order is determined and listed in Table 2. The kernels used to train SVM are also included in this table.

Table 2 Order and parameters of hierarchical classification of organs of CE videos

Order	Dividing classes	Kernel
1	Esophagus vs. Rest	RBF($\sigma = 0.15$)
2	Small intestine vs. Stomach and Colon	RBF($\sigma = 0.1$)
3	Stomach vs. Colon	RBF($\sigma = 0.5$)

In the learning process, 50 frames were randomly selected from each class of the training video to train a SVM. In each incremental step, randomly selected 20 frames from the remaining training video frames were used to update the classifier. The iteration repeats until the training examples exhaust. Table 3 lists the results of the final classifiers tested using the test videos. The performance of our method is highly satisfactory. With the majority of frames acquired in stomach and small intestine, the

average accuracies are 86.9% and 94.4%, respectively. Images acquired in colon are disturbed with noise from feces and fluid, which results in large number of dark images and causes performance drop in the classification accuracy.

Table 3 Classification performance of digestive organs in CE videos

Video	Esophagus	Stomach	Small Intestine	Colon
1	100.0%	87.6%	94.2%	85.3%
2	94.4%	85.8%	95.3%	82.2%
3	95.0%	87.2%	94.7%	84.3%
4	100.0%	86.4%	94.1%	83.7%
5	90.0%	87.7%	93.9%	94.3%

At the end of incremental training only 12% of the frames were part of the skins among the four classes for the hierarchical SVMs. Apparently, the smaller number of examples demands much less memory space for learning process and, hence, provides a plausible mechanism for handling large amount of data set. When new samples are added, the classifier is updated efficiently in contrast to the conventional batch learning methods.

4 Conclusion

In this paper we presented an incremental SVM to learn from large data set with emerging trend and dynamic patterns. To overcome high computational demands from large data set, we develop a method to identify and employ a subset of samples in the training process. Our method extends the reduced convex hull concept and defines the approximate skin segments of convex hulls. The intuition is that only the samples within the skin are retained in training. When additional samples become available, they will be used together with the skin of the convex hull constructed from previous data set. Therefore, a much less number of instances is used in the training process.

Experiments were conducted using a synthetic 2D data set, six real world data sets from UCI repository, and six CE videos. Our results demonstrated highly competitive performance that requires much less resource. Based on our experimental results, the following conclusions can be drawn.

- With more examples included in the training process, the classifier trained with our method improves its performance.
- The training process was able to recover from an intermediate performance degradation when additional instances are included for training.
- The average performance of classifying CE video is above 86.9%, which is very competitive.
- The amount of memory space required in the training process could be one eighth of what is required by the conventional SVM, which cast new light on processing large data set within constrained resource.

The accuracy for our CE video segmentation could be further improved if temporal information is utilized. This method can be easily extended for other temporal signal classification with large data size and non-stationary pattern.

References

[1] J. P. S. Cunha, M. Coimbra, P. Campos, and J. M. Soares. Automated topographic segmentation and transit time estimation in endoscopic capsule exams. *IEEE Transactions on Medical Imaging*, 27(1):19–27, Jan 2008.

[2] M. Mackiewicz, J. Berens, and M. Fisher. Wireless capsule endoscopy color video segmentation. *IEEE Transactions on Medical Imaging*, 27(12):1769–1781, Dec 2008.

[3] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, Berlin, 1999.

[4] Jerome H. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[5] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11(1):124–136, Jan 2000.

[6] D. Crisp and C. Burges. A geometric interpretation of ν-SVM classifiers. In *Proceedings of Advances in Neural Information Processing Systems*, pages 244–250, 1999.

[7] M.E. Mavroforakis and S. Theodoridis. A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, May 2006.

[8] M.E. Mavroforakis, M. Sdralis, and S. Theodoridis. A geometric nearest point algorithm for the efficient solution of the SVM classification task. *IEEE Transactions on Neural Networks*, 18(5):1545–1549, Sept. 2007.

[9] P. Chen, C. Lin, and B. Schölkopf. A tutorial on ν-support vector machines:. *Appl. Stoch. Model. Bus. Ind.*, 21(2):111–136, 2005.

[10] K. P. Bennett and C. Campbell. Support vector machines: hype or hallelujah? *SIGKDD Explor. Newsl.*, 2(2):1–13, 2000.

[11] D. Zhou, B. Xiao, H. Zhou, and R. Dai. Global geometry of SVM classifiers. Technical report, Institute of Automation, Chinese Academy of Science, 2001.

[12] E.G. Gilbert. An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM Journal of Control*, 4(1):61–80, 1966.

[13] B. Giritharan, X. Yuan, and J. Liu. Resampling method for balancing training data in video analysis. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Medical Imaging*, volume 7624, page 762403, Feb 2010.

[14] A. Asuncion and D.J. Newman. UCI machine learning repository at URL: http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research*, 16:321–357, 2002.

[16] B. Giritharan, X. Yuan, J. Liu, B.Buckle, J.Oh, and S.J. Tang. Bleeding detection from capsule endoscopy videos. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4780–4783, Vancouver, BC, Aug. 2008.

[17] J. Liu and X. Yuan. Obscure bleeding detection in endoscopy images using support vector machines. *Optimization and Engineering*, 10(2):289–299, 2009.