Latent Class Logistic Regression

Matthew Arnold
Department of Computer Science
University of Toronto
Toronto, ON, Canada
matthewmatical@gmail.com

Rafael Lacerda
Department of Computer Science
University of Toronto
Toronto, ON, Canada
lacerda@cs.toronto.edu

Abstract

Noisy labels are a common sight in real world data. Basic models such as logistic regression assume correct labels and could overfit to accommodate incorrect labeling. We propose an extension to logistic regression that is robust to noisy labels by incorporating a latent variable that corresponds to the real class and can be used to identify mislabeled examples.

1 Introduction

Clean datasets such as MNIST are suggested as minimal effort solutions for learning techniques and pattern recognition while avoiding preprocessing efforts [1] and are commonly used in the machine learning community to evaluate model performance. In practice, real world data tends to be noisy [2][3] and may contain a variety of errors [2]. These errors can negatively affect the performance of supervised classification models [4]. Label noise in particular is potentially more harmful in this sense than feature noise [5][6].

Three kinds of label noise have been identified in literature: (1) Noisy Completely at Random (NCAR), where a labeling error is independent of random variables and the true class. In this case, mislabeled instances are evenly distributed among the true classes. (2) Noisy at Random (NAR) errors are still independent of the random variables, but dependent on the true class. (3) Noisy Not at Random (NNR) errors are dependent on the random variables, that is, misclassification may occur for similar data points [5].

The fact that cheaper data annotation methods can be used at the expense of increasing label noise is worth considering. By using models that are robust to label noise, noisy data can be relabeled, increasing the quality of annotation. Thus, such models allow for lower annotation costs while attenuating its impact on accuracy [7].

To address these issues, we propose modifying the standard regularized multinomial logistic regression model to include a latent variable for the true label. By learning what the data should look like for a given class, a discriminator can estimate class probabilities when evaluating a new example. Information on the given label as well should not be discarded, as it may contain valuable information regarding systematic errors and use it to learn the distribution of true labels given the original label (Figure 1).

Developing models that are robust to label noise is an active field of research with an extensive body of work. Some approaches attempt to train multiple classifiers on different folds of the data, flagging data points where there is too much disagreement [8][9][10]. Other approaches attempt to cluster similar data points with no supervision and then infer the true class label by the noisy annotation [11][12].

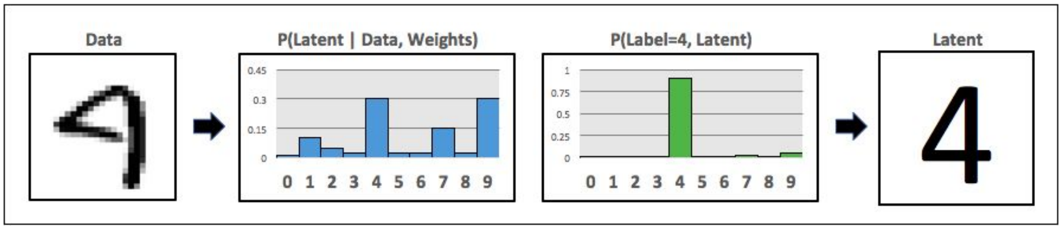

Figure 1: Using an ambiguous MNIST digit (4 or 9) as an illustrative example, the latent class is inferred by the data as well as the given label. Arrows indicate computational flow.

2 Related Work

Another common solution is to model the label noise and handle annotation errors during training [3], which includes our own proposal.

Tibshirani and Manning present an extension to binary logistic regression that incorporates the possibility of mislabeling into the objective function by introducing "shift parameters" that change datapoints' classes if appropriate. This modification adds one extra parameter per datapoint and allows each one to be flipped to the perceived class [3]. This maintains simplicity in the logistic regression model, but seems to be restricted to the binary case.

Bekker and Goldberger approached the issue of training a neural network on data with label noise by assuming that the true labels pass through a noisy channel with unknown parameters. The initial weights are generated by a typical neural network, which are fed into the expectation-maximization algorithm. The E-step incorporates true label estimation dependent on a noise model, which is updated in the subsequent M-step, followed by maximization of the neural network parameters. By modeling the label noise, they achieve results superior to those of a standard neural network in NCAR and NAR noise taxonomies [13]. However, there is no closed form for the loss function defined. The neural network is completely detached from the noise model and only updates its parameters after a few iterations of EM.

Xiao et al. deal with the same issue by using a set of CNNs to learn the labels and the latent classes. To bridge the gap, the label noise is also modeled as a latent variable, taking into account two kinds: mislabeled cases where the input data is ambiguous (NNR) and pure random mislabeling (NCAR) [7].

3 Latent Class Logistic Regression

Our proposed model, Latent Class Logistic Regression (LCLR) is based on the standard multinomial logistic regression model, defined as (1):

$$P(c \mid x, w)_{standard} = \frac{exp(w_c^T x)}{\sum_{c'=1}^{K} exp(w_c^T x)} \qquad (1)$$

Where weights w are the logistic regression parameters, data points are x, K is the number of classes and c is the class index. We propose inclusion of a latent variable r that models the true class. Our model of $P(c \mid x, w)$ is the result of application of a noisy channel $P(c \mid r)$ to r (3):

$$P(c, r \mid x, w) = P(c \mid r) \cdot P(r \mid x,\ w) \qquad (2)$$

$$P(c \mid x, w) = \sum_{r}^{K} P(c \mid r) \cdot P(r \mid x, w) \qquad (3)$$

Our chosen model for $P(c \mid r)$ is a K-by-K matrix Θ (4), and our chosen model for $P(r \mid x, w)$ is logistic regression (5). Optimal values for Θ and w are obtained by first initializing w with naive logistic regression, then by performing gradient descent on the negative log-likelihood of $P(c \mid x, w)$ over the dataset.

$$P(c \mid r) = \Theta_{cr} \qquad (4)$$

$$P(r \mid x, w) = \frac{exp(w_r^T x)}{\sum_{r'=1}^{K} exp(w_{r'}^T x)} \qquad (5)$$

Through application of Bayes' rule, we can model r in terms of our knowledge of the noisy channel (6), increasing the evidence used for evaluating our predictions after training.

$$P(r \mid c, x, w) = \frac{P(r, c \mid x, w)}{P(c \mid x, w)} \qquad (6)$$

$$P(r \mid c, x, w) = \frac{P(c \mid r) P(r \mid x, w)}{\sum_{r'=1}^{K} P(c \mid r') \cdot P(r' \mid x, w)} \qquad (7)$$

Unlike Bekker & Goldberger as well as Xiao et al., our classifier is a self-contained logistic regression and makes no use of neural nets. Further in comparison to Bekker and Goldberger's approach, LCLR can be optimized through gradient descent, as the noise model is incorporated directly into the loss function. Compared to Tibshirani's solution, this classifier is multinomial and does not include parameters for each data point.

4 Experiments

To assess model performance, we compare the accuracies of LCLR and standard logistic regression on the MNIST data set. We generate random noise and systematic noise at varying levels. Detailed accuracy values for the experiments are available in the appendix. All models were trained with gradient descent using the *Autograd* Python library [14].

4.1 NCAR Label Noise

Our first experiment was to corrupt the dataset labels with NCAR with error rate ε evenly among classes, independent of data. The motivation behind this is to compare the accuracy of standard logistic regression and LCLR given different noise levels. Figure 3 shows that LCLR slightly but consistently outperforms standard logistic regression with noise values up to 50%, but underperforms for values above.

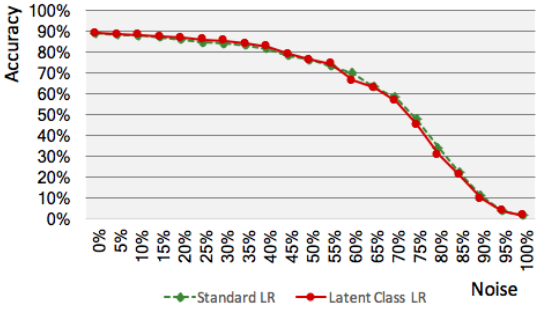

Figure 3: Comparison of test set accuracy between standard logistic regression and the latent variable modification under increasing levels of NCAR noise.

4.2 NNR Label Noise

Noisy Not at Random noise, or "systematic" noise was generated by choosing an arbitrary constant $\alpha = 0.75$ and three arbitrary pairs of plausibly confusable digits (4 and 9, 1 and 7, 3 and 8). Given an error rate ε, the first digit in each pair will corrupt to the other with probability $\varepsilon \cdot \alpha$, and the second digit in each pair will corrupt to the other with probability $\varepsilon \cdot (\alpha - 1)$. NNR noise is an 'ideal case' for our model, since the proportions it generates correspond directly to entries in the θ matrix. Given that we look at a subset of 60% of digits, select for possible corruption with frequency ε, then corrupt 50% of the digits on average, 30% noise is the maximum. Figure 4 shows that, while unstable, LCLR outperforms standard logistic regression given NNR noise in most cases — especially at high noise levels.

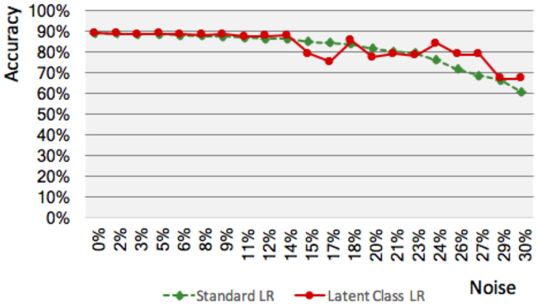

Figure 4: Comparison of test set accuracy between standard logistic regression and the latent variable modification under increasing levels of NNR noise.

5 Discussion

While LCLR has the potential to outperform standard logistic regression, the difference in classification accuracy is almost nothing under random label noise (NCAR). The optimal θ under NCAR noise is the identity matrix, which effectively reduces the model to simple logistic regression. This validates our initial perception that LCLR can only improve accuracy under systematic noise (NNR). An extension of our model that learns a more complicated noisy channel, such as $P(c \mid r, x)$ may be able to better model NCAR noise.

Since we optimize using logistic regression, the model must be reparameterized slightly to ensure that the θ matrix entries correspond to valid probabilities: θ columns must contain positive real numbers between 0 and 1 that sum to 1. Our chosen softmax reparameterization guarantees this, but at a cost: probability mass is spread very liberally across the matrix (for example, an entry with probability 0.000001 could become 0.06). This hurts the performance of the model, and is likely to blame for several of our observed instabilities.

We have developed a modification to standard logistic regression that is robust to systematic errors, which could allow for cheaper data annotation with little impact on accuracy. This classifier can also be used to flag labels as noisy where there is disagreement between the classifier's prediction and the label as highlighted by Xiao et al. [7].

Due to our overly simplistic model of the noisy channel $P(c \mid r)$, we believe that using a more complex noisy channel model will improve performance. For example, a convolutional neural network could be used to model $P(c \mid r, x)$, similar to the solution developed by Xiao et al. [7].

This classifier might also be used in ensemble methods such as those in [8][9][10] to further eliminate noise from mislabeled data.

References

[1] LeCun, Yann, Corinna Cortes, and Christopher JC Burges. "The MNIST database of handwritten digits." (1998).

[2] Maletic, Jonathan I., and Andrian Marcus. "Data Cleansing: Beyond Integrity Analysis." Iq. 2000.

[3] Tibshirani, Julie, and Christopher D. Manning. "Robust Logistic Regression using Shift Parameters." ACL (2). 2014.

[4] D. Nettleton, A. Orriols-Puig, and A. Fornells, “A study of the effect of different types of noise on the precision of supervised learning techniques,” Artificial intelligence review, 2010.

[5] Frénay, Benoît, and Michel Verleysen. "Classification in the presence of label noise: a survey." IEEE transactions on neural networks and learning systems 25.5 (2014): 845-869.

[6] X. Zhu and X. Wu, “Class noise vs. attribute noise: A quantitative study,” Artif. Intell. Rev., vol. 22, no. 3, pp. 177–210, 2004.

[7] Xiao, Tong, et al. "Learning from massive noisy labeled data for image classification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[8] Brodley, Carla E., and Mark A. Friedl. "Identifying mislabeled training data." Journal of artificial intelligence research 11 (1999): 131-167.

[9] Verbaeten, Sofie, and Anneleen Van Assche. "Ensemble methods for noise elimination in classification problems." International Workshop on Multiple Classifier Systems. Springer Berlin Heidelberg, 2003.

[10] Venkataraman, Sundara, et al. "Distinguishing mislabeled data from correctly labeled data in classifier design." Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on. IEEE, 2004.

[11] Rebbapragada, Umaa, and Carla E. Brodley. "Class noise mitigation through instance weighting." European Conference on Machine Learning. Springer Berlin Heidelberg, 2007.

[12] Rebbapragada, Umaa, et al. "Improving onboard analysis of Hyperion images by filtering mislabeled training data examples." Aerospace conference, 2009 IEEE. IEEE, 2009.

[13] Bekker, Alan Joseph, and Jacob Goldberger. "Training deep neural-networks based on unreliable labels." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.

[14] Maclaurin, Dougal, David Duvenaud, and Ryan P. Adams. "Autograd: Reverse-mode differentiation of native python." ICML workshop on Automatic Machine Learning. 2015.

Appendix

Table 1:

Model accuracy for random noise levels

Noise	Standard LR	Latent Class LR
0%	0.8909	0.8912
5%	0.8836	0.886
10%	0.8794	0.8828
15%	0.871	0.8745
20%	0.8603	0.8679
25%	0.8474	0.8597
30%	0.8414	0.8553
35%	0.8331	0.8404
40%	0.8158	0.8259
45%	0.7819	0.7922
50%	0.764	0.7631
55%	0.732	0.7424
60%	0.6996	0.6631
65%	0.6317	0.6296
70%	0.583	0.5647
75%	0.4772	0.4519
80%	0.3404	0.3053
85%	0.2198	0.2073
90%	0.1079	0.0938
95%	0.0361	0.036
100%	0.0122	0.0121

Table 2:

Model accuracy for systematic noise levels

Noise	Standard LR	Latent Class LR
0%	0.8909	0.8912
2%	0.8888	0.8876
3%	0.8865	0.8854
5%	0.8868	0.8888
6%	0.8808	0.8864
8%	0.8781	0.8833
9%	0.8757	0.8851
11%	0.8696	0.8757
12%	0.8632	0.877
14%	0.8624	0.8804
15%	0.8484	0.7911
17%	0.8436	0.7516
18%	0.8385	0.8545
20%	0.8153	0.7738
21%	0.8025	0.7911
23%	0.7931	0.7821
24%	0.7609	0.84
26%	0.7148	0.7879
27%	0.6864	0.7881

29%	0.6622	0.6719
30%	0.6031	0.6703