Classification on Soft Labels is Robust Against Label Noise

Christian Thiel

Institute of Neural Information Processing, University of Ulm, 89069 Ulm, Germany
christian.thiel@uni-ulm.de

Abstract. In a scenario of supervised classification of data, labeled training data is essential. Unfortunately, the process by which those labels are obtained is not error-free, for example due to human nature. The aim of this work is to find out what impact noise on the labels has, and we do so by artificially adding it. An algorithm for the noising procedure is described. Not only individual classifiers are studied, but also ensembles of classifiers whose answers are combined, increasing the overall performance. Also, we will answer the question if classifiers trained on soft labels are more resilient to label noise than those trained on hard labels.

1 Introduction

In supervised classification, we naturally can not work without labels that are associated with our training data. Obtaining labels, hard or soft, is prone to errors, human or otherwise. That means that a classification algorithm has falsely labeled data in his training set, which, in extreme cases, might render it useless. In this paper, we will evaluate the impact that label noise has on the accuracy of single classifiers, and also multiple classifier schemes.

The training data available for a specific classification task must not necessarily be labeled hard. That is, a training sample might belong, to different degrees, to multiple classes simultaneously. For example, multiple experts might not agree on the diagnosis for the sample, or hear different emotions in a spoken sentence [1,2]. In fact, problems in the field of medical or life sciences, like predicting the secondary structure of proteins [3], often produce and require soft labels. We will compare the noise-resilience of hard-trained versus soft-trained classifiers.

In the following section, we review previous work on the topic of label noise, before presenting a model to artificially add distinct levels of noise. After describing the experimental setup, we present our results. A summary wraps up this article.

2 Previous works

There is not much literature on how label noise should be modeled and dealt with. One exception is Anluin and Lairds paper [4] that details how to embed training data with flipped labels into the framework of Probably Approximately Correct (PAC) learning [5], even allowing malicious noise. They give a lower bound on the number of training samples necessary. The analysis is only applicable to a ”flip” kind of noise and does not hold for more than two classes. Then, some works exist that do present algorithms that can deal with noise, but mostly with the restrictions mentioned above. One example is [6], where the approach is to learn the parameters of the model that generates the noise. Closely related, in [7] the label flip probabilities are incorporated into the training target criterion. The thrust of that paper is different, however, as the setting is semi-supervised learning, where no noise is actively added, but only a faction of the training data has labels associated. Being an extention of the methods proposed by McLachlan in [8], the algorithm changes the labels attributed to the unlabeled training data with each iteration, minimising the modified *Classification Maximum Likelihood* criterion.

We have ourselves previously investigated the impact of noise on classification accuracy, with a focus on fuzzy labels [9]. Our current work uses a more intricate noise model and does not limit the scope of classifiers to Fuzzy KNN.

3 Modelling label noise

We are taking the training data from our sensors as it is, but the class information associated with each object may be erroneous. This we call label noise. To experimentally determine the impact of label noise on classification accuracy, we need to artificially add noise according to a certain model. In a two-class case, a given portion of the training data would get randomly selected and the associated label flipped to the opposite class. This methods extends to the multi-class case, with the label being flipped to one of the other classes in a random manner (as employed in [9]). But since we are dealing with fuzzy labels, where not only one class is given in the label, those noise models are not applicable. Thus, we employ the procedure described in Algorithm 1:

Algorithm 1 Adding noise to labels

```
Input: Normalised labels $label_i$ ($i = 1 \ldots \#$ samples), desired $noise$ level
for all $label_i$ do
  % Generate random label
    $rlabel$ = for each class, draw from uniform distribution in $(0, 1)$
    $rlabel = normalise\_label(rlabel)$
  % Mix original and random label
    $label_i = label_i * (1 - noise) + rlabel * noise$
end for
Output: Normalised labels $label_i$ with added noise
```

This approach models the noise as it could appear in real-world scenarios, and can be understood intuitively. Note that it is important to first normalise the random label and then combine it with the original one using a weighted sum. Normalising only in the end would seriously flatten the fuzzy labels and decrease their variance[1]. It is obviously essential that the original label and the random label be normalised in the same way, or their combination would yield unpredictable results. In our experiments, we simply made the labels to sum up to one.

For certain applications, it might be useful to use a different noise model than the uniform one we employed. For example, one could treat each class label differently, either by assigning a special variance for the randomisation, or using a different noise level for each. A totally different approach to generate the new noised labels would be to see each label as a point on the hyperplane of possible (regarding the normalisation) labels. To add noise, one would advance on this hyperplane into a random direction, with the length of this vector being determined by the desired noise level.

4 Experimental setup

We want to evaluate the impact that noise on the training labels has on the accuracy of single classifiers and multiple classifier architectures. To this end, different levels of artificial noise (see Section 3) are added to the labels. For each level, four basic classifiers, based on different features, are trained, and their decisions combined. Essentially, we are interested in the behaviour of classification accuracy as we increase the level of label noise. The entire experiment is run twice, once with the fuzzy labels, once with hard labels that have been derived from the fuzzy labels. This allows us to see whether it is beneficial to use soft labels, or if hard labels are to be preferred. In all our experiments, conversion of soft labels to hard labels, often called *hardening*, is done via the *winner takes all* rule, also known as *maximum membership* rule. It works by assigning the class with the highest membership value all the weight, and zero to the rest. For example, hardening [0.3 0.4 0.1 0.2] would result in [0 1 0 0].

The fruits data set employed comes from a robotic environment, consisting of 840 pictures from 7 different classes (apples, oranges, plums, lemons,... [10]). Each image was divided evenly into multiple parts (indicated by PxP in the following), the feature values calculated independently for each part and then concatenated. Using the results in [11], we selected the four individual features on which classifiers had the highest accuracy, albeit with the restriction that the features should be fundamentally different. The features selected are (described in detail in [11], dimensions given in brackets): *Colour Histograms* (3x3, 216 dim) in the RGB space. Orientation histograms on the edges in a greyscale version of

[1] We did an experiment on 700 fuzzy labels. Without noise, 428 of them had a class membership with a value above 0.5. After adding 30% noise (*noise* = 0.3), this dropped to 250, flattening the labels a bit. Had we used the normalising only at the end, this value would have significantly dropped to 38. Looking at the mean variance of the labels, the original ones had 5.4e-2, the noised ones 2.7e-2, and for the end-normalised ones it dropped down to 1.2e-2.

the picture. Here we used both the *Sobel* operator (4x4, 128 dim) and the *Canny* (3x3, 72 dim) algorithm to detect edges. As weakest feature, colour histograms in the black-white opponent colour space were calculated ($APQBW$, 2x2, 32 dim). All results were obtained using 5-fold cross validation.

As the data set initially only had hard labels (a banana is quite clearly a banana), we had to convert them to soft labels first. This was done using the fuzzy K-Means clustering [12] algorithm. First, we clustered the data, then assigned a label to each cluster centre according to the hard labels of the samples associated with it to varying degrees. Then, each sample was assigned a new soft label as a sum of products of its cluster memberships with the centres' labels[2].

The basic classifiers we used were Radial Basis Function networks (for the Sobel and Colour Histogram features) and Fuzzy-Input Fuzzy-Output Support Vector Machines (for Canny and APQBW) which we will call F^2-SVMs. Those two classifier types can handle soft training labels, give soft answers and generally show a good classification performance.

The number of kernels in the Gaussian RBF network [13] was set to 47 using a simple heuristic formula, their position determined by running a fuzzy c-means algorithm [12] on the training data. The individual variance of the kernels was set using an experimental observation of Breimann [14], which allows to have only one parameter to optimise for the whole net[3].

The Fuzzy-Input Fuzzy-Output Support Vector Machines [2] employed are, as their name suggests, SVMs able to take training data with fuzzy labels, and to give a fuzzy reply to test samples. The choice of the parameter C common to all machines, which controls the sensitivity to class memberships, had to be determined beforehand using cross validation. The overall architecture is One-Against-One [15], the kernels were polynomials of degree three.

The combination of the classifier's decisions is accomplished using several established Multiple Classifier System architectures in parallel. See [16] for an introduction, and [17] for a comparison of the performance of many methods. We tested the following schemes: Minimum, Median, Average, Dempster-Shafer orthogonal sum rule [18,19,20], Decision Templates ([17], using measure S_1), simple probabilistic product, and an optimal least squares solution calculated using the pseudoinverse. A theoretical comparison of the last three ones can be found in a previous paper of ours [21].

The basic performance measure in our experiments is the classifier accuracy. That is, we harden the soft labels and soft outputs, to find out the agreement between them. There are plenty of other comparison metrics available, a good

[2] Clustering is done for all classes together, using the APQBW feature. After looking at the resulting labels we chose a fuzzifier of 1.3. Hardened fuzzy labels agree with the hard labels in about 80% of the cases.

[3] Using a kernel of the form $f_i(x) = \frac{1}{(\alpha_k d_{i,k})^2} K\left(\frac{x-c_i}{\alpha_k d_{i,k}}\right)$, good results can be obtained if the following ratio including the free parameter α_k is kept constant: $\frac{\alpha_k \overline{d_k}^2}{\sigma(d_k)}$. Here $\overline{d_k}$ is the mean of the kth nearest neighbour distances and $\sigma(d_k)$ their standard deviation.

survey can be found in [22]. The authors of that paper also conceived and tested a very promising new measure, but had to conclude that ”the best course of action to obtain all the accuracy information is to support the interpretation of the descriptive measures with a detailed inspection of the full fuzzy error matrix”. So, as long as there are no other widely accepted methods for comparison, we decided to use classifier accuracy for our graphs.

5 Results

The most important findings of our experiments can be seen in Figure 1: Even the individual classifiers hold up to noise incredibly well. The classifier fusion step is always able to improve over the individual results. Most importantly, classifiers working with the soft labels have higher accuracy.

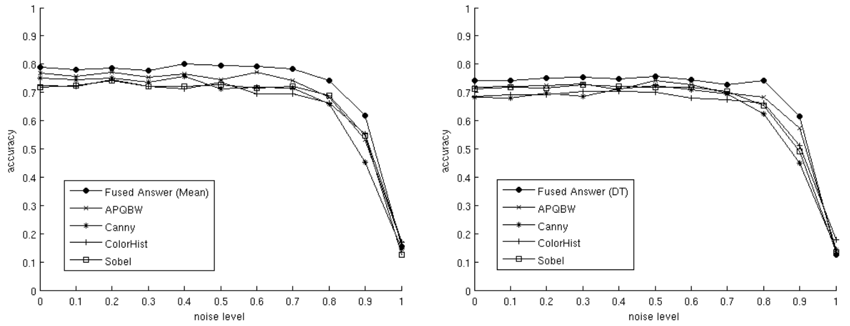

Fig. 1. Behaviour of classification accuracy when adding more and more noise. Shown are plots for the four basic classifiers and their fused answer (method given in brackets). The classifiers for the left plot were trained on soft labels, whereas only hard labels were provided to the ones in the right plot.

A more detailed analysis of the performance of the soft-trained versus the hard-trained classifiers is shown in Figure 2. The individual soft classifiers have, in most cases, a higher accuracy (shown as gain in percent points) than their hard counterpart. Taking the mean value, soft wins. But this advantage gets less and less noticeable once more and more noise is added. The better performance of soft trained classifiers comes, in our opinion, from the ability of the classifiers to take advantage of interdependencies that are encoded in the fuzzy labels, for which we found experimental evidence in separate F^2-SVM-experiments [2] on a dataset of emotional speech [23].

As can be seen rather clearly in Figure 1, the extra fusion step is really worth doing (reasons for this can be found in [16]). The combined answer is always more accurate than even the best single classifier. But of course not all fusion schemes are equally powerful, and when investigating this issue, it turns out one has to make the distinction between the hard-trained and soft-trained setups. In the case with only hard labels, Median, Product, Decision Templates, and Pseudoinverse fusion rules are in the top group, a clear winner for all noise levels[4] can not be declared. For all crossvalidation runs and noise levels, the worst result of the top group is only 6.0 percent points away from the best fusion result for this run (MaxDistanceToBest = 6.0). As for the soft-trained case, only Median and Decision Templates form the top group, with Median being the most stable. The Product rule had to be discarded, as in some single cross validation runs it has outlier drops in accuracy worse than 10 percent points. The MaxDistanceToBest here is very low, only 3.6 percent points.

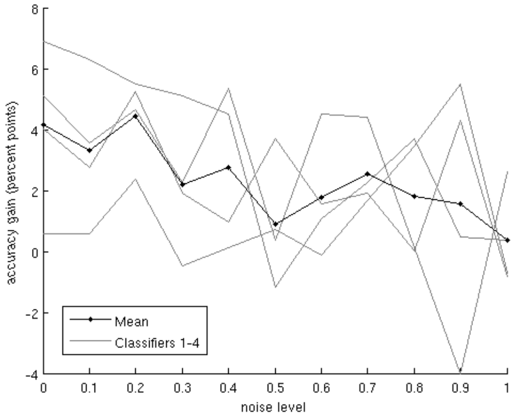

Fig. 2. Accuracy gain in percent points of the four soft-trained classifiers over their hard-trained counterparts. The dotted black line gives the mean value.

One observations strikes as particularly surprising: the high resilience of even single classifiers, trained on hard or soft labels, to added noise. Revisiting Figure 1, we see that despite adding 80% noise on the training labels, the best single classifier still has the very high accuracy of 68%. To put this into perspective, we shall look at the effects of noise on the labels from another angle. If we did not treat them as fuzzy labels, but are only interested in the class with the highest probability, the behaviour shown in Figure 3 comes up. The hardened noisy labels agree pretty much with the hardened original, not noised labels. For example, at the above-mentioned noise level of 80%, there still is agreement of 56.4%, meaning more than half of the samples associated (via hardening) with one class are originally from that class. This seems to still be enough for the classifiers to train reasonably well. So, the fuzzy labels can take quite some amount of such noise before it poses problems in our classification setting.

A short note on fuzzy versus hard in this context: the findings that the classifiers are quite resilient to high label noise levels is only valid for the soft labels. As shown, the noise added does deteriorate a fuzzy label gracefully, and it will take high noise levels until the winning class with the highest probability changes. As the hard labels in this experiment have been derived by hardening the available soft labels, they share this property. Any noise added directly on a hard label, which is only possible using the "flip" rule, would instantly change its

[4] For the fusion architecture selection, we disregarded the noise levels from 80-100% noise, for those cases are not relevant in practical applications and exhibit very volatile behaviour.

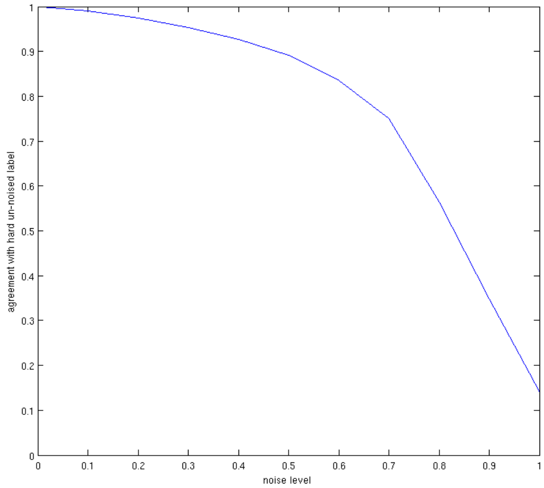

Fig. 3. The x axis gives the level of noise added to the fuzzy labels, as described in Section 3. The y axis shows what portion of the hardened noised labels is still the same as the hardened un-noised labels.

winning class. Unfortunately, a direct comparison of such noise and our model is not possible.

As a more general note, experiments undertaken for this paper suggest that when classifiers are trained with hard labels, their responses should also be taken to be hard, and fused with corresponding schemes, to achieve the highest accuracy.

6 Summary

We investigated the effects of noise that was added to the training labels. Noising was accomplished by calculating the new label as a weighted sum of the original and a completely random label. It turned out that even individual classifiers hold up very well to high noise levels (see Figure 1). Combining several classifiers improves the overall accuracy further, but the right architecture has to be selected. Comparing classifiers trained on soft versus hard labeled data, it turned out that the soft approach is more resistant to noise.

References

1. Steidl, S., Levit, M., Batliner, A., Nöth, E., Niemann, H.: "Of all things the measure is man" - Automatic Classification of Emotions and Inter-Labeler Consistency. In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005. (2005) 317–320
2. Thiel, C., Scherer, S., Schwenker, F.: Fuzzy-Input Fuzzy-Output One-Against-All Support Vector Machines. Proceedings of the 11th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES 2007 (2007)
3. Bondugula, R., Duzlevski, O., Xu, D.: Profiles and Fuzzy K-Nearest Neighbor Algorithm for Protein Secondary Structure Prediction. In Chen, Y.P.P., Wong, L., eds.: Proceedings of the 3rd Asia-Pacific Bioinformatics Conference, World Scientific (2005) 85–94

4. Angluin, D., Laird, P.: Learning from Noisy Examples. Machine Learning **2** (1988) 343–370
5. Valiant, L.G.: A Theory of the Learnable. Commun. ACM **27** (1984) 1134–1142
6. Lawrence, N.D., Schölkopf, B.: Estimating a kernel Fisher discriminant in the presence of label noise. In: Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann (2001) 306–313
7. Amini, M.R., Gallinari, P.: Semi-supervised learning with an explicit label-error model for misclassified data. In: IJCAI03. (2003)
8. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. John Wiley & Sons (1992)
9. El Gayar, N., Schwenker, F., Palm, G.: A study of the robustness of KNN classifiers trained using soft labels. In F., S., S., M., eds.: Artificial neural Networks in Pattern Recognition. Volume 4087 of LNAI., Springer Verlag (2006) 67–80
10. Fay, R., Kaufmann, U., Schwenker, F., Palm, G.: Learning Object Recognition in a NeuroBotic System. In Groß, H.M., Debes, K., Böhme, H.J., eds.: 3rd Workshop on SelfOrganization of AdaptiVE Behavior SOAVE 2004. Number 743 in Fortschritt-Berichte VDI, Reihe 10. VDI (2004) 198–209
11. Fay, R.: Feature Selection and Information Fusion in Hierarchical Neural Networks for Iterative 3D-Object Recognition. PhD thesis, University of Ulm, Germany (2007)
12. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press (1967) 281–298
13. Powell, M.J.D.: Radial basis functions for multivariate interpolation: A review. In Mason, J.C., Cox, M.G., eds.: Algorithms for Approximation. Clarendon Press, Oxford (1987) 143–168
14. Breiman, L., Meisel, W., Purcell, E.: Variable Kernel Estimates of Multivariate Densities. Technometrics **19** (1977) 135–144
15. Kahsay, L., Schwenker, F., Palm, G.: Comparison of multiclass SVM decomposition schemes for visual object recognition. In: DAGM 2005. Volume 3663 of LNCS., Springer (2005) 334–341
16. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley (2004)
17. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: An experimental comparison. Pattern Recognition **34** (2001) 299–314
18. Shafer, G.: Dempster-Shafer Theory. Online http://www.glennshafer.com/assets/downloads/articles/article48.pdf (∼2002)
19. Dempster, A.P.: A generalization of Bayesian inference. Journal of the Royal Statistical Society **30** (1968) 205–247
20. Shafer, G.: A Mathematical Theory of Evidence. University Press, Princeton (1976)
21. Schwenker, F., Dietrich, C., Thiel, C., Palm, G.: Learning decision fusion mappings for pattern recognition. ICGST International Journal on Artificial Intelligence and Machine Learning (AIML) **6** (2006) 17–21
22. Binaghi, E., Brivio, P.A., Ghezzi, P., Rampini, A.: A fuzzy set-based accuracy assessment of soft classification. Pattern Recognition Letters **20** (1999) 935–948
23. Strauss, P.M., Hoffmann, H., Minker, W., Neumann, H., Palm, G., Scherer, S., Schwenker, F., Traue, H., Walter, W., Weidenbacher, U.: Wizard-of-oz data collection for perception and interaction in multi-user environments. In: International Conference on Language Resources and Evaluation (LREC). (2006)