# Learning with Cascade for Classification of Non-Convex Manifolds

Xiangsheng Huang
Institute of Automation
Chinese Academy of Science
Beijing, China 100080

Stan Z. Li
Microsoft Research Asia
Beijing, China 100080

Yangsheng Wang
Institute of Automation
Chinese Academy of Science
Beijing, China 100080

## Abstract

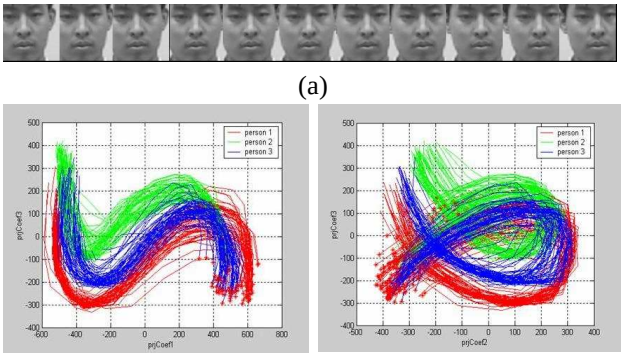
*Images of a visual object, such as human face, reside in a complicated manifold in the high dimensional image space, when the object is subject to variations in pose, illumination, and other factors. Viola and Jones [1, 2, 3] have successfully tackled difficult nonlinear classification problem for face detection using AdaBoost learning. Moreover, their simple-to-complex cascade of classifiers structure makes the learning and classification even more effective. While training with cascade has been used effectively in many works [4, 5, 6, 7, 2, 3, 8, 9, 10], an understanding of the role of the cascade strategy is still lacking.*

*In this paper, we analyze the problem of classifying non-convex manifolds using AdaBoost learning with and without using cascade. We explain that the divide-and-conquer strategy in cascade learning has a great contribution on learning a complex classifier for non-convex manifolds. We prove that AdaBoost learning with cascade is effective when a complete or over-complete set of features (or weak classifiers) is available. Experiments with both synthesized and real data demonstrate that AdaBoost learning with cascade leads to improved convergence and accuracy.*

## 1. Introduction

An interesting problem in pattern recognition and computer vision is how to model and classify images of visual objects, such as the human face, under extrinsic variations in photometry and geometry. It has been found that distributions of images in low dimensional linear subspaces such as those based on principle component analysis (PCA) under perceivable variations in viewpoint, illumination are highly nonlinear, non-convex, complex, and perhaps twisted as shown in Fig 1. Linear methods, such as PCA, de-correlate the low order moments while the imaging process is a highly nonlinear function of various factors. They can hardly remove extrinsic variations in order to achieve high recognition rate for identifying the intrinsic identities of objects.

Recently, several nonlinear approaches, including ISOMAP [11], locally linear embedding (LLE) [12], and




(a) (b) (c)

Figure 1: PCA subspace of translated faces. (a) translated faces, (b) the result of translated faces projected onto the first and the third dimension, (c) the result of translated faces projected onto the second and the third dimension

Laplacian Eigenmaps (LEM) [13], have emerged as nonlinear methods for modelling nonlinear manifolds of data distributions. ISOMAP performs nonlinear dimensionality reduction by applying multi-dimensional scaling (MDS) on the geodesic distance matrix. LLE represents nonlinear manifold using multilocally linear PCA representations. LEM attempts to maintain ordering between points in a local embedding. But all these methods are computational complex, and could not ensure the separability of these manifolds after dimensionality reducing.

Another major advance in nonlinear classification is AdaBoost algorithms, introduced by Freund and Schapire [14]. These provide a simple yet effective stagewise learning approach: It learns a sequence of more easily learnable weak classifiers, and boosts them into a single strong classifier by a linear combination of them.

Originating from the PAC (probably approximately correct) learning theory [15, 16], AdaBoost provably achieves arbitrarily good bounds on its training and generalization errors [14, 17] provided that weak classifiers can perform slightly better than random guessing on every distribution over the training set. It is also shown that such simple weak classifiers, when boosted, can capture complex deci-

sion boundaries [18].

Relationships of AdaBoost to functional optimization and statistical estimation are established recently. It is shown that the AdaBoost learning procedure minimizes an upper error bound which is an exponential function of the margin on the training set [19]. Several gradient boosting algorithms are proposed [20, 21, 22], which provides new insights into AdaBoost learning. A significant advance is made by Friedman *et al.* [23]. It is shown that the AdaBoost algorithms can be interpreted as stagewise estimation procedures that fit an additive logistical regression model. Both the discrete AdaBoost [14] and the real version [17] optimize an exponential loss function, albeit in different ways. The work [23] links AdaBoost, which was advocated from the machine learning viewpoint, to the statistical theory.

Viola and Jones [1, 2, 3] have made a successful application of AdaBoost to face detection. Moreover, their simple-to-complex cascade of classifiers structure makes the computation even more efficient. Their system is the first real-time frontal-view face detector which runs at about 14 frame per second for a 320x240 image [1]. Asymmetric Boost is a variation of Adaboost for dealing with asymmetric, skewed distributions [3]. The original AdaBoost minimizes a quantity related to classification error; it does not minimize the number of false negatives.

A recent algorithm of cascade type classifier is the maximal rejection [24] for yes-or-no type of pattern classification. A cascade of individual LDA classifiers is constructed, in which negative training examples are extracted by bootstraping or successive rejection operations. However, it has only a single linear classifier for each stage. Therefore it can not reject negative examples inside the hull of the positive set and so it is impossible to obtain the zero error rate on the training set and hence test set if the manifold is non-convex.

Many researchers have used cascade of classifiers [4, 5, 6, 7, 2, 3, 8, 9, 10]. It is found that the cascade structure not only increases the speed of classification by focusing attention on promising regions of the image, but also make training easier. However, a detailed analysis of the role of cascade is lacking.

In this paper, the problem of classifying non-convex manifolds using AdaBoost learning is analyzed. We explain the advantage of the cascade structure from the view of the divide-and-conquer strategy. It is the divide-and-conquer strategy that has a great contribution on learning a complex classifier for non-convex manifolds. Training a single strong classifier without using cascade may not converge to a low error rate and may lead to many undesirable weak classifiers with low efficiency. In contrast, the cascade strategy splits a hard problem into several easier subproblems, and solves them one by one in cascade. Experiments with synthesized data and with real data demonstrate that AdaBoost learning with the cascade structure leads to greater performance.

The rest of the paper is organized as follows: Section 2 presents problem formulation and motivation. The divide-conquer strategy is analyzed in Section 3. Section 4 provides experiment results.

## 2. Problem Analysis

We focus on learning for two class (positive and negative) problem. Provided that we are given a large training set $L$ of $N$ labelled training examples $(x_1, y_1), \ldots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example $x_i \in R^n$. $N$ is typically of the magnitude of millions, or even infinity. Our aim is to learn a classifier who can output a class probability estimate function $P\{y|x\}$.

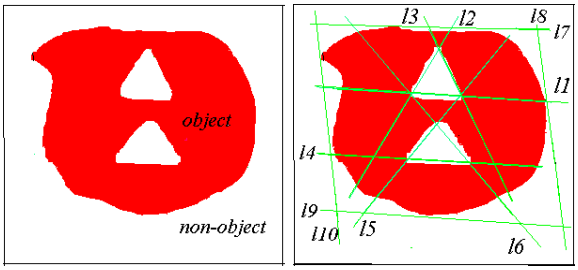
Due to limitation on computational resources, in practice, we could use just a subset of training data to train a single strong classifier in each stage. The size of the training set is restricted to $n$, where $n \ll N$.

Two important questions need be answered: how to select new training sets for each stage, and how to combine the strong classifiers. In order to find the solutions of these questions, a more important factor, the non-convexity and complexity of manifold should be considered.

Although Freund has proved that the error of AdaBoost is bounded above by a exponential function [14], in practice, many weak classifiers, which conflict with each other, may be learned. In many cases, training a single strong classifier without using cascade can hardly converge to a required performance specification, and instead it could lead to many undesirable weak classifiers with low efficiency.

Fig 2(a) gives an illustrative situation for a two class problem. The red (darker) area represents the positive samples, whereas the white areas stand for the negative samples. Note there are two holes of negative inside the red area. In order to reject all white areas, ten simple features are learned to form a strong classifier as in Fig 2(b). The first feature is line $l1$, and the area above $l1$ is regard as negative examples. The second feature is $l2$, and the area right-bottom to $l2$ is considered as negative examples, and so on. If the two holes (negative samples) want to be excluded without using cascade, the six features ($l1$, $l2$, $l3$, $l4$, $l5$ and $l6$) are required to be satisfied at one time. However, these features conflict with each other; and no area (examples) exists to satisfy these features. In order to reduce this conflict and reject all negative examples, a straightforward AdaBoost may learn a strong classifier consisting many undesirable weak classifiers and the weak classifiers at the end usually are of low efficiency.

In contrast, the cascade structure is a divide-conquer strategy, which slices off part of negative samples stage by stage, while fixing positive samples. Namely, the cascade strategy splits a hard problem into several easier subprob-




(a) (b)

Figure 2: Non-convex manifolds (a) and their "separation" (b)

lems, and solves them one by one. Therefore, a cascade of strong classifiers does not require so many weak classifications to be satisfied at one time and this makes the training easier. As shown in Fig 2(b), if we use one stage to reject one hole, then only three features are required to satisfied at one time, ($l1$, $l2$ and $l3$) or ($l4$, $l5$ and $l6$). Obviously, it is easier for the cascade type to achieve a good classification than that of a single strong classifier.

## 3. Learning Nonconvex Classifier Using Cascade

Due to the non-convexity and complexity of the two manifolds, it is hard to separate these manifolds using a single strong classifier. The cascade training is aimed to overcome this problem. This is illustrated in Fig 3. Green examples represent positive samples, while red examples represent negative samples. We fix all the positive examples at each stage, and bootstrap $n$ negative examples which are misclassified by the previous stages. This procedure is sketched in Fig 3(a)-(j). Fig 4 illustrates a cascade training process.

Polytopes which enclose some negative samples for the rejection are requires to handle non-convex manifolds. The following theorem proves that such polytopes exist.

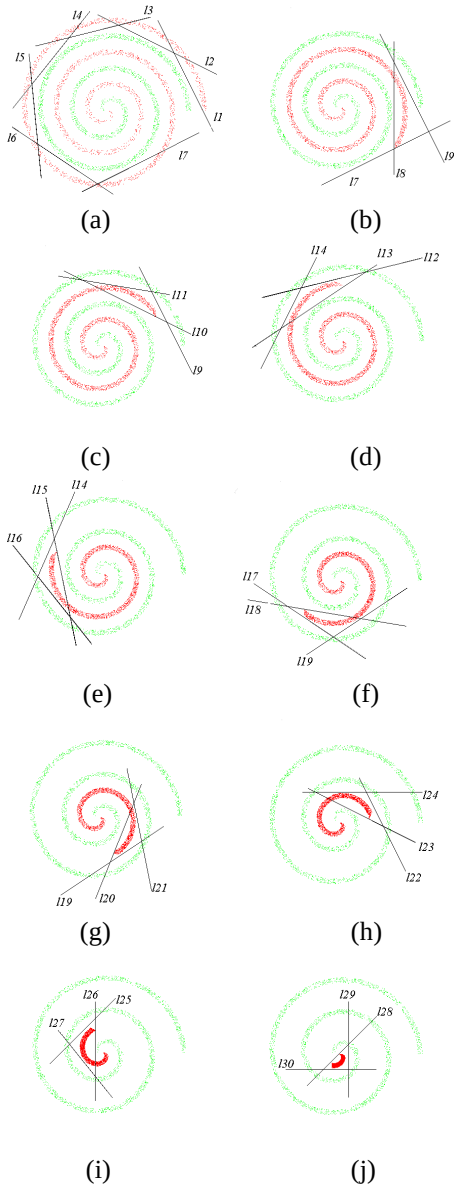
**Theorem:** In an $n$ dimensional Euclidean space, $n+1$ hyperplanes can be found to construct a polytope (a finite region of $n$-dimensional space enclosed by a finite number of hyperplanes) with arbitrary size.

**Proof:**

(1) Obviously, an arbitrary segment can be bounded by two points, $p_0$ and $p_1$ in one dimensional case, as Fig 5 (a) shows.

(2) In two dimensional case, $p_2$ is on the second dimension, as Fig 5 (b) shows. The first line can be defined by point $p_2$ and point $p_0$. The second line can be defined by point $p_2$ and point $p_1$. The last line can be defined by point $p_0$ and point $p_1$. So, a triangle with arbitrary size can be construct by these three lines.

(3) In three dimensional Euclidean space, $p_3$ is on the




(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3: The procedure of two manifolds separation using cascade.

---

1. For stage $s$=1 to $S$
   a)Generate a bootstrap set $B$ with size $n$ ;
   b)Train a strong classifier $f_s$ with $B$;
2. Output the final classifier $g(f_1, ...f_S)$, a combination of $f_i$'s.

---

Figure 4: Training cascade of strong classifiers.

third dimension, as Fig 5 (c) shows. The first plane can be defined by $p_3$, $p_0$ and $p_1$. The second plane can be defined by $p_3$, $p_0$ and $p_2$. The third plane can be defined by $p_3$, $p_1$ and $p_2$. The last plane can be defined by $p_0$, $p_1$ and $p_2$. So,

a tetrahedron with arbitrary size can be bounded by these four planes (plane $p_3p_0p_1$, plane $p_3p_0p_2$, plane $p_3p_1p_2$ and plane $p_0p_1p_2$).

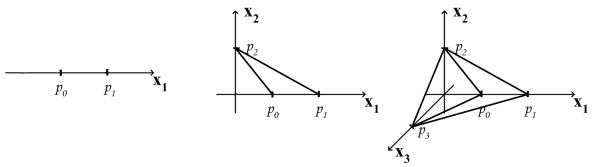
(4) In four dimensional Euclidean space, $p_4$ is on the fourth dimension. The first hyperplane can be defined by $p_4$, $p_0$, $p_1$ and $p_2$. The second hyperplane can be defined by $p_4$, $p_0$, $p_1$and $p_3$. The third hyperplane can be defined by $p_4$, $p_0$, $p_2$ and $p_3$. The fourth hyperplane can be defined by $p_4$, $p_1$, $p_2$ and $p_3$. The last hyperplane can be defined by $p_0$, $p_1$, $p_2$, and $p_3$. So, a polychoron with arbitrary size can be bounded by these five hyperplanes.

(5) Given that $n$ hyperplanes can be found to construct a polytope with arbitrary size in $n-1$ dimensional Euclidean space.

(6) In $n$ dimensional Euclidean space, $p_n$ is on the $n$th dimension. The first hyperplane can be defined by $p_n$, $p_0$, $p_1$, ..., $p_{n-2}$. The second hyperplane can be defined by $p_n$, $p_0$, $p_1$, ..., $p_{n-3}$, $p_{n-1}$, and so on. We can use $p_n$ and the combination of $p_0$, ..., $p_{n-1}$ to construct $n$ hyperplanes. The last hyperplane can be defined by $p_0$, $p_1$, ..., $p_{n-1}$. So, a polytope with arbitrary size can be constructed by these $n+1$ hyperplanes.

As discussed above, $n+1$ hyperplanes can be found to construct a polytope with arbitrary size in an $n$ dimensional Euclidean space.

**End Proof**




(a) (b) (c)

Figure 5: Three examples of bounded region

Training in cascade is a divide-conquer strategy. As the theorem indicates, no matter how complex the manifold is, we can find a polytope to bound some negative examples, and reject them in stages. Therefore, a nonconvex classifier consisting of several linear classifiers can always be found to reject part of negative manifold in all cases, as long as feature space is complete or over-complete.

While maximal rejection classifier (MRC) [24] can also be regarded as cascade. However, it has only a single linear classifier for each stage, and is almost impossible to form a ploytope in each stage. Therefore it can not reject negative examples inside the hull of the positive sample set and so it is impossible to obtain the zero error rate on the nonconvex training set of positive examples and hence test set if the manifold is non-convex.

# 4. Experimental Results

Two experiments are performed to compare AdaBoost tarining with cascade and with non-cascade. The first is based on synthesized data sets; the other is a real application for face alignment quality evaluation.

## 4.1. Experiment on 2D artificial data sets

An synthesized data set was generated with positive samples encompassed by negative samples, as shown in Fig 6(a). Red points represent positive examples, and green points stand for negative examples respectively. Those examples which have been correctly classified *at the end of each training stage* are re-labeled in black color. The error rate curves as functions of number of weak classifiers is shown in Fig 7.

From Fig 6(b) and the red curve in Fig 7, we see that the AdaBoost learning without cascade can barely achieve high accuracy; the correctly classification rate it can get is about 97% even after 1300 iterations, and the training accuracy can hardly be improved after that. In this case, after conquering those big block parts of the negative examples, the single stage Adaboost learning falls into a dilemma: it have to cut part of negative examples inside the hull of posirive data set out, which conflicts with some weak classifiers learned before. In contrast, the cascade strategy tends to overcome this problem, as shown in Fig 6(c)-(g)and the blue curve in Fig 7. We see that after 500 iterations, learning with cascade converges more quickly than without. It is more flexible for cascade to handle non-convex manifold.

## 4.2. Face Alignment Quality Evaluation

This experiment compares the two scheme using real data derived for face alignment evaluation. The motivation for this application is the following: Alignment between the input and target objects has great impact on the performance of image analysis and recognition system. Active Shape Models (ASM)[25] and Active Appearance Models (AAM) [26, 27] provide an important framework for alignment. However, an effective method for the evaluation of ASM/AAM alignment results has been lacking for classifying between qualified and un-qualified alignment (see Fig 8). Bad alignment results, can drop system performance. An evaluation method is need to accept or reject an alignment result.

In this application, the positive training examples are qualified alignment results and negative are un-qualified alignment results, as shown in Fig 9. They are generated as follows: Examples of good and bad alignment are collected. All the shapes are aligned or warping to the tangent space of the mean shape $\overline{S}$. After that, the texture $T_0$ is warped correspondingly to $T \in R^L$ where $L$ is the number of pixels in the mean shape $\overline{S}$.

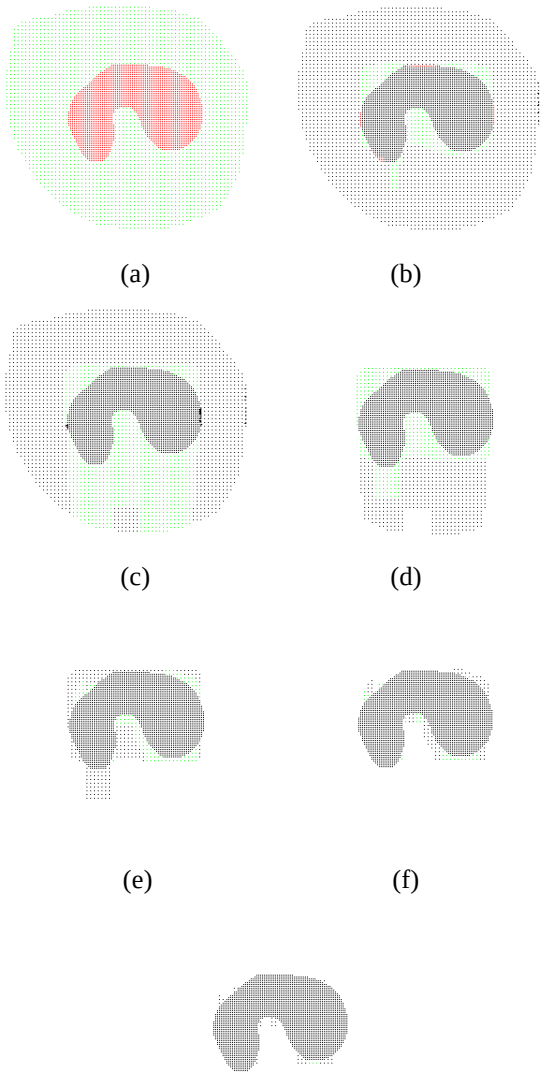


Figure 6: Comparison between Cascade and Non-cascade: (a) positive and negative manifolds; (b) result of learning without cascade with 1300 weak classifiers; (c)-(g) results at the end of each of the five training stages with cascade, where the total numbers of weak classifiers are 1300

In this experiment, 2536 positive examples and 3000 negative examples are used to train a strong classifier. The 2536 positive examples are derived from 1268 original positive examples plus the mirror images. The negative examples are generated by random rotating, scaling, shifting positive examples' shape points. A strong classifier is trained to reject $92\%$ negative examples, while correctly accepting $100\%$ of positive examples.

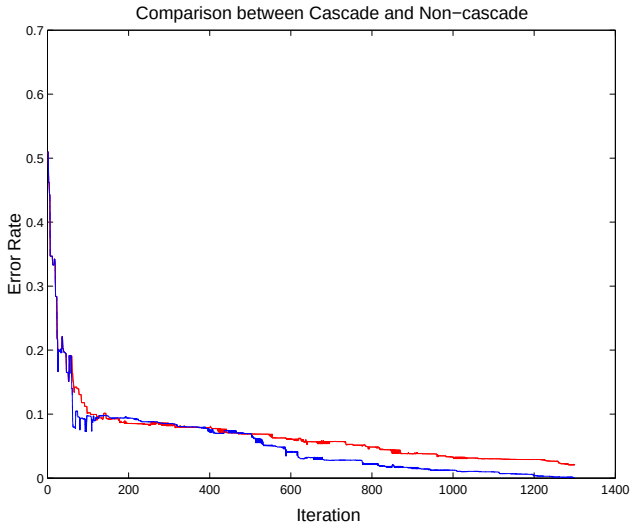



Figure 7: Error rate curves as functions of number of weak classifiers for AdaBoost learning with (blue curve) and without (red curve) cascade.

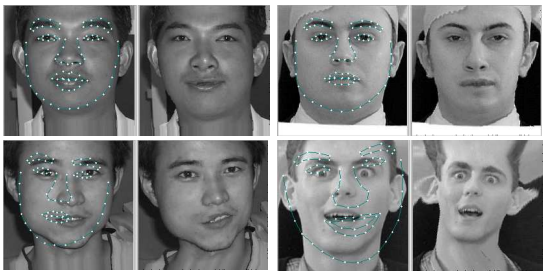


Figure 8: Instances of qualified (top) and un-qualified (bottom) examples and the images warped to the mean face according to the alignment results. The warped face looks strange when the alignment is no good.

A cascade of classifiers is trained to obtain a computational effective evaluation function. When training a new stage, negative examples are bootstrapped based on the strong classifiers trained in the previous stages. The details of training a cascade of 5 stages is summarized Table 1. As the result of training, we achieved 100% correct acceptance and correct rejection rates on the training set. The total number of weak classifiers of cascade is 1024.

Table 1: Training results (WC: weak classifier; $n$: number)

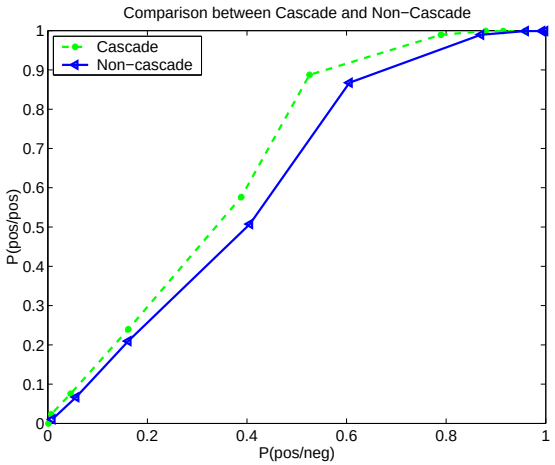
| stage | $n$ of pos | $n$ of neg | $n$ of WC | False Alarm |
|---|---|---|---|---|
| 1 | 2536 | 3000 | 22 | 0.076 |
| 2 | 2536 | 3000 | 237 | 0.069 |
| 3 | 2536 | 888 | 294 | 0.263 |
| 4 | 2536 | 235 | 263 | 0.409 |
| 5 | 2536 | 96 | 208 | 0.000 |

Figure 9: Training Set of Positive Examples (top) and Negative Examples (bottom)

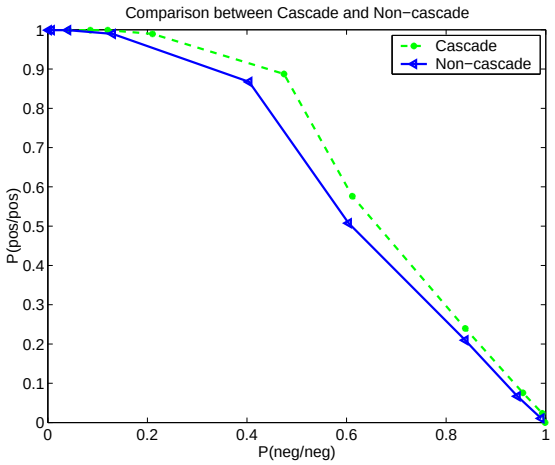
In training without cascade, all the 2536 positive examples and 43000 negative examples are used at one time to train a single strong classifier. The learning without cascade could not achieved 98.5% correct acceptance and correct rejection rates on training set, even after learning 1200 weak classifiers.

During the test, a total of 1528 aligned examples (800 qualified images and 728 un-qualified images), which are not seen during the training, are used. We evaluate each face images and give a score in terms of the confidence value $H_M(x)$ for the learning based method. The qualified and un-qualified alignment decision is judged by comparing the score with the normalized threshold of 0.

Fig. 10 quantitatively compares the two methods in terms of their ROC curves (Fig. 10(a)) and correct curves (Fig. 10(b)), where the axis label $P(pos/neg)$ means the false positive rate and so on. From Fig. 10(b), we can see that the equal error rate of the cascade is about 40%, while that of non-cascade is 47%.




(a)




(b)

Figure 10: Comparisions between the cascade and non-cascade.

## 5. Summary and Conclusions

In this paper, we analyzed the problem of AdaBoost learning with and without using cascade for classifying the non-convex manifolds. We explained why AdaBoost learning with cascade not only increases the speed of classification, but also make training easier, especially, when the manifolds of examples is non-convex and complex. Training in cascade makes use of a divide-and-conquer strategy. It splits a hard problem into several easier subproblems and solves them one by one. We proved that such a divide-and-conquerer strategy works when a rich enough, ie a complete or over-complete, set of features (and hence weak classifiers) is available. Experimental results with synthesized and real data sets demonstrate advantages of learning with cascade.

## References


[1] P. Viola and M. Jones, "Robust real time object detection", in *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 13 2001.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 12-14 2001.

[3] P. Viola and M. Jones, "Asymmetric AdaBoost and a detector cascade", in *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, December 2001.

[4] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural network-based face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–28, 1998.

[5] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[6] B.Heisele, T.Mukherjee, and T.Poggio, "Feature reduction and hierarchy of classifiers fro fast object detection in video images", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai Marriott, Hawaii, December 2001, pp. 18–24.

[7] S.Romdhani, P.Torr, B.Schoelkopf, and A.Blake, "Computationally efficient face detection", in *Proceedings of IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001, pp. 695–700.

[8] S. Z. Li, L. Zhu, Z. Q. Zhang, A. Blake, H.J. Zhang, and H. Shum, "Statistical learning of multi-view face detection", in *Proceedings of the European Conference on Computer Vision*, Copenhagen, Denmark, May 28 - June 2 2002, vol. 4, pp. 67–81.

[9] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection", Mrl technical report, Intel Labs, Dec 2002.

[10] Jianxin Wu, James M.Rehg, and Matthew D.Mullin, "Learning a rare event detection cascade by direct feature selection", in *3rd Int'l Workshop on Statistical and Computational Theories of Vision*, Nice, France, October 2003.

[11] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, , no. 5500, pp. 2319–2323, December 22 2000.

[12] Sam Roweis and Lawrence Saul, "Nonlinear dimensionality reduction by locally linear embedding", *Science*, , no. 5500, pp. 2323–2326, December 22 2000.

[13] Partha Niyogi Mikhail Belkin, "Laplacian eigenmaps for dimensionality reduction and data representation," *Technical Report, University of Chicago*, January 2002.

[14] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, August 1997.

[15] L. Valiant, "A theory of the learnable", *Communications of ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[16] Michael J. Kearns and Umesh Vazirani, *An Introduction to Computational Learning Theory*, MIT Press, Cambridge, MA, 1994.

[17] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998, pp. 80–91.

[18] L. Breiman, "Arcing classifiers", *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[19] R.E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods", *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, October 1998.

[20] J.H. Friedman, "Greedy function approximation: A gradient boosting machine", *The Annals of Statistics*, vol. 29, no. 5, October 2001.

[21] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean, "Functional gradient techniques for combining hypotheses," in *Advances in Large Margin Classifiers*, A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., pp. 221–247. MIT Press, Cambridge, MA, 1999.

[22] R.S. Zemel and T. Pitassi, "A gradient-based boosting algorithm for regression problems", in *Advances in Neural Information Processing Systems*, Cambridge, MA, 2001, vol. 13, MIT Press.

[23] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting", *The Annals of Statistics*, vol. 28, no. 2, pp. 337–374, April 2000.

[24] M. Elad, Y. Hel-Or, and R. Keshet, "Pattern detection using a maximal rejection classifier", *Pattern Recognition Letters*, vol. 23, pp. 1459–1471, October 2002.

[25] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models: Their training and application", *CVGIP: Image Understanding*, vol. 61, pp. 38–59, 1995.

[26] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *ECCV98*, 1998, vol. 2, pp. 484–498.

[27] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face recognition using active appearance models", in *Proceedings of the European Conference on Computer Vision*, 1998, vol. 2, pp. 581–695.