

Pruning Training Sets for Learning of Object Categories

Anelia Angelova[§]

Yaser Abu-Mostafa[‡]

Pietro Perona[‡]

[§]Computer Science Department

[‡]Electrical Engineering Department

California Institute of Technology, Pasadena, CA 91125

Abstract

Training datasets for learning of object categories are often contaminated or imperfect. We explore an approach to automatically identify examples that are noisy or troublesome for learning and exclude them from the training set. The problem is relevant to learning in semi-supervised or unsupervised setting, as well as to learning when the training data is contaminated with wrongly labeled examples or when correctly labeled, but hard to learn examples, are present. We propose a fully automatic mechanism for noise cleaning, called 'data pruning', and demonstrate its success on learning of human faces. It is not assumed that the data or the noise can be modeled or that additional training examples are available. Our experiments show that data pruning can improve on generalization performance for algorithms with various robustness to noise. It outperforms methods with regularization properties and is superior to commonly applied aggregation methods, such as bagging.

1. Introduction

Learning an unknown target function from examples is a difficult problem in its own right. The task is further complicated if some of the examples in the dataset are mislabeled or are otherwise hard to learn for the chosen model, as is the case with real-life data. Finding troublesome examples is a 'chicken-and-egg' dilemma because good classifiers for the object category are needed in order to determine which examples are noisy. On the other hand, learning on noisy data may result in poor classifiers. We explore whether generalization performance can be improved by eliminating noisy examples and how to reliably identify them.

Real-life training data can have various sources of contamination. For example, wrongly labeled examples might be present due to human mistakes while labeling or a result of data received in a semi-supervised fashion, figure 1 (top). Quite often, examples that are hard to learn can also fall into the dataset, figure 1 (bottom), because the data collection and labeling is done independently from the learning process, by people unaware of its future use.



Figure 1: **Face data with classification noise (non-face examples wrongly labeled as faces (top)) and with within-sample outliers (hard to learn face examples (bottom)).** The training data we use contains both sources of noise. Noisy or hard to learn examples are marked with a red box manually. The goal of data pruning is to identify and eliminate them automatically in order to improve generalization performance.

Pruning of noisy examples is important to fully automate the process of data collection and learning for object recognition. The efforts in this direction have been focused on creating complicated models or features [5], particularly tailored to the target domain. Our key idea is that automatic noise elimination can be done without incorporating very complicated machinery or domain specific information but, instead, using only weaker clues about the target and that they can be retrieved even in very noisy cases.

1.1. Previous Work

Robust methods are an important ingredient in most computer vision algorithms which work with real-life data and applications. A large body of research, a detailed survey of which is beyond the scope of the paper, addresses robust techniques in various machine vision applications. To name a few: short and wide baseline stereo, motion segmenta-

tion, optical flow estimation. Data cleaning is also pertinent to data mining and database applications. Although previous research in robust methods in computer vision and data mining has been used as inspiration we take an alternative route in addressing the problem of learning from examples in a robust way.

Learning in the presence of noise is shown to be possible for both classification [1] and malicious noise scenarios [9] but depend on the existence of potentially infinite training data. For the most realistic, malicious noise scenario, it is proved that no general method for learning with noise exists [9].

Some methods for learning in noisy cases use a general penalty on all examples - regularization methods [17], [18], or give preference to simpler models - model selection. An alternative approach is to average the decisions of several functions, as in bagging [3], in order to smooth out the influence of noisy examples. Those methods are not very successful in high noise cases or when there are particularly outlying examples. In our work we suggest to remove those examples, in order to be more robust in learning.

Previous research demonstrates that removing examples is worthwhile [13], [16], [14]. The methods have various mechanisms to identify examples, suspicious, surprising or close to the boundary, but the decision as to which of the examples are troublesome and should be excluded remains problematic. Those methods are either not fully automatic [16], require an additional independent set [14] or depend on human supervision, although as a last resort, to make the method more reliable [13].

Learning with poorly labeled examples is reminiscent of training with both labeled and unlabeled data [2]. The main difference is that those methods use reliably labeled data to start the learning from [2], [21]. In our formulation we have to identify which are the reliable examples.

Finally, RANSAC [6], has been widely used in various computer vision applications. The RANSAC method assumes the target model is known and performs multiple trials of selecting small subsets of the data to exactly estimate the model. The final solution is the one with maximal support from the data. To ensure that a good solution exists a large number of trials have to be performed. Considerable amounts of noise can be tolerated but the strong assumption on the generating model being known and the requirement of a fixed number of examples (presumably small) to estimate the model parameters are not always easy to guarantee.

Similarly to RANSAC, many methods are successful in identifying noisy examples after assuming particular generative models or distributions for the underlying data or noise [5], [21] and applying RANSAC for the selected model [5]. Conversely, we do not want to assume particular distributions for the data and demonstrate the data pruning method in discriminative learning scenario. This is

a very important point because the assumed model can be fairly complicated, as is the constellation model [5] or the GMM [21]. Instead we opt for a simpler and more accessible approach which is also more general.

1.2. Outline of the paper

In section 2 we propose a data pruning mechanism for automatic noise cleaning. The challenge of the method is how to identify examples which deteriorate the performance and how to automatically do so. A modification of the data pruning method, specific for learning of object categories, which uses weaker learners, is explored in section 3. The usefulness of data pruning is demonstrated in experiments for face recognition with noisy examples, on very challenging dataset collected from the web, section 4. Data pruning outperforms commonly used robust-to-noise algorithms.

2. Data pruning method

In our setup we have training data in which some of the examples are outliers or are otherwise noisy. The learning model is not fixed and parametric as in [5], [21], but discriminative, such as Neural Network, AdaBoost or SVM, for which the popular RANSAC type technique would require too many trials to guarantee a good solution. The noisy examples are not known and they are potentially going to affect the learning process and the solution in an adversary way. Our goal is to identify the examples which are noisy or troublesome for learning. If we just measure how well the examples do with respect to the decision boundary it would not be successful because the algorithm may have overfitted and learned those outlying examples very well. This problem is well known in robust statistics [20].

In data pruning method we propose the idea of learning independently different aspects of the data through learning multiple classifiers and arbitrate which examples are most likely noisy or wrongly labeled by probabilistic reasoning. To realize that general idea, we suggest two very simple instances of the proposed steps: bootstrapping and Naive Bayes algorithm, figure 2.

Although simple, the proposed approach, has important ingredients. Firstly, the method comes from the intuitive idea that outliers or noisy examples would lie in a part of the feature space where their neighbors have contradicting label. However, evaluating which are the neighbors is very problematic in the initial feature space because of the curse of dimensionality in high dimensions. Alternatively, in data pruning, the affinity of the examples is determined by stronger than the initial feature space learners, which are selected according to the data. Those multiple learners are created in such a way so that they are influenced differently by potentially troublesome examples. In

essence, the learners’ responses can be considered a higher-level feature space which has mostly relevant features and is of smaller dimension. In this more meaningful feature space we perform probabilistic reasoning to identify inconsistencies among the data.

It is useful to compare this approach to RANSAC. RANSAC performs multiple trials to guarantee that at least one clean data subset is present among the trials. A large number of trials is needed, especially for very noisy datasets or when a subset of larger size is used for training, as is often true with discriminative models (SVM, Neural Networks). Subsequently, the learner with best performance (presumably trained on least contaminated data subset) is selected. In data pruning we set out to identify the noisy examples and eliminate them rather than perform numerous of trials hoping to achieve a clean data subset.

2.1. Constructing multiple classifiers

As stated before, the learning algorithm would be influenced in an adverse way by noisy examples or outliers and would overfit or create a poor discriminatory boundary. The trouble is that we do not know which examples are actually causing the poor behavior. Our approach is to allow those troublesome examples to influence multiple independent learners in different ways. Thus the troublesome examples can be identified as the ones which create disagreement among classifiers. This can be achieved in various ways: by randomization of the training data or excluding random subsets of examples from the data or by selecting different subsets of input features or injecting noise. We create multiple diverse classifiers using bootstrapping because it is known to encourage diversity and give an imbalance in influence of certain random examples [3].

2.2. Combining classifiers’ votes

We apply an inference machine to find out which examples have created discordant classifications. In order to find wrongly labeled examples we compute a probabilistic estimate of the true label of an example and compare it to the given label. We are interested in finding the probability $p(y|\mathbf{X})$ of the label y of an example, given the data \mathbf{X} . The label can be determined by the ratio

$$\frac{P(y = 1|\mathbf{X})}{P(y = 0|\mathbf{X})} = \frac{P(\mathbf{X}|y = 1)P(y = 1)}{P(\mathbf{X}|y = 0)P(y = 0)}.$$

The ratio is compared to a threshold (in our case 1) to decide the most likely label of the example. After estimating the new labels, the examples whose labels disagree with the original ones are pruned.

The probabilities $P(y = 1)$ and $P(y = 0)$ are set to our prior belief of the data. $P(\mathbf{X}|y = 0)$ and $P(\mathbf{X}|y = 1)$

can be modeled and estimated from the data. We use Naive Bayes classifier [4] and decompose the data into several independent attributes A_j . In our case the attributes are the projections of the input data on the multiple classifiers:

$$P(\mathbf{X}|y = c) = \prod_{j=1}^J P(A_j|y = c), c \in \{0, 1\}.$$

We model the projections on each attribute as Gaussian distributions. Naive Bayes is justified if we assume that the attributes are independent. Quite often, however, the rule can be successfully applied even if the attributes are not fully independent [4].

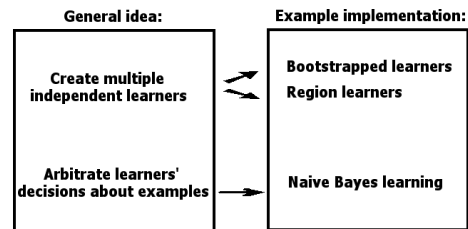


Figure 2: **Data pruning idea (left) and its particular implementation with bootstrapping and Naive Bayes (right). An alternative idea with ‘region’ learners will be presented in section 3.**

2.3. Summary, data pruning

In summary, the proposed data pruning technique requires two components: creating multiple semi-independent classifiers learned on the input data, each one of them capturing different aspects of the target, and a probabilistic reasoning machine to identify examples which are in contradiction with most learners and therefore noisy. To explore how effective this simple idea is, we select in our experiments simple instances of the required components, namely, bootstrapping and Naive Bayes classifier, figure 2.

3. Cleaning contaminated face data

In this section we apply the data pruning method for learning face category by training on data collected from the web, figure 4. The face dataset is quite challenging and would contain difficult examples, the so called within-class outliers, due to extreme face orientations, occlusions (glasses, hats, etc.), or variable lightning conditions. Additionally, we introduce artificially mislabeled background examples to test our ideas in a controlled experiment parameterized by the level of noise.

We test data pruning with two algorithms on the different ends of noise tolerance spectrum: SVC (the regularized version of SVM [18], we will use both names interchangeably) and AdaBoost [7]. In particular, SVC is robust to noise because of the regularization mechanism, whereas AdaBoost is known to be sensitive to even small amounts of noise leading to overfitting [17].

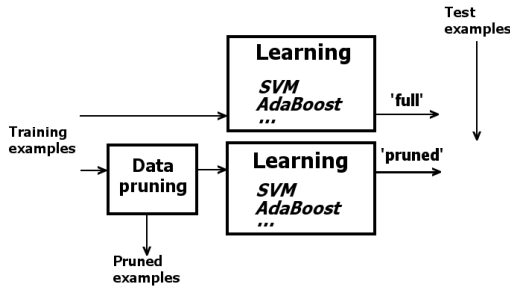


Figure 3: **Testing the data pruning method. Both the 'full' and 'pruned' methods use the same resources in terms of training/test data and learning algorithm.**

To test the usefulness of data pruning mechanism we compare the performance of the same learning algorithm (SVM or AdaBoost) with (the 'pruned' method) and without data pruning (the 'full' method) on the same initial training set, figure 3. Both methods use the same training set but eventually, the 'pruned' one will be learning on a smaller set, a disadvantage in terms of good generalization or if useful examples have been erroneously pruned. We report our results on an independent test set which is the same for both methods. The learning algorithms are allowed to select their optimal set of parameters with cross validation on the training data they have available, which is a fair way to give the 'full' and 'pruned' methods equal opportunities.

3.1. Experimental setup

The training set, figure 4, is composed of 1000 examples in which the number of correctly labeled face and non-face examples is equal, the remaining number of examples, dependent of the noise level, are background examples which are wrongly labeled as faces. For example, for the 90% noise case we will have 900 non-face examples labeled as faces, 50 correctly labeled faces and 50 correctly labeled non-faces. The test set is composed of 500 face and 500 non-face examples. It is independent of the training set and has not been contaminated. Within each run the same test set is used to report the results of learning on pruned and full sets. The results are averaged over 10 runs.

In order to learn the object category the target patches are aligned first. We applied feature projections, proposed



Figure 4: **A subset of the training data, face examples (top) are contaminated with wrongly labeled non-faces, non-face examples (bottom) are random patches cut from images not containing the target. Face data may also contain within-class outliers - correctly labeled but hard to learn examples.**

by Murphy et al. [15] using only the first moment statistics (sum of pixels) in the rectangular regions, instead of the second and fourth moment statistics as in [15]. We have 30 masks, 13 filter channels and 1 measurement per mask and channel, a total of 390 features [15]. The feature projections are selected in this particular way because the dataset is quite challenging and pixel-wise correspondence or even correspondence within small neighborhood is highly unlikely to give consistent response among the face examples, no matter that the faces are aligned. Using several filter channels, rather than raw pixel values, helps reduce variations in illumination.

3.2. Data pruning with weaker learners

In the previous section we proposed to construct multiple learners by bootstrapping the data. Those bootstrapped learners are of comparable power to the learning model (AdaBoost or SVM). Here, we test the hypothesis of whether data pruning can be performed with learners which are much weaker than the target model. An intuition of the idea is given in figure 5. The weaker learners would be correct to some extent on the data but cannot be used alone for final classification. Each one of them could capture some aspect of the data and learn to classify with respect to it. Subsequently those classifiers would agree with large probability on good face examples and disagree on wrongly labeled or outlying examples.

We create the weak learners in the following fashion. We split the image in 23 slightly overlapping regions, similar to the masks in [15] excluding the larger rectangles so that

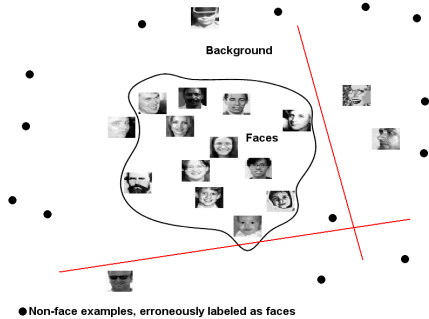


Figure 5: **Schematic visualization of the idea of data pruning with weaker learners. Weaker learners (marked with red) are not as powerful as the 'true' decision boundary (marked with black) but can learn enough about the target to be able to identify wrongly labeled or outlying examples.**

not to receive fully dependent classifiers. We call them region learners, figure 6. Each classifier receives the information from filter channels [15] in only one of those regions and is trained using AdaBoost of 100 iterations with rectangular features [19]. The features are the sum of pixels in rectangles of various sizes at different positions of the sub-region. The proposed weaker learners use only a restricted view of the face image, figure 6. They are not independent but would capture different aspects of the data. They are weaker, indeed, because a 100 iteration AdaBoost is not sufficient to learn a full face category [19].



Figure 6: **'Region' learners, an alternative to bootstrapped ones, ignore the information outside a pre-defined region. They are weaker learners and exploit the fact that the target is a visual category. The region learners implemented in the paper are AdaBoosts with rectangular features [19].**

4. Experimental results

We perform data pruning with the two variations of creating multiple classifiers as described before, see figure 2: with bootstrapped learners (received from learning on multiple subsets of the data), section 2, and with region learners,

section 3. Note that pruning with bootstrapped learners is algorithm dependent, i.e. if using SVM, for example, we need to learn several SVMs on different bootstrapped samples; data pruning with region learners is algorithm independent, i.e. we can use the same data pruning technique with AdaBoost or SVM. With the latter, we wanted to test if it is possible to perform noise cleaning using model independent learners which are of considerably weaker power than the original algorithm. We use 30 learners for bootstrapped and 23 for region learners.

Our results for SVM algorithm, figures 7 and 8, show that pruning the data is helpful for both bootstrapped and region based learners, especially in the presence of noise. The test error decreases if we preprocess the data and prune it. The scatter plots, figure 7, show the errors in each individual run so that we can see that pruning the data is consistently better and is not due to isolated felicitous cases. Figure 8 presents the same information from figure 7 but in different form, to allow convenient comparison.

Not surprisingly, for 90% noise, the pruning method gives very little advantage. The algorithm cannot find any consistent projections from this data so that to give reasonable opinion of which examples are wrongly labeled because of poor signal-to-noise ratio and challenging face examples in the data. A good thing, however, is that in this case the pruning mechanism gives up on eliminating examples rather than deteriorating the performance, figure 7. This is due to the probabilistic reasoning machine which eliminates only examples determined as troublesome with large confidence. This is an important advantage of the pruning technique because what one would expect at this noise level is a catastrophic pruning of examples leading to a significant deterioration in performance. We do not notice a significant improvement in the test error from pruning in the no noise or small noise cases. This is due to the fact that the algorithm is SVM with regularization which can cope with significant levels of noise in the data. A gratifying result is that performance does not deteriorate with pruning when there is no added noise in the training data.

Examples which have been selected to be pruned are shown on figure 9 and statistics of how well wrongly labeled examples are identified is shown in figure 10. We can notice that among the examples selected for elimination a large majority are wrongly labeled background patches. Some of them are faces which are hard for learning, especially if one thinks in the context of our feature space - which is rectangles of various size at particular location in the image i.e. they would not capture large variations in head orientation, occlusions, extreme lightning conditions.

To see how data pruning behaves with algorithms with different robustness to noise we provide experiments also with AdaBoost, using region based pruning, figure 11. Comparing to SVM with the same pruning mechanism we

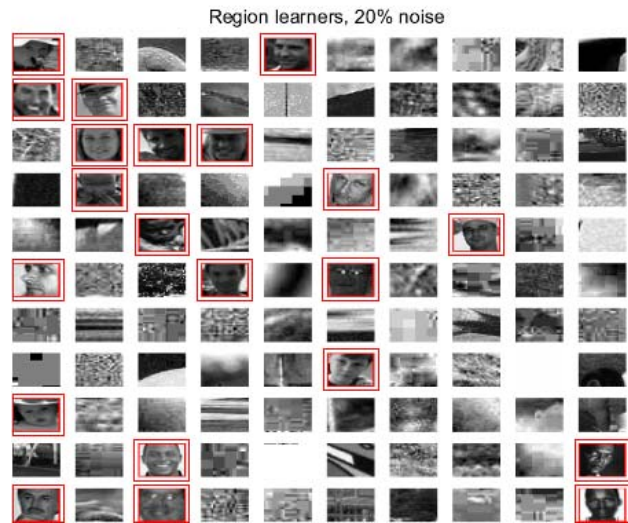
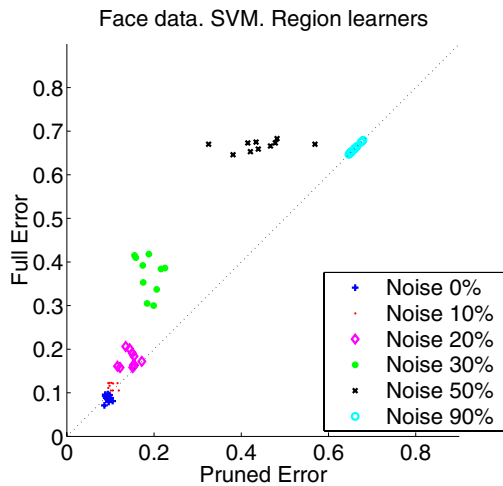
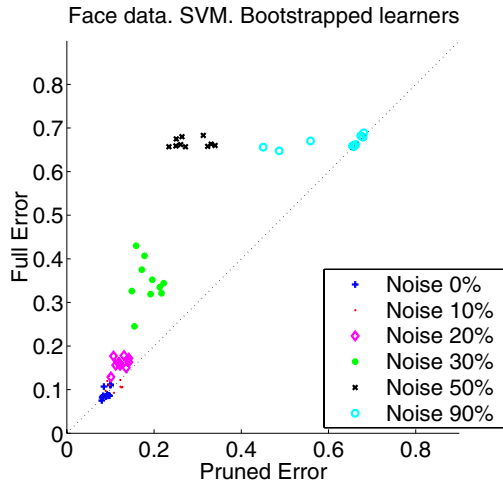


Figure 7: **Data pruning results. Test errors of learning with full and pruned methods for different levels of noise. Here we have the same learning algorithm (SVM with regularization) but different data pruning methods: with bootstrapped (top) and region (bottom) learners.**

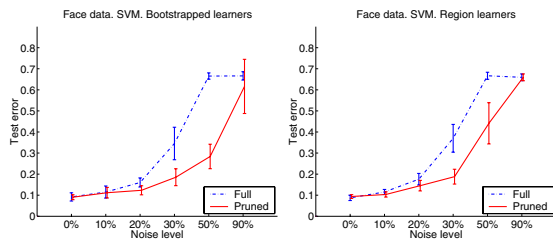


Figure 8: **Summary of the results in figure 7.**

can notice the reactions of both algorithms to different levels of noise. Namely, AdaBoost is sensitive to even small

Figure 9: **Examples which were originally labeled as 'faces' in the training data and which the algorithm identified as worth removing. Apart from spotting wrongly labeled background examples, the algorithm also identifies for elimination hard to learn face examples. The actual face examples are outlined with a red box. Bootstrapped learners (top) and region learners (bottom).**

amounts of noise [17] and would benefit from pruning in any noisy situation. SVM is robust to reasonable amounts of noise, but would need pruning much more than AdaBoost in high noise cases.

In summary, for reasonably large amounts of noise, data pruning methods can identify and remove noisy examples and by doing so can improve the generalization for both SVM and AdaBoost. This makes the data pruning technique very promising in learning from noisy datasets.

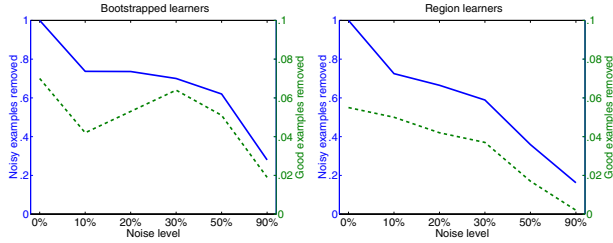


Figure 10: **Statistics on the identified wrongly labeled examples. Bootstrapped (left) and region learners (right). The solid line, left scale, shows the portion of correctly identified wrongly labeled examples among the ones with artificially flipped labels. The false alarm rates (correctly labeled examples which the algorithm proposed for pruning) are shown with a dashed line, right scale. The latter are measured against the whole data. For the no noise case, correctly detected portion of flipped labeled examples is formally set to 1 for the sake of visualization (there are no examples with flipped labels in this case). Both data pruning techniques are successful in identifying some of the wrongly labeled examples, even so in the 90% noise case. Some of the examples counted as false alarms might be genuinely hard to learn faces which are worth removing.**

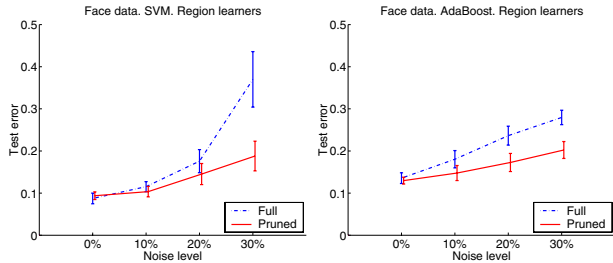


Figure 11: **Data pruning with different learning algorithms. Here we use the same data pruning procedure (based on region learners) with learning algorithms with various sensitivity to noise: SVC (left), robust to noise because of its regularization capabilities, and AdaBoost (right), sensitive to even small amounts of noise.**

4.1. Comparison to aggregating classifiers

As we are using multiple classifiers to decide on which examples to prune, we wanted to test if simply aggregating them, e.g. by bagging [3] would not give comparable performance to our pruning mechanism. Bagging would produce a classifier which can span a more complex space, a subspace of linear combination of functions, than the initial classifier, which is a single function. Our results show,

that pruning noisy examples is more useful than bagging in this setting, figure 12. By aggregating many classifiers, bagging tries to average out the noise effects which is of little avail for small or reasonable noise levels. For high noise cases bagging starts to approach the data pruning results. The comparison is given only for pruning based on bootstrapping because with the region learners the bagged classifier would be using completely different model and feature space so it makes no sense to compare it to the original algorithm, SVM or AdaBoost.

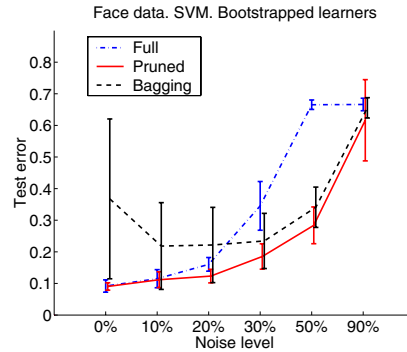


Figure 12: **Comparison between data pruning and aggregating multiple bootstrapped learners (bagging).**

5. Conclusions and future work

The proposed data pruning method is a first important step towards learning algorithms dealing with large noise levels or in semi-supervised settings. We have seen that in noisy situations it makes more sense to try to identify and eliminate the noisy examples rather than apply the commonly used robust-to-noise methods on the whole data.

The proposed data pruning procedure is fully automatic, unlike [16], [13], does not use additional resources, for example more training data or extra test set, like [14], and can be applied with any learning algorithm. It does not assume that the target distribution can be modeled as previous approaches [5], [21]. This comes at the cost of more computation for learning the multiple classifiers. Naturally, we have explored a more efficient approach where weaker learners are used. In this case the pruning method has very little time overhead. In fact, we experienced that SVM classifier needs considerably larger time to converge to a solution if the data is very contaminated than combined data pruning and learning on a cleaner set. This makes the method very appealing to apply in noisy cases. An interesting research direction is to explore whether data pruning is possible with weak learners in the sense of AdaBoost [7], i.e. slightly better than random guessing, and whether they would be

sufficient to learn in our high noise setting.

We have seen that pruning noisy examples from the data is worthwhile for both algorithms with regularization properties like SVC and algorithms which are sensitive to noise such as AdaBoost. So, in high noise cases, it can be very helpful as an additional mechanism before training, no matter if the algorithm is robust to noise or not. We also find that data pruning is superior to largely used aggregation methods, such as bagging. An important future research direction is to explore the conditions for troublesome example elimination from theoretical point of view. A natural extension is to be able to control the level of independence of classifiers in the spirit of [11] because bootstrapping might not be creating sufficiently independent learners.

The method could be applied for acquiring data with very little supervision. Scores of images containing a requested target object can be retrieved automatically by Web search machines, such as Google [8], as a response to keyword queries. As the image search and indexing is not content based, there would be many images which contain the target category but also numerous ones which are not relevant to the target. A natural extension to our data pruning method could be useful in this setting. An imminent step is redefining the feature set in order to allow translation invariant detection of features. We plan to address this problem in our future work.

Finally, we should recall here the no-free-lunch theorem for learning with noise [12] which states that we should not expect to improve generalization if we have no prior information about the noise level or the data. Independent research points to the fact that applying regularization methods on truly noise-free dataset is detrimental [10]. The above mentioned results tell us that we should not expect miracles from noise cleaning algorithms in general, and from data pruning in particular, if we have no information about the noise level or the data. However, for all practical purposes, where the data is suspected to contain outliers or noise the data pruning method can be used successfully, as shown in this paper, on the very challenging face dataset.

Acknowledgments

This research is supported by the NSF Center for Neuro-morphic Systems Engineering grant EEC-9402726.

References

- [1] Angluin, D., Laird, P., Learning from noisy examples, *Machine Learning*, 2, 343:370, 1988
- [2] Blum, A., Mitchell, T., Combining labeled and unlabeled data with co-training, *COLT: Conference on Computational Learning Theory*, 1998
- [3] Breiman, L., Bagging predictors, *Machine Learning*, 24(2), 123-140, 1996
- [4] Duda, R., Hart, P., Stork, D., *Pattern Classification*, John Wiley and Sons, Inc., New York, 2001
- [5] Fergus, R., Perona, P., Zisserman, A., A Visual Category Filter for Google Images, *ECCV*, p.242-256, 2004
- [6] Fischler, M., Bolles, R., Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *CACM*, 24(6):381-395, 1981
- [7] Freund, Y., Schapire, R., A decision-theoretic generalization of online learning and an application to boosting, *Proc. 2nd European Conf. on Computational Learning Theory*, 1995
- [8] [<http://www.google.com>]
- [9] Kearns, M., Li, M., Learning in the presence of malicious errors, In *Proc. Twentieth Annual ACM Symposium on Theory of Computing*, p.267-280, Chicago, Illinois, 2-4 May 1988
- [10] Koistinen, P., Asymptotic theory for regularization: One-dimensional linear case, *Advances in Neural Information Processing Systems* 10, p.294-300, 1998.
- [11] Liao, Y., Moody, J., Constructing heterogeneous committees using input feature grouping: Application to economic forecasting, *Advances in NIPS*, 12, 2000.
- [12] Magdon-Ismael, M., No Free Lunch for Noise Prediction, *Neural Computation*, 12:547-564, 2000.
- [13] Matic, N., Guyon, I., Bottou, L., Denker, J., Vapnik, V., Computer aided cleaning of large databases for character recognition, *Int. Conf. on Pattern Recognition*, 1992
- [14] Merler, S., Caprile, B., Furlanello, C., Bias-variance control via hard points shaving, *International Journal of Pattern Recognition and Artificial Intelligence*, 2004. In Press.
- [15] Murphy, K., Torralba, A., Freeman, B., Using the forest to see the trees: a graphical model relating features objects and scenes, *Advances in NIPS*, 2003
- [16] Nicholson, A., *Generalization Error Estimates and Training Data Valuation*, Ph.D. Thesis, California Institute of Technology, 2002
- [17] Rätsch, G., Onoda, T., Müller, K., Regularizing AdaBoost, *Advances in Neural Information Processing Systems* 11, 564-570, 2000.
- [18] Schölkopf, B., Smola, A., *Learning with kernels*, The MIT Press, Cambridge MA, 2002.
- [19] Viola, P., Jones, M., Rapid object detection using a boosted cascade of simple features, *CVPR*, 2001
- [20] Weisberg, S., *Applied linear regression*, John Wiley & Sons, 1985
- [21] Xie, L., Perez, P., 'Slightly supervised learning of part-based appearance models', Workshop on Learning, CVPR, 2004.