Detecting Mislabeled Data Using Supervised Machine Learning Techniques

Mannes Poel(✉)

Human Media Interaction, University of Twente, Enschede, The Netherlands
m.poel@utwente.nl

Abstract. A lot of data sets, gathered for instance during user experiments, are contaminated with noise. Some noise in the measured features is not much of a problem, it even increases the performance of many Machine Learning (ML) techniques. But for noise in the labels (mislabeled data) the situation is quite different, label noise deteriorates the performance of all ML techniques. The research question addressed in this paper is to what extent can one detect mislabeled data using a committee of supervised Machine Learning models. The committee under consideration consists of a Bayesian model, Random Forest, Logistic classifier, a Neural Network and a Support Vector Machine. This committee is applied to a given data set in several iterations of 5-fold Cross validation. If a data sample is misclassified by all committee members in all iterations (consensus) then it is tagged as mislabeled. This approach was tested on the Iris plant data set, which is artificially contaminated with mislabeled data. For this data set the precision of detecting mislabeled samples is 100% and the recall is approximately 5%. The approach was also tested on the Touch data set, a data set of naturalistic social touch gestures. It is known that this data set contains mislabeled data, but the amount is unknown. For this data set the proposed method achieved a precision of 70% and for almost all other tagged samples the corresponding touch gesture deviated a lot from the prototypical touch gesture. Overall the proposed method shows high potential for detecting mislabeled samples, but the precision on other data sets needs to be investigated.

Keywords: Mislabeled data · Supervised Machine Learning

1 Introduction

A lot of data sets, gathered for instance during user experiments, are contaminated with noise. Some noise in the measured features is not much of a problem, it even increases the performance of many Machine Learning (ML) techniques Quinlan (1986). But for noise in the labels the situation is quite different, label noise deteriorates the performance of all ML techniques (Brodley and Friedl 1999). Hence the detection of mislabeled data is of utmost importance in many ML applications. The research question we address in this paper is "To what

D.D. Schmorrow and C.M. Fidopiastis (Eds.): AC 2017, Part I, LNAI 10284, pp. 571–581, 2017.
DOI: 10.1007/978-3-319-58628-1_43

extent can one detect mislabeled data using a committee of supervised Machine Learning models?"

The outline of the paper is as follows, in Sect. 2 we give an overview on the related work concerning detecting mislabeled data. In Sect. 3 we introduce the methodology for applying supervised Machine Learning techniques in order to detect mislabeled data. The results of this methodology on two cases is presented in Sect. 4. These results are discussed in Sect. 5 and we round off with conclusions and future work in Sect. 6.

2 Related Work

2.1 Statistical Outlier Methods

One of the traditional approaches to outlier detection is to model a mislabeled sample as an outlier and apply standard statistical outlier detection methods Barnett and Lewis (1994). In such an approach the subset of samples with the same label is modelled as a probabilistic (mixture) model $p(x)$ or a kernel based nearest neighbor estimator (Parzen windows) and a threshold θ is determined such that if the likelihood of a sample s is less than the threshold, i.e. $p(s) < \theta$, then the sample s is tagged as a mislabeled sample. Hence the approach is similar to outlier detection in, for instance, medicaid fraud van Capelleveen et al. (2016).

2.2 Local Based Methods

Another local based model compares the label of a sample with the labels of the k-nearest neighbors. Based on the labels of these k-nearest neighbors the label of the sample under consideration can be classified as mislabeled. For instance, if all the k-nearest neighbors have the same label and this label is different from the label of the sample then the label could be considered noise. See for instance the work of Wilson (1972).

2.3 Single Model Based Methods

Two typical examples of the use of single classification models for detecting mislabeled data are Support Vector Machines (SVMs) and Adaboost

Support Vector Machines. The classification by a SVM is determined by an optimal hyperplane in feature space and this hyperplane is completely determined by the so-called support vectors. Moreover these support vectors are a subset of the training set (Bishop 2007). The intuition behind these support vectors (data points) is that they are close to the boundary between the different classes and are hard to classify. Hence these support vectors could be an indication of mislabeled data points. This is the approach explored by Ekambaram et al. (2016). They showed that the support vectors can reduce the search space for mislabeled data. These support vectors are deleted from the training set and

after retraining the generalization performance on the test set is evaluated. The main goal was to investigate the performance but not the detection of mislabeled data. But one can deduce in advance that the precision for detecting mislabeled data is low due to the fact that a SVM always has support vectors and thus also in cases where all data is correctly labeled.

Adaboost. Adaboost (Freund et al. 1996) is an iterative algorithm which constructs a strong committee classifier based on weak classifiers. It is an iterative procedure in which hard to classify examples get more weight than easy to classify examples. In each iteration a new weak classifier is trained based on the weights of each data sample. The weight of the data sample denotes the relative contribution of this data sample to the error function. After training, the data samples with the highest weights are those data samples which are hard to classify by the committee. The approach discussed in (Cao et al. 2012) is that mislabeled data are in general hard to classify and thus get high weights when applying Adaboost. These examples are deleted from the training set and after retraining the performance increase on the test set is evaluated. The main goal of this study is to evaluate the performance after deleting data samples with high weights (potentially mislabeled data) and does not focus on precision or recall. Once again this reduces the search space for mislabeled data but the precision is low.

2.4 Ensemble Based Methods

The use of ensemble based methods for detecting mislabeled data is already discussed in the paper by Brodley and Friedl (Brodley and Friedl 1999), which is one of the first papers on using Supervised Machine Learning for the detection of mislabeled data. The ensemble they used consisted of Decision Trees, k-Nearest Neighbor and Linear Discriminant. Moreover the difference between majority voting and consensus to tag an instance as mislabeled was investigated. The approach was empirically validated on 5 data sets. The overall conclusion was that detecting and filtering out the data tagged as mislabeled improves the generalization performance, but if the mislabel rate is too high then, as expected, the method breaks down. The main focus was to investigate the generalization performance but they also looked into the precision of their method, by introducing artificial noisy labels. As expected the consensus approach has higher precision than majority voting, but the precision was not optimal. This means that sometimes samples with correct labels (probably the hard to classify samples) were discarded from the training set. A more extensive overview of classification in the presence of mislabeled data can be found in Frénay and Verleysen (2014).

The focus of the research described in this paper is different: the main focus here is on precision and recall of mislabeled data using an ensemble based method. The overall aim is to detect mislabeled samples with high precision and in a second step, not covered in this paper, to remove the samples from the training set or relabel these samples, for instance by manual inspection, in order to improve classification performance.

3 Methodology

Recall that we want to investigate in how far one can detect mislabeled data using supervised ML techniques. Normally mislabeled data is detected by humans (visually) analyzing the data and checking wether for a given data sample the corresponding label is correct. This analysis can be supported by automatic clustering techniques (an unsupervised technique) to detect suspicious (data sample, label) pairs. In this paper we follow a different approach. The idea is to replace the human inspectors by a committee of trained supervised Machine Learning models, such as Bayesian Classifiers, Random Forest and Logistic regression. If a data sample is misclassified by all models then this sample is tagged as suspicious. These models are trained in the given data set, which in theory contains mislabeled data, so they are not optimal. To compensate for this non optimality we train and apply the models several times using k-fold Cross Validation (k-CV). Since we want to detect mislabeled data, high performance of a model is not required, but of course it is an advantage. Moreover investigating data samples can be very time consuming, for instance if one has to check the video or sound recordings, so we strive for a high precision as opposed to high recall with many false positives. The overall method is as follows. Select a diverse set of supervised ML models, the more diverse the more independent the models are. In our case we selected a Bayesian model, Random Forest, Logistic classifier, a Neural Network and a Support Vector Machine. Next train and apply the models to the given data set using k-CV, and determine how many times a data sample is misclassified. For one run of k-CV and five models the maximum number of misclassifications is also five. This result could depend on the partition of the k-folds and moreover the training of the ML models is non-optimal due to the presence of mislabeled data. Hence we repeat this k-CV approach several times, say 10 times, and data samples which are misclassified all the time (in this case $5 \times 10 = 50$ times) are flagged as suspicious and are candidates for future investigation by for instance human experts. We investigated the precision and recall of the above approach on the well-known Iris Plants data set Fisher (1936) by introducing mislabeled data by randomly switching the label of randomly selected data samples. Afterwards we tested our approach on the touch data set Jung et al. (2016), a data set more prone to mislabels. For this touch data set the mislabels are unknown and hence we only focus on precision.

3.1 Iris Data Set

The Iris Plant data set is a well known (toy) data set for validating Machine Learning methods, see Fisher (1936). The data set consists of 150 samples of three different Iris species and each sample coded by four numerical features; sepal width, sepal length, petal width and petal length. First we test our approach on the original data set and afterwards introduce artificial mislabeled data in order the analyse the precision and recall of our methodology. Randomly 1 up to 30 data samples were selected for which the label was randomly switched

to another class. Since we randomly select mislabeled data we repeated each experiment of introducing mislabels twenty times.

3.2 Touch Data Set

The touch data set, introduced in Jung et al. (2016), is a corpus consisting of social touch gestures. The main goal of this data set (corpus) is to work towards a data driven approach for touch recognition and benchmarking. The corpus was constructed by a user experiment in which 32 subjects had to perform 14 different social touch gestures in three different variations (gentle, normal and rough) on a sensor grid wrapped around an artificial mannequin arm, see Fig. 1 for how the user experiment was conducted. The participants first saw a movie in which

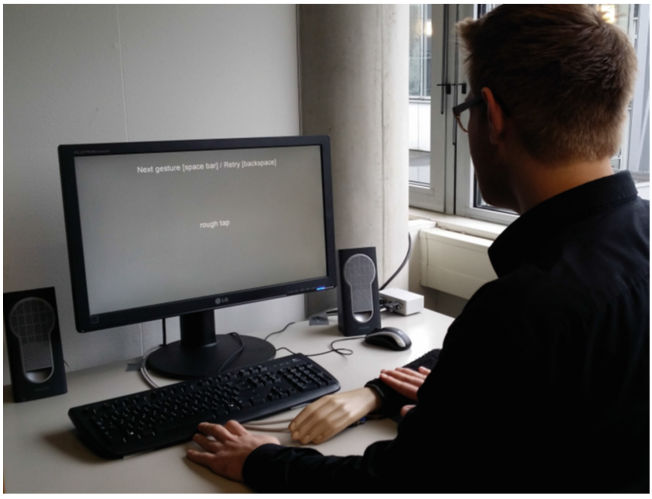


Fig. 1. Setup of the touch experiment.

the 14 different touch gestures on the mannequin arm where demonstrated. In order to practice participants were instructed to repeat every gesture. After this demonstration and practice each subject had to perform 252 gestures in random order. The reason to show the instruction at the start of and not during the experiment was to allow for individual freedom in performing the requested touch gestures. But this also led to forgetting the instruction, being uncertain about the gesture to be performed or recalling the wrong instruction, and therefore not performing the right touch gesture, leading to a corpus containing mislabeled data.

The experiment was video recorded and this gives an opportunity to inspect the label of a given recorded touch gesture. But this is a very time consuming procedure. For this research we focus only on the rough touch gestures in this data set, in total 2602.

For our analyses we first normalized the features between −1 and 1 and afterwards projected the normalized features on the first 12 principal components. We ran the experiments for 1000 iterations and each iteration applied a 5-fold CV for all the 5 members of the committee of classifiers. This gives the opportunity to investigate the effect of the number of iterations on the number of tagged samples. Afterwards a random subset of 20 samples is used to investigate the precision of our approach. Observe that we cannot assess the recall of our approach on this data set because the subset of mislabeled samples is unknown.

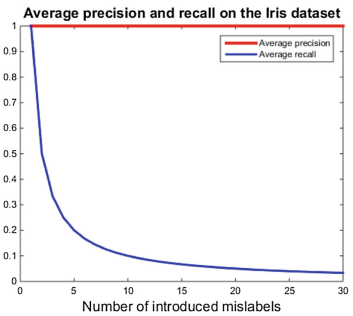
4 Results

We tested our approach on the Iris Plant and Touch data sets. The results are reported below.

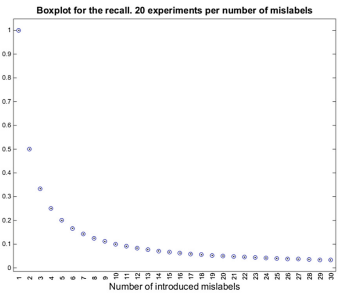
4.1 Iris Plant Data Set

If we apply our methodology with 10 iterations of 5-CV no data samples were flagged as misclassified.

The maximum number misclassifications was 8 and in order to be flagged a data sample must be misclassified 10 times. This makes the Iris Plant data set a good candidate for a supervised approach by introducing mislabeled data. We randomly selected between 1 and 30 data samples for which we randomly switched the label to another class. Since we randomly introduce mislabeled data we repeated each experiment of introducing mislabels twenty times. The results for precision and recall can be found in Fig. 2.



(a) The average precision and recall over 20 runs.



(b) Box plot for the recall.

Fig. 2. The average precision and recall over 20 runs, including a boxplot for the recall.

Observe that the average recall is dropping but stays above zero, but the precision is always one. This means that the data samples flagged by our method are in the set of mislabeled data samples. It also follows that by an iterative

approach all the mislabeled data can be detected. Since the precision is one we can remove or relabel the flagged data points. For this new data set the number mislabeled is less and it follows from Fig. 2(a) that the precision is still 1. Hence we can apply the method over and over again, removing or relabel the flagged data samples. We will end up with a data set with no mislabeled data. Wether this also holds for other data sets needs be investigated.

4.2 Touch Data Set

On the Touch data set we applied 1000 iterations of 5-CV. In each iteration the performance of the classifiers was between 55 and 60% and on average 550.7 samples were tagged in each iteration with a standard deviation of 11.2. The development of the number of samples which were always misclassified is depicted in Fig. 3.

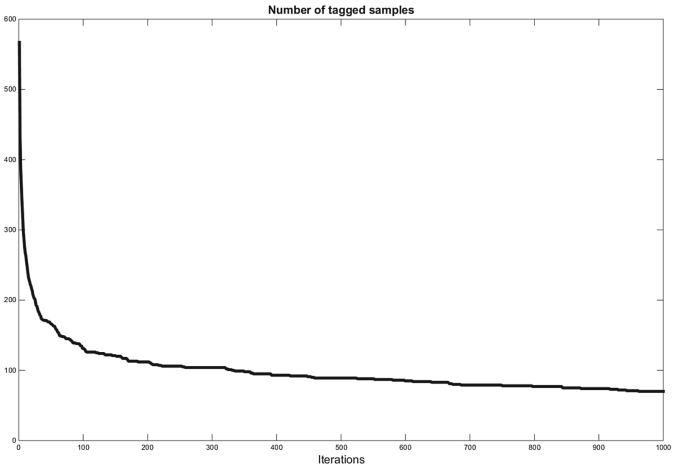


Fig. 3. Number of tagged samples plotted against the number of iterations.

Observe that the curve flattens after approximately 200 iterations and after 1000 iterations 70 samples are tagged as mislabeled. Some statistics on the tagged samples, the number of tags per gesture are: *grab* 3, *hit* 10, *massage* 3, *pat* 10, *pinch* 5, *poke* 2, *press* 9, *rub* 6, *scratch* 3, *slap* 4, *squeeze* 4, *tap* 5 and *tickle* 0. Meaning that only three gestures – *hit*, *pat* and *press* – are responsible for 29 out of 70 (41%) tagged samples. With respect to the subjects, subject 23 was responsible for 7 of the tagged examples, subject 27 for 6 and subject 9, 21 and 30 for 5. Meaning that only 5 out of 32 subjects are responsible for 40% of all tagged samples. A factor analysis reveals that for subject 9 the gesture *pat* was tagged 3 times and for subject 23 the gesture *grab* was tagged 3 times, so all tagged *grab* gestures are from this subject.

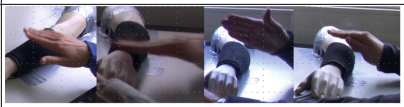
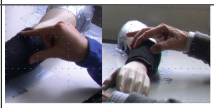
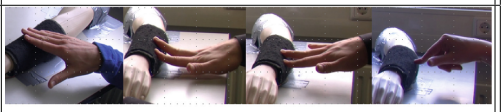
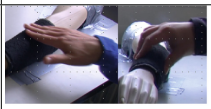
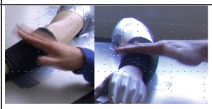
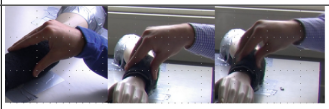
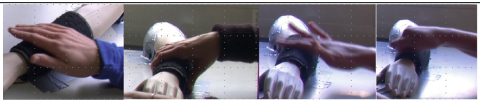
From these 70 tagged samples we randomly took 20 samples for further inspection, meaning watching the video in order to detect what actually happened, see Table 1. For each tagged gesture we have depicted the instruction as the first picture in the row followed by pictures showing the most characteristic (distinctive) property of the tagged gesture.It is clear from these screenshots that for the *grab* gesture the tagged sample is substantially different from the instruction touch, it is in principle a *tickle* gesture. For the *hit* gesture the first two are substantially different from the instruction (the prototypical gesture), not using a fist but the flat hand, but the last one seems to resemble the instructions a lot. Looking at the *massage* gesture one can clearly see that the participant is not focusing purely on performing the gesture but is also reading from a piece of paper. Moreover the performed gesture resembles a stroke. For the *pat* gesture it is hard to see from the screenshots if there is a substantial difference between the instruction and the performed gesture. The instruction is two times a vertical pat on the sensor grid, inspection of the videos reveals that in the tagged samples there are not two pats but only one and there is also a sliding movement. This can also be clearly seen in the third screenshot. The first two tagged *pat* gestures resemble the *slap* gesture more closely and the last one more (three times) a *stroke*. The tagged *pinch* gesture seems not to differ that much from the instruction, so the tag could be wrong. For all the three tagged *press* gestures the difference with the prototypical instruction is clear. The *rub* instruction is an up and down movement orthogonal to the arm, the tagged *rub* gesture does not contain this up and down movement and more resembles a *tickle* or a *scratch*. The first tagged *scratch* gesture (second in the row in Table 1) is just a hit using the fingers, no scratching behavior. The second one is a scratch in the direction of the arm, so it is a scratch but in a different direction than in the instruction. For the tagged *slap* gesture almost no difference with the instruction can be detected, only almost no sidewards movement, so it is not clear why this sample was tagged as mislabeled. The *squeeze* instruction is a squeeze without moving the fingers, in both tagged *squeeze* gestures the subject moved the fingers towards each other, but it can still be considered a *squeeze* but not prototypical (similar to the instruction). For the last gesture to analyse, the *stroke*, it is already clear from the screenshot that the first one is mislabeled, it is a *grab*. The second gesture is depicted by two screenshots, the first part is a *stroke* but only using two fingers and the last part is a stroke using the fist. So it is more a sequence of different types of *strokes*, first two fingers and afterwards the fist.

Summarizing, for the sample of 20 tagged samples 14 tagged samples were clearly mislabeled, a precision of 70%. For most of the other tagged samples the touch gesture differs a lot from the prototypical instruction but still could be considered as labeled correctly.

5 Discussion

The goal of the study was to construct a method based on a committee of different supervised ML models to detect mislabeled samples in corpus. The

Table 1. Screenshots per touch class. Touch instruction (first in the row) and then the tagged samples. The last two screenshots in the last row (*stroke* gesture) are from the same gesture.

Gesture label	Screenshots
grab	
hit	
massage	
pat	
pinch	
press	
rub	
scratch	
slap	
squeeze	
stroke	

committee used consists of five different types of models, a Bayesian model, Random Forest, Logistic classifier, a Neural Network and a Support Vector Machine. The reason for selecting different types of models was to assure that the models are independent. But it is not clear in how far the models are really independent because they are trained on the same data set and how far the results depend on the chosen models. First we tested the approach on the Iris data set. Initially no mislabeled samples were detected, but this was not validated by experts or visual inspection of the data set. The next step was to contaminate this data set with artificially mislabeled data. These artificially mislabeled samples were detected with high precision and low but sufficient recall, see Fig. 2(a). This makes it possible to remove all mislabeled samples from the data set.

But for the Touch data set the situation is quite different. There are mislabeled samples in the data set, as can be seen in the results described in Sect. 4, but the exact set of mislabeled samples is unknown and also depends on personal interpretation. In the construction of the Touch data set the participants were given instructions on how to perform the touch gesture at the start of the experiment, on the one hand to give the participant a clue about the gestures to perform and on the other hand to allow for personal interpretation and freedom. This is for instance clear from one of the tagged *scratch* gestures, which is definitely a scratch but differs in the direction from the instruction. If in all other trials for the *scratch* gesture the participants followed the instruction then this tagged *scratch* gesture is quite different from the prototypical *scratch* gesture and hence always misclassified. But this instruction at the start can also cause the participants to make a mistake in the recall of the instruction and perform a different touch gesture, as can be seen in one of the tagged *grab* gestures which is a *tickle*. One can conclude that in a naturalistic setting such as the investigated Touch corpus the correctness of the labeling depends on the personal interpretation of the touch label by the participant (personal freedom of interpretation) and the goal for and methodology by which the corpus is constructed.

In the related work described by Brodley and Friedl (1999), Ekambaram et al. (2016), Frénay and Verleysen (2014), Guan and Yuan (2013) the main focus is classification performance on the test set by removing tagged mislabeled samples from the training set. Hence it is hard to compare our findings with their results.

6 Conclusions

It is known that mislabeled data can adversely affect the performance of supervised ML methods. In order to detect such mislabeled data samples we proposed a detection method based on a committee of five different supervised ML models; a Bayesian model, Random Forest, Logistic classifier, a Neural Network and a Support Vector Machine. The precision of this committee was evaluated on the Iris and Touch data sets. The results show that on the contaminated Iris data set the committee had always a precision of one and the recall was low, around 10%, but sufficient.

On the Touch data set the situation is quite different due to the nature of the data set. The generation of the data set allows for personal freedom and

interpretation of touch gestures. In total 70 samples were tagged as potentially mislabeled. A random sample of 20 tagged instances was selected and evaluated by inspecting the corresponding videos. The result was a precision of 70%. For most of the other tagged samples the touch gesture differs a lot from the prototypical instruction but still could be considered as labeled correctly.

Overall the proposed method shows high potential for detecting mislabeled samples, but the precision on other data sets needs to be investigated.

Acknowledgments. This work was partially supported by the Dutch national program COMMIT.

References

Barnett, V., Lewis, T.: Outliers in Statistical Data. Wiley, Chichester (1994)

Bishop, C.: Pattern Recognition and Machine Learning, 2nd edn. Springer, New York (2007)

Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. J. Artif. Intell. Res. **11**, 131–167 (1999)

Cao, J., Kwong, S., Wang, R.: A noise-detection based AdaBoost algorithm for mislabeled data. Pattern Recognit. **45**(12), 4451–4465 (2012)

van Capelleveen, G., Poel, M., Mueller, R.M., Thornton, D., van Hillegersberg, J.: Outlier detection in healthcare fraud: a case study in the medicaid dental domain. Int. J. Account. Inf. Syst. **21**, 18–31 (2016)

Ekambaram, R., Fefilatyev, S., Shreve, M., Kramer, K., Hall, L.O., Goldgof, D.B., Kasturi, R.: Active cleaning of label noise. Pattern Recognit. **51**, 463–480 (2016). http://www.sciencedirect.com/science/article/pii/S0031320315003519

Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**(2), 179–188 (1936)

Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 845–869 (2014)

Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. ICML **96**, 148–156 (1996)

Guan, D., Yuan, W.: A survey of mislabeled training data detection techniques for pattern classification. IETE Tech. Rev. **30**(6), 524–530 (2013)

Jung, M.M., Poel, M., Poppe, R., Heylen, D.K.J.: Automatic recognition of touch gestures in the corpus of social touch. J. Multimodal User Interfaces **11**, 1–16 (2016). http://dx.doi.org/10.1007/s12193-016-0232-9

Quinlan, J.R.: Induction of decision trees. Induction Decis. Trees Mach. Learn. **1**(1), 81–106 (1986)

Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. **3**, 408–421 (1972)