Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2004-023
AIM-2004-028

April 21, 2004

# Cascading Regularized Classifiers

Luis Perez-Breva

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

## Abstract

**A**mong the various methods to combine classifiers, Boosting was originally thought as an stratagem to cascade pairs of classifiers through their disagreement. I recover the same idea from the work of Niyogi et al. to show how to loosen the requirement of weak learnability, central to Boosting, and introduce a new cascading stratagem. The paper concludes with an empirical study of an implementation of the cascade that, under assumptions that mirror the conditions imposed by Viola and Jones in [VJ01], has the property to preserve the generalization ability of boosting.

# 1 Introduction

Boosting, [Sch90], is a simple scheme to combine 3 classifiers (weak learners) by majority vote and possibly outperform a single one. It has been related to Game Theory and Margin classifiers [FS96, SFBL97], however an explanation for the ability to generalize of boosting methods has so far eluded all the research efforts and increased the interest in cascades of classifiers. A notable example of this interest can be found in the work of Niyogi et al [NPS00, NPS01], which presents a similar 3 classifier scheme that replaces distribution reweighting by explicit decorrelation. That work proves equivalence with boosting and points out a connection to AdaBoost (the first implementation of boosting [Fre95, Sch99]).

In this paper I review the work from Niyogi et al. from the set theory perspective and for the case of linear classifiers. In that context, set theory will help us gain intuition on the cascade of classifiers, and will point to an interesting conclusion: weak learnability is a stronger assumption than required. This work concludes with a (unorthodox) implementation of an algorithm that effectively cascades kernel machines.

# 2 Three Classifier schemes

Boosting [Sch90] proposes to build a three weak [1] classifier scheme according to the following procedure:

- Train a classifier $h_1$ on data drawn from a probability distribution $D$.
- Build a new Distribution $D_2$ by drawing examples from $D$, such that $h_1$ performs 50% correct over $D_2$.
- Train a second classifier $h_2$ on $D_2$.
- Build a third distribution $D_3$ resampling from $D$ such that $h_1$ and $h_2$ disagree on their output on $D_3$.
- Train a third classifier $h_3$ on $D_3$.

The final output of the three classifier scheme is the majority vote of $h_1(\mathbf{x})$, $h_2(\mathbf{x})$ and $h_3(\mathbf{x})$ on a given input $\mathbf{x}$. Note that the third classifier can be replaced by a new 3 classifier scheme recursively, leading to the original Boosting stratagem.

[NPS01] suggest to decorrelate the errors of two classifiers $h_1, h_2$, and let a third one $h_3$ decide upon their disagreement. The correlation of $h_1$ and $h_2$ is represented as the covariance of their errors ($\varepsilon_{h_i}$):

$$C(h_1,h_2) = E((\varepsilon_{h_1} - E(\varepsilon_{h_1}))E((\varepsilon_{h_2}) - E(\varepsilon_{h_2}))) \quad (1)$$

and two uncorrelated classifiers result from solving the constrained minimization:

$$\begin{array}{ll} & \min_{h_1,h_2 \in H} C(h_1,h_2) \\ \text{subject to:} & \\ & p(\varepsilon_{h_2}) \leq p \\ & p(\varepsilon_{h_1}) \leq p \end{array} \quad (2)$$

where $p < \frac{1}{2}$ for $h_1$ and $h_2$ are weak learners. Hence, decorrelation pursues direct minimization of the probability of both classifiers agreeing when wrong.

## 2.1 Decorrelating Classifiers

Let us depart slightly from its original formulation, and introduce the "decorrelating classifiers" scheme from the set theory perspective. We shall assume that the sets are built sampling enough data points from the true distribution, so that division by the cardinality of the universe will yield true expectations. No assertion is made with respect to the generalization ability of this scheme at this time.

Figure 1 illustrates the minimization process as suggested in [NPS01]. $\varepsilon_{h_i}$ represents the set of errors performed by a classifier $h_i \in H$ when trained on the set $D$. Decorrelating two classifiers $h_1$ and $h_2$ is equivalent to minimize the cardinality of the intersection between their respective error sets (Figure 1 feels intuitively clearer than this wording). Note that, as illustrated in Figure 1(b), decorrelation is likely to have the side effect of increasing the cardinality of the error sets. A seemingly undesired property. We overcome this intuition by convincing ourselves that the yet to be discovered third classifier will handle the extra work without a blink. And defer a more formal discussion of why this property is irrelevant for later in the text. Then we can

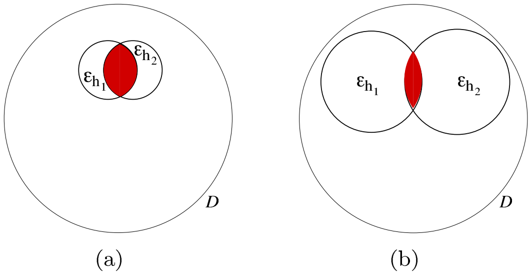


Figure 1: Decorrelating Classifiers. The outmost set represents the universe of samples drawn from a probability distribution $D$. The inner sets defined formally as $\varepsilon_{h_i(\mathbf{x})} = \{\mathbf{x} \in D \mid h_i(\mathbf{x}) \neq f(\mathbf{x})\}$ represent the set of samples that were incorrectly classified by $h_i$. **(a)** Initial state upon choosing two classifiers $h_1$ and $h_2$ from the hypothesis space $H$. **(b)** After decorrelating the classifiers $h_1$ and $h_2$ according to the minimization scheme from equation (2)

[1] a weak learner performs $\epsilon$-close to chance on the true distribution. Its probability of error is $\frac{1}{2} - \epsilon$ *for some* $\epsilon > 0$

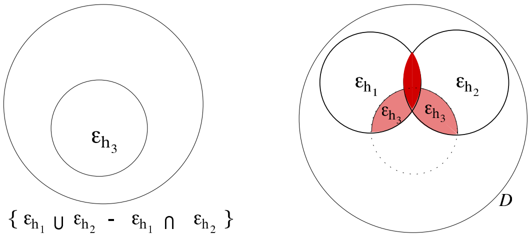
express the agreement set of $h_1$ and $h_2$ using set notation as:

$$A = \overline{(\varepsilon_{h_1} \cup \varepsilon_{h_2})} \bigcup (\varepsilon_{h_1} \cap \varepsilon_{h_1}), \tag{3}$$

which can be worded as the union of the set in which both classifiers are correct and the set in which both are wrong[2]. The agreement set, like in the original boosting, will be directly classified, unlike the disagreement set:

$$\overline{A} = (\varepsilon_{h_1} \cup \varepsilon_{h_2}) - (\varepsilon_{h_1} \cap \varepsilon_{h_1}), \tag{4}$$

which will be left for the third classifier to decide. Note that equations (3) and (4) imply that we should look forward to minimizing $(\varepsilon_{h_1} \cap \varepsilon_{h_1})$, in order to increase performance of the combination of classifiers. This is indeed a first hint on why the increase in the size of the error sets will become irrelevant.



(a) (b)

Figure 2: Training the third classifier on the disagreement set. **(a)** the outmost set represents the training set of the third classifier, that is, the disagreement set from figure 1(b), the inner circle represents the set of errors of classifier $h_3$. **(b)** Three classifier Scheme representation. The set of errors of the three classifier scheme appears shaded. The overall scheme will fail in $\varepsilon_{h_1} \cap \varepsilon_{h_2}$, and when $h_3$ produces an error over the disagreement set between $h_1$ and $h_2$. Note that $h_3$ could still perform errors in the agreement set of $h_1$ and $h_2$, although it will never be invoked if $h_1$ and $h_2$ already agreed.

Upon training the third classifier, (see Figure 2) the error of the three classifier scheme will be fully determined by the cardinality of the set satisfying $h_1(\mathbf{x}) = h_2(\mathbf{x}) \neq f(\mathbf{x})$ (with $f$ being the unknown true function) and the errors of $h_3$ on the disagreement of $h_1$ and $h_2$. Formally we write the probability of error as:

$$\begin{aligned} P_e &= \frac{1}{\mid D \mid} \left( \mid \varepsilon_{h_1} \cap \varepsilon_{h_2} \mid + \mid \varepsilon h3 \mid \right) \\ &= \frac{1}{\mid D \mid} \mid \varepsilon_{h_1} \cap \varepsilon_{h_2} \cup \varepsilon_{h_3} \mid, \end{aligned} \tag{5}$$

and note that it depends on the relative sizes of the error sets. The worst case scenario corresponds to $\varepsilon_{h_2}$ being a subset of $\varepsilon_{h_1}$, and the third classifier making mistakes on the remaining portion of $\varepsilon_{h_1}$ (see figure 3(a)), formally:

$$\varepsilon_{h_2} \subseteq \varepsilon_{h_1} \quad \text{and} \quad \varepsilon_{h_3} = \varepsilon_{h_1} - \varepsilon_{h_2},$$

which implies a maximum probability of error, from equation (5):

$$P_e = \frac{1}{\mid D \mid} \mid \varepsilon_{h_1} \mid. \tag{6}$$

We conclude that under these conditions, the final combination scheme would not be able to outperform the worst classifier in the combination ($h_1$). (Symmetric reasoning applies in case $\varepsilon_{h_1} \subseteq \varepsilon_{h_2}$)
Conversely, should the first two classifiers never agree on a mistake (i.e no unrecoverable errors), we would be looking at the best case scenario (see figure 3(b)), which we formalize as

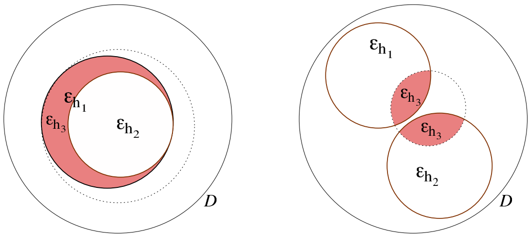
$$\varepsilon_{h_2} \cap \varepsilon_{h_1} = \emptyset,$$

and yields a minimum probability of error, from equation (5):

$$P_e = \frac{1}{\mid D \mid} \mid \varepsilon_{h_3} \mid. \tag{7}$$

That is, in the best case scenario the final performance will depend exclusively on the ability of $h_3$ to classify the error sets of $h_1$ and $h_2$; in fact, this argument formalizes the intuitive idea expressed earlier about leaving the work to the third classifier.

Finally note that $h_3$ can be replaced by a new three classifier scheme using $h_1^1$,$h_2^1$ (compare figures 2(a) and 1(a)) and do so iteratively for every $h_3^i$ until further iterating is impossible given the size of the disagreement set. Niyogi et al [NPS01] further elaborate this point to find a connection between AdaBoost and the *decorrelating classifiers* scheme.



(a) (b)

Figure 3: **(a)** Worst and **(b)** best case scenario analysis. See the text for a detailed explanation

## 2.2 Decorrelating weak learners

In the previous section we have omitted an important assumption made in both [Sch90, NPS01]. Let us assume that our classifiers are weak learners:

$$p(\varepsilon_{h_i}) = \frac{1}{2} - \epsilon \quad \text{for some } \epsilon > 0. \tag{8}$$

[2] When both are wrong, $\{\mathbf{x} \mid h_1(\mathbf{x}) \in (\varepsilon_{h_1} \cap \varepsilon_{h_2})\}$, we will call the points *unrecoverable errors*

This imposes an additional constraint on the size of our sets, and modifies the worst case scenario to:

$$\varepsilon_{h_2} \subseteq \varepsilon_{h_1} \;\text{ and }\; \varepsilon_{h_3} = \frac{\mid \varepsilon_{h_1} - \varepsilon_{h_2} \mid}{2} - 1$$

leading to the following bounds in the performance of the three classifier scheme:

$$\begin{array}{ll} P_e > & \frac{1}{|D|}\left(\frac{|\varepsilon_{h_1} \cup \varepsilon_{h_2}|}{2} - 1\right) \\ P_e < & \frac{1}{|D|}\left(\mid \varepsilon_{h_1} \cap \varepsilon_{h_2} \mid + \frac{|\varepsilon_{h_1} - \varepsilon_{h_2}|}{2} - 1\right) \end{array} . \tag{9}$$

Translating the set-theoretic bounds to probabilities under the assumption of a large enough universe of samples will take these bounds (9) to their appearance in [Sch90, NPS01]. This should be no surprise for the only novelty introduced up to this point was the set-theoretic perspective for an otherwise well known setting. Nonetheless the introduction of the set-theoretic perspective accounts for more than simple visualization aid. It has allowed us to follow a different order in the presentation with weak learnability, a central argument in [Sch90], appearing only at the end. And invites us to review if weak learnability is indeed the weakest possible assumption allowing us to cascade classifiers in a (original)-boosting-like manner. In the remainder of this article, unless otherwise stated in the text, we will purportedly omit assuming weak learnability.

### 2.3 Weak learnability is not a requirement for $h_1$ and $h_2$

In section 2.2 we have first introduced the requirement from [NPS01] of both classifiers being weak learners. I believe that within the *Decorrelating Classifiers* scheme, it was mostly introduced to point out the equivalence with boosting as it does not arise from the scheme itself. Formally, (using set notation) weak learnability requires, $h_1$ and $h_2$ satisfy:

$$\begin{array}{l} \mid \varepsilon_{h_1} \mid = \frac{|D|}{2} - m_1 \;\text{ and} \\ \mid \varepsilon_{h_2} \mid = \frac{|D|}{2} - m_2, \;\text{for some } 0 < m_1, m_2 \ll D \end{array} \tag{10}$$

Assume that given two arbitrarily weak learners, it is still possible to achieve full decorrelation[3]. Then the agreement set will contain $\epsilon \leq m_1 + m_2$ correctly classified points. The overall performance of the three classifier scheme (see equation (7)) will depend on the performance of $h_3$ (regardless of it being a single classifier or a cascade of them) on the remaining $\mid D \mid - \epsilon$ samples. However, if full decorrelation is not achieved then from equation (5) the total number of errors will be:

$$\varepsilon = \mid \varepsilon_{h_1} \cap \varepsilon_{h_2} \mid + \mid \varepsilon_{h_3} \mid . \tag{11}$$

The minimization from (2) implies that in that case, we hit the constraints before fully decorrelating $h_1$ and $h_2$, and the combination is forced to "give up" on the intersection between them as unrecoverable errors.
From the analysis in section 2.1, our goal is twofold: maximize the agreement set (3) and minimize the number of *0unrecoverable errors*. According to that goal we redefine the optimization problem as:

$$\begin{array}{ll} & \max_{h_1, h_2 \in H} \left|\overline{\varepsilon_{h_1} \cup \varepsilon_{h_2}}\right| \\ \text{subject to:} & \\ & \begin{array}{ll} \left|\varepsilon_{h_1} \cup \varepsilon_{h_2}\right| - \left|\varepsilon_{h_1} \cap \varepsilon_{h_2}\right| & < |D| \\ \left|\varepsilon_{h_1} \cap \varepsilon_{h_2}\right| & \leq m \text{ for some } m > 0 \end{array} \end{array} \tag{12}$$

which does not require weak learnability. And impose a non-zero $m$ constrain on the maximum number of "unrecoverable errors" to avoid the trivial solution ($h_1(\mathbf{x}) = -h_2(\mathbf{x})$).

As we iterate, the final performance of the cascade will depend on the number of "unrecoverable errors", and the performance of the last classifier on the remaining samples. The iteration can continue until only 2 sample points remain, it is linearly separable, and the performance of the combination depends only on the number of "unrecoverable errors" (which is upper bounded by the number of iterations times $m$).

In words, we have just replaced the weak learnability requirement imposed on the two initial classifiers by the somehow weaker requirement of forcing them to be cooperatively strong. This apparently innocent replacement will be very useful to device a way to combine kernel machines, which are more often strong classifiers than they are weak.

Finally, we shall not prove rigorously the equivalence to the *Decorrelating Classifiers* nor to *Boosting* as presented in [Sch90, NPS01], because I believe that it is at least intuitively clear that the only difference with this approach is the assumption of weak learnability.

## 3 Decorrelating linear classifiers

Having introduced the new cascade of classifiers, let us see how to use it with Kernel Machines. Kernel machines are minimizers of functionals of the form:

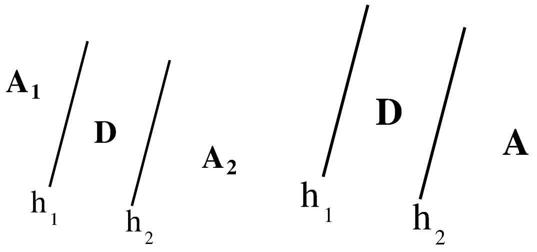
$$H[h] = \frac{C}{\ell} \sum_{i=1}^{\ell} V(y_i, h(\mathbf{x}_i)) + \parallel h \parallel_K^2 \tag{13}$$

[3] The hypothesis space is an obvious constraint imposed to our classifiers, but so is weak learnability as it constraints the ability of the minimization scheme to achieve maximum decorrelation

where $V(\cdot,\cdot)$ is a cost functional and $\parallel h \parallel_K^2$ is the norm of function $h$ in a Reproducing Kernel Hilbert Space defined by Kernel $K$.

We choose to study the case of a dot product Kernel (linear classifier), not only because of its simplicity and the usual arguments concerning hypothesis and feature space [Vap95], but also because it would otherwise be difficult to assert anything about possible overfitting without a rigorous analysis of the type of data targeted.

We further restrict the space of possible classifiers to that of parallel linear classifiers, the reason for such a dramatic reduction of freedom will hopefully become clear immediately, simplicity. Figure 4(a) shows the agreement(**A**) and disagreement(**D**) sets given two parallel linear classifiers. For the sake of this analysis I wish to introduce a last (quite unorthodox) restriction: set the classifier $h_1$ at infinity [4], that is:



(a) (b)

Figure 4: Agreement (**A**) and disagreement (**D**) sets of two parallel linear classifiers $h_1$ and $h_2$, (a) at an arbitrary location, and (b) when $h_1$ is placed at $\pm\infty$

$$h_1(\mathbf{x}) = \begin{cases} 1 & \text{if } h_1 \text{ at } -\infty, \\ -1 & \text{if } h_1 \text{ at } \infty. \end{cases} \tag{14}$$

and then Figure 4(b) applies. As in section 2.1 we are now concerned by the number of errors in the agreement set, which we control in the choice of $h_2$. We consider a cost function for $h_2$

$$V(y, h_2(\mathbf{x})) = (1 - yh_2(\mathbf{x}))_+,$$

and introduce the agreement and disagreement sets cconsidering $h_1$ in the cost function as follows:

$$V_A(y, h_1(\mathbf{x}), h_2(\mathbf{x})) = \theta\big(h_1(\mathbf{x})h_2(\mathbf{x}) - 1\big)\big(1 - yh_2(\mathbf{x})\big)_+ \tag{15}$$

$$V_D(y, h_1(\mathbf{x}), h_2(\mathbf{x})) = \theta\big(1 - h_1(\mathbf{x})h_2(\mathbf{x})\big)\big(1 - yh_2(\mathbf{x})\big)_+ \tag{16}$$

with $\theta$ defined as:

$$\theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \textit{otherwise.} \end{cases}$$

Equation (14) allows us to redefine $h_1(\mathbf{x}) = \delta \ \forall \ \mathbf{x}$. Which for the case $\delta = 1$, turns (15) and (16) into:

$$V_A(y, h_2(\mathbf{x})) = \begin{cases} (1 + h_2(\mathbf{x})) & \text{if } y < 0 \wedge h_2 > -1 \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

$$V_D(y, h_2(\mathbf{x})) = \begin{cases} (1 - h_2(\mathbf{x})) & \text{if } y > 0 \wedge h_2 < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

The restrictions imposed, however unorthodox, have led us to relate decorrelation to the asymmetry in the cost of each class. Given the choices of kernel and cost functional, we may use a linear Support Vector Machine with different regularization term for each class [Osu98]. And choose for each possible choice of $\delta \in \{0, 1\}$, among the minimizers of (13) for the one with minimum cost.

### 3.1 ...and cascading them

At the risk of abusing of the patience of the reader, I should like to emphasize that with the above construction, the cascade is now fully dependent on the disagreement set only (recall figure 4(b)), which can thereon be treated as a completely new problem. This means that given a pair of classifiers $h_1$ and $h_2$, we shall consider their agreement set for classification purposes and train a new classifier or cascade of them on their disagreement set.

## 4 Empirical analysis

Section 3 led us to the conclusion that we can choose to cascade linear classifiers if we adequately control the asymmetry in the regularization term for each class. The problem arises when attempting to set the regularization parameters, for then information on the geomtery of the problem, namely, the sampling as well as the distribution of the data is required beforehand for every new iteration.

We can overcome this problem by making some assumptions on the data and its geometry. I choose to assume that regardless of where the hyperplane lies at every iteration, noise exists only in the direction perpendicular to the hyperplane and it is symmetrically distributed with respect to it. This assumption is close enough to nonsense, for it can only be true in the highly uninteresting case of all the hyperplanes in successive iterations being parallel. Yet, this assumption makes modifying the regularization parameters equivalent to

[4] One might argue that a classifier at infinity is nothing but a mere mathematical construct, and will probably be right. We choose to see the classifier at infinity as a calibration tool for "chance", and as such, however unrthodox in machine learning, we note the similarity with setting boundary conditions in physical problems such as electrostatics and gravitational potentials.

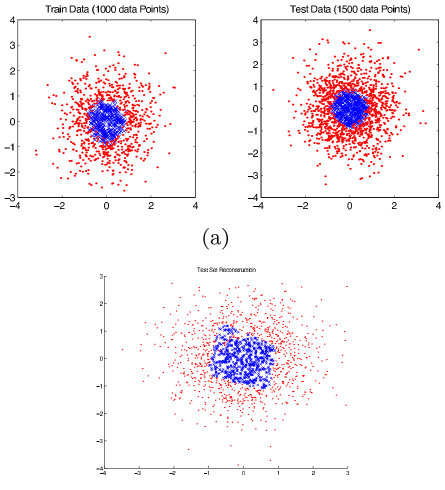
controlling the threshold, thus similar to the work by [VJ01].

The easiness for identifying errors as support vectors makes Support Vector Machines [Vap95] a good candidate to train the classifier that will be at a finite location with the one at infinity being parallel to it. The following simple algorithm examines the error support vectors and determines the threshold ($b_{h_i}$)of the classifier as well as its direction (i.e., which side will be considered the disagreement):

```
▷ Find $\mathbf{x}_p = \underset{false\ positives}{\arg\max}\ f(\mathbf{x})$ .
        if $\nexists\ \mathbf{x}_p$
              $mfp \leftarrow \infty$
        else
              $mfn \leftarrow f(\mathbf{x}_p)$
              if $mfp \neq \max_{\mathbf{x} \in S.V.} f(\mathbf{x})$
                    $mfp \leftarrow mfp + \epsilon$
▷ Find $\mathbf{x}_n = \underset{false\ negatives}{\arg\min}\ f(\mathbf{x})$ .
        if $\nexists\ \mathbf{x}_n$
              $mfn \leftarrow -\infty$
        else
              $mfn \leftarrow f(\mathbf{x}_n)$
              if $mfn \neq \min_{\mathbf{x} \in S.V.} f(\mathbf{x})$
                    $mfn \leftarrow mfn - \epsilon$
▷ Assign values to $b_{h_1}$ and $b_{h_2}$
        if $abs(mfn) < abs(mfp)$
              $b_{h_1} \leftarrow mfn$
              $b_{h_2} \leftarrow \infty$
        else
              $b_{h_1} \leftarrow -\infty$
              $b_{h_2} \leftarrow mfp$
```

Figure 5 shows the results of training and testing the cascade on a clearly non linearly separable synthetic problem, and its reconstruction ability for the test data. Figure 6 shows the results of training and testing the cascade on a similar synthetic problem. Figure 6(b) shows how the cascade of linear classifiers does not overfit the data.

Finally, when ran on a real, high dimensional problem such as face classification with the data from [HPP00], the cascade still outperforms a single linear classifier (see figure 7), and proves to not overfit the training data. However, the cascade is not able to outperform the results reported in [HPP00] for a second degree polynomial kernel. In this case, the different origins of the train and the test data appear as the most plausible cause for the slightly worse performance of the cascade; in close connection to its higher dependence on the geometry of the data.



(a)

(b)

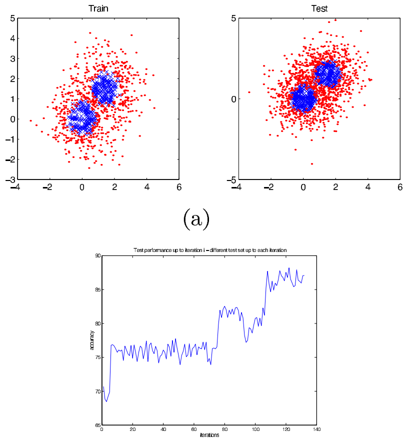
Figure 5: **(a)** Synthetic train and test data. The problem is clearly non linearly separable. **(b)** Reconstruction of the test data with the simplified algorithm. Note that the algorithm is able to approximate non-linear decision surfaces using only linear classifiers.

## 5 Conclusions

We have seen how a new perspective on majority voting allows us to loosen the requirement of weak learnability in Boosting-like architectures of classifiers. Extending the work of Niyogi et al. to consider cooperatively strong learners, we have presented a novel algorithm to combine, better say cascade, kernel machines under somehow unorthodox conditions. Such combination of kernel machines first introduces the possibility of harnessing the power of kernel machines in the form of a cascade that simplifies the choice of the kernel. The results of the brief empirical study following the analysis encourage further developing this line of methods, which could soon become an alternative to the always difficult choice of a kernel.

Further research would strongly benefit from a study of the relationship between the asymmetry in the regularization parameter of each class and the geometry of the data. To the best of my knowledge, studies of such relationship are scarce if at all existing in the current literature. The difficulty of the analysis of the regularization parameter is often appointed as the main cause for disregarding that avenue of research.

A careful study of the significance of the classifier at infinity as a calibration for chance, in connection to boundary value problems appears as a convenient alternative to the analysis of the regularization parameter. And is the current focus for extensions of the work presented here.



(a)

(b)

Figure 6: **(a)** Synthetic train and test data. The problem is clearly non linearly separable. **(b)** Test Performance of the cascade. Each point in the curve represents a newly generated random instance of the original problem, thus the noisy appearance.

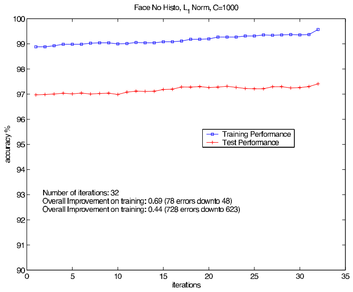


Figure 7: Results on the face dataset(361-dimensions). A single $2^{nd}$-degree polynomial SVM missclassified 520 samples on the same test set.

## Acknowledgements

I would like to thank Antonio Torralba, Sayan Mukherjee, and Tomaso Poggio for their proverbial patience.

## References

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[FS96] Y. Freund and R.E. Schapire. Game theory, on-line prediction and boosting. In *Proc. 9th Annu. Conf. on Comput. Learning Theory*, pages 325–332. ACM Press, New York, NY, 1996.

[HPP00] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. A.I. memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000.

[NPS00] P. Niyogi, J. B. Pierrot, and O. Siohan. Multiple classifiers by constrained minimization. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[NPS01] P. Niyogi, J. B. Pierrot, and O. Siohan. On decorrelating classifiers and combining them. Unpublished, Talk at MIT, January, January 2001.

[Osu98] E. Osuna. *Support Vector Machines: Training and Applications.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1998.

[Sch90] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Sch99] R.E. Schapire. A brief introduction to boosting. In *Proc. 16th International Joint Conference on Artificial Intelligence*, 1999.

[SFBL97] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In *Proc. 14th International Conference on Machine Learning*, pages 322–330. Morgan Kaufmann, 1997.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[VJ01] P. Viola and M.J. Jones. Robust real-time object detection. Technical report, Cambridge Research Laboratory, One Cambridge Center, Cambridge, MA 02142, USA, February 2001.