Learning Fashion Traits with Label Uncertainty

Assaf Neuberger
Amazon Lab 126
Herzliya, Israel 4672560
neuberg@amazon.com

Eli Alshan
Amazon Lab 126
Herzliya, Israel 4672560
alshan@amazon.com

Gal Levi
Amazon Lab 126
Herzliya, Israel 4672560
gallevi@amazon.com

Sharon Alpert
Amazon Lab 126
Herzliya, Israel 4672560
alperts@amazon.com

Eduard Oks
Amazon Lab 126
Herzliya, Israel 4672560
oksed@amazon.com

ABSTRACT

We consider the task of predicting subjective fashion traits from images. Specifically, we are interested in understanding which outfit actually better suites the user. Since these traits are highly subjective, they tend to be noisier. One solution is to annotate each example several times, but this makes it hard to collect large amounts of data. So, for practical reasons, large data sets have only a few human annotations for each example. This approach introduces sampling uncertainty since labels are estimated using only a small set of human annotations. In this paper, we provide a closed-form expression to model the label uncertainty induced by sub-sampling. We show that for fashion related traits our model can basically quantify the ability of a learning algorithm to learn from noisy data. We further use this model to construct a custom neural network loss function which is able to better learn fashion traits.

KEYWORDS

label noise, fashion, aesthetics

ACM Reference format:
Assaf Neuberger, Eli Alshan, Gal Levi, Sharon Alpert, and Eduard Oks. 2017. Learning Fashion Traits with Label Uncertainty. In *Proceedings of KDD2017 Fashion Workshop, Halifax, Nova Scotia, Canada, August 2017,* 5 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

In recent years there has been a significant performance improvement in image classification tasks. The improvements can be attributed to the combined effect of advances in Deep Convolutional Neural Networks and the availability of large annotated data sets. Almost all of the large datasets (as shown by [3, 4]) depict objective traits such as: planes, cars or animals. Therefore, the the collection of large amounts data can be done reliably using various crowd sourcing or automated tools. In contrast to objective traits, there are only a handful of small data sets with labeled subjective traits. In our context, subjective traits are traits that dependent on the annotator's personal preference, such as fashion or aesthetics.

The key problem in obtaining data for subjective traits is that the annotations provided by humans are influenced by subjective consideration and are therefore noisier than their objective counterparts. To increase the labels reliability, each example must be annotated numerous times by different annotators. For instance, in [1] each example was annotated, on average, by 205 human annotators, each providing a feedback on the aesthetics of the image. Acquiring these datasets poses a huge challenge for training complex models like deep convolutional networks, as it requires several orders of magnitude more annotations.

An alternative approach is simply to sub-sample the panel of annotators. Namely, for each example only a subset of annotators provides their vote one each image. The size of the subset might vary depending on budget or systematic constraints. This approach indeed reduces the number of annotations. However, this approach introduces sampling uncertainty in the labels that varies from one labeled example to the other. This makes it much harder for a learning algorithm to obtain good enough performance on the test set [15].

In this paper, we deal with the problem of learning subjective fashion traits from sub-sampled data. Specifically, we deal with the problem of selecting the better outfits out of two different outfits. We provide a closed-form expression to model the label uncertainty induced by sub-sampling. Using this model, we present two different contributions. First, we examine the impact of the label noise on the train and test sets, as well as derive upper bounds on classification accuracy. Second, we train a neural network with a dedicated loss function that simultaneously predicts the label and its uncertainty. We demonstrate that by constructing the network this way, we can better estimate the quality or fashionability of an outfit.

2 RELATED WORK

Automatic detection of subjective or social traits has been a key research area in computer vision for quite a long time. Among some of the relevant works are subjective image aesthetics and emotion [1], facial attractiveness [8] and evaluation of facial beauty [6].

Handling noisy labels has also been a significant area of research in the last years. This is due to the fact that large annotated datasets rely on various crowd-sourcing platforms to obtain ground-truth labels. A common approach for dealing with noise is to model

KDD2017 Fashion Workshop, August 2017, Halifax, Nova Scotia, Canada

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

the true label and other factors influencing the labeling process as latent variables. Then, an inference algorithm, like expectation maximization (EM), is applied to estimate the latent variables (see the survey of [10]). Initially, [2] estimated both the true label and annotator miss-label rate. Later works by [13, 14] also model the annotation difficulty. However, the methods described above mainly aim to clean noisy labels, separately from the training stage of a machine learning algorithm. In contrast, the work of [11] jointly estimates the true label, annotator reliability and the parameters of a logistic regression classifier.

Recently, there have been several works that deal with training deep neural networks with noisy labels. Hinton *et al.* [9] suggested solving a noisy binary classification problem using a surrogate loss that models the miss label probability. [12] extended this idea to a multi-class problem by adding another noise layer to estimate the miss label probability for each class. Lastly, [5] suggested a similar solution; however, the authors used an EM scheme that also models dependency between the label noise and the input.

Most of the works for handling noisy labels assume that the annotation process is objective. In this case, the skill of the human annotator plays an important role because you can basically get the true label by a single reliable annotator. The methods described above work better than simple majority voting because they model the annotator skill. However, in subjective tasks like ours, the number of annotators for each example play a much important role since the target is the opinion of a large panel of annotators. Hence, the label noise also stems from sampling noise, and not just annotator skill noise.

3 MODELING SUBJECTIVE LABEL UNCERTAINTY

3.1 Uncertainty Model

Assume we have a large panel of N human annotators (*e.g.*, 100). The annotators are all qualified fashion experts, so we can assume that all of them have the same skill. Each annotator receives a pair of images (A_i, B_i) of the same person wearing different outfits. The annotator casts a vote, either "A" or "B" for the outfit he or she finds to be better for the user. In the ideal case, we would have N annotations for each pair, giving us the consensus $c_i = a_i/N$, where a_i denotes the number of votes for image A_i. However, due to budget restrictions we have only $n_i \ll N$ annotations, resulting in a noisy estimation of the consensus, $\hat{c_i} = \frac{\hat{a_i}}{n_i}$, where $\hat{a_i}$ is a noisy estimate for the number of votes for image A_i. Similarly, b_i and $\hat{b_i}$ are defined for image B_i.

In order to estimate the uncertainty in computing $\hat{c_i}$ due to sampling, we first assume that for each image pair i the annotators votes are independent given the true consensus c_i.

Each annotator's vote v_{ik}, is a binary random variable, where, without loss of generality, "1" indicates a vote for image A_i and "0" for image B_i. Sampling a single vote v_{ik} out of the n_i votes of a pair, is equivalent to sampling without replacement from a finite set of N Bernoulli random variables. This is known to follow a *Hypergeometric* distribution which is a generalization of the Binomial distribution. Therefore, for a pair i with n_i votes the probability of the noisy consensus c_i given by:

$$p(\hat{c_i}|c_i, n_i, N) \equiv p(\hat{a_i}|a_i, n_i, N) = \frac{\binom{a_i}{\hat{a_i}} \cdot \binom{N-a_i}{n_i-\hat{a_i}}}{\binom{N}{n_i}}. \quad (1)$$

Where the identity above holds, up to quantization effects of c_i. However, we would like to estimate the true consensus c_i, given the noisy consensus $\hat{c_i}$. To compute that we apply Bayes rule:

$$p(c_i|\hat{c_i}, n_i, N) = \frac{p(c_i|N) \cdot p(\hat{c_i}|c_i, n_i, N)}{p(\hat{c_i}|n_i, N)} \quad (2)$$

Since the likelihood function (1) follows a Hypergeometric distribution, and the prior $p(c_i|N)$ follows a *Beta-binomial* distribution, the posterior also follows a Beta-binomial distribution. This holds due to the conjugate prior property. Hence, we can write the prior as:

$$p(c_i|N) = BetaBin(a_i|\alpha, \beta, N) = \binom{N}{a_i} \frac{B(\alpha + a_i, \beta + N - a_i)}{B(\alpha, \beta)} \quad (3)$$

and the posterior probability as:

$$p(c_i|\hat{c_i}, n_i, N) \equiv p(a_i|\hat{a_i}, n_i, N) = BetaBin(a_i|\alpha + \hat{a_i}, \beta + n_i - \hat{a_i}, N) \quad (4)$$

The parameters α, β are estimated by taking a small sample (around 4K) of the train data annotated by all N annotators. Then, both α, β are estimated by maximum likelihood estimation. Fig.1 presents the empirical posterior distribution versus the analytical posterior given by the Beta-binomial distribution.

3.2 Class Uncertainty

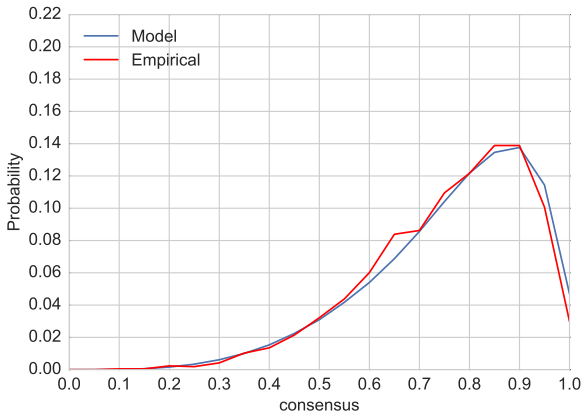
In the previous section, we have established the uncertainty in the estimation of the true consensus c_i. In a classification problem, we are interested in measuring the class uncertainty in the labeled data. The class definition is merely a quantization of the values of c_i into bins. For fashion related traits, the simplest quantization is into two bins. This basically means selecting the better outfit out of the two. In the binary case, $0 \leq c_i < 0.5$ implies that the panel finds image B to be more appealing, and the case $0.5 \leq c_i \leq 1$ implies that image A is more appealing. Recall that for each pair i we have n_i votes and a noisy estimation of the consensus $\hat{c_i}$. So, the uncertainty in the label of pair i reflects the probability that the true consensus c_i is actually in the other bin. In the binary case, it is the probability of having a label *flip*.

The uncertainty is determined by two factors: the number of sampled annotations n_i and the difficulty comparing the two images. If the value of $\hat{c_i}$ is either close to 1 or 0, this means that there is a clear preference for one image over the other. However, $\hat{c_i} \approx 0.5$ indicates that there is no clear preference for either A_i or B_i and determining which is better is much harder.

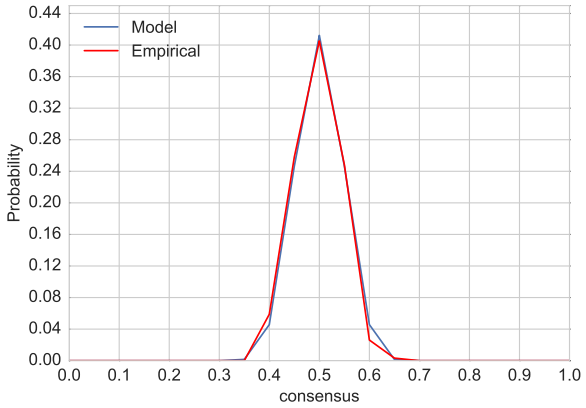
Formally, for the binary case, the uncertainty is computed by integrating over the consensus values of the opposite side of the consensus range

$$uncertainty(\hat{c_i} \geq 0.5, n_i|N) = \int_0^{0.5} p(c_i|\hat{c_i}, n_i, N) dc_i \quad (5)$$

$$uncertainty(\hat{c_i} < 0.5, n_i|N) = \int_{0.5}^{1} p(c_i|\hat{c_i}, n_i, N) dc_i \quad (6)$$



(a) $\hat{a_i} = 3, n_i = 3$



(b) $\hat{a_i} = 20, n_i = 40$

Figure 1: **model (blue) vs Empirical (red) posterior distribution for $n_i = 3, 40$, we can see that our model fits well the empirical distribution**

In the general case, the value of c_i in quantized into more than two bins. In this case, the uncertainty is calculated as the flip probability, over all the bins which are different from the bin of $\hat{c_i}$:

$$uncertainty(\hat{c_i}, n_i|N) = \sum_{c_i \neq \hat{c_i}} p(c_i|\hat{c_i}, n_i, N) \quad (7)$$

3.3 Dataset Uncertainty

In the majority of machine learning classification problems, the labels are considered "clean" and the upper bound for the algorithm accuracy is 100% (although it is not tight). In our case, the test set labels pose inherent noise, which is far from being uniform across labels. Therefore, it is critical to provide tight bounds for accuracy, because otherwise the interpretation of the algorithm failures can be misleading.

We denote u_i (7) as the uncertainty of the i^{th} label. The true value c_i may be quantized as a binary variable (as shown in previous section), or to more fine-grained values.
As a result, we can calculate the average uncertainty $\bar{u}$ for a given labeled dataset.

4 EXPERIMENTS

We consider the problem of deciding which outfit is better given a pair of outfits. We preform two types of experiments. The first, we demonstrate that the uncertainty model is able to predict when a learning algorithm becomes saturated and unable to generalize due to label noise. The second, we construct a custom neural network loss that directly models the posterior probability in (4). We demonstrate that this custom loss is superior to other standard loss functions.

4.1 Dataset

Our data set consists of image pairs showing individuals wearing different outfits. The images are captured not in a studio but in a natural environment (e.g. outdoors, at home etc.). Each pair depicts the same person wearing two different outfits. In order to estimate which outfit is more fashionable each pair was annotated by several fashion experts. The number of annotation per image pair varies between 1-100.

We divide the dataset into three non-overlapping sets: train (1M), validation (58K) and test (7K). The sets are divided such that a user can only appear in one set.

The test set D_{test} is constructed such that each pair $i \in D_{test}$ is annotated at least 60 times. This is because the expected accuracy of a given algorithm A is bounded by the test set uncertainty $\bar{u}_{test}$. Beyond that we cannot differentiate between the accuracy of various learning algorithms. If we require 60 annotations for each pair, the posterior distribution (4) is much narrower and the test set uncertainty is close to zero ($\bar{u}_{test} = 0.03$).

4.2 Implications of label uncertainty on learning

Following the uncertainty model in section 3, we study the implications of label uncertainty on the ability of a learning algorithm to generalize over the test set. In our experiments, we trained a neural network with binary classification targets where the labels are determined according to the majority vote of the annotation panel for that particular pair. In our experiments, we used a Siamese residual network [7] with weight sharing.

In order to test the relationship between the uncertainty of the training set and the ability of a learning algorithm to generalize from it. We selected several different sets from the entire training set. Each such set contains $180K$ with varying degrees of uncertainty and annotation budget. One each set, trained the same algorithm (Siamese Resnet50) and measured its binary accuracy on the common test set (7K densely annotated pairs). Table 1 summarizes the performance of the network trained over each sub-set. We can see that one can invest 4.3 annotations per image and still have a rather noisy train set over which the algorithm struggles to generalize. On the other hand, you can simply annotate each image once and gain roughly 9% increase in accuracy. We can observe similar behavior on the train set accuracy. It increases when the uncertainty decreases. It seems that the label noise makes the classification

problem less separable, making the fitting of the classifier more difficult.

It is interesting to see, that when the train set uncertainties are similar, the performance of the networks is similar as well. This despite the fact that one set has nearly three times more annotations per image than the other. Therefore, the uncertainty of the train set labels is a better predictor for the algorithm success than the number of annotation per sample.

# Pairs	Average annotations per pair	Train uncertainty	Train accuracy	Test accuracy
180K	4.3	0.40	0.700	0.584
180K	1.0	0.27	0.765	0.673
180K	3.1	0.25	0.758	0.669
180K	5.2	0.15	0.839	0.703

Table 1: Train sets with same pairs number and various uncertainty levels

We further tested the suggested model, by training the same network using two different training sets. The first contains only pairs with uncertainty smaller than 0.3. The second set is produced by augmenting the first set with 111K pairs with pairs with uncertainty ranging from 0.3 to 0.4 Table 2 summarizes the performance of the network over each training set. We can see that augmenting the set with 111K pairs with high uncertainty didn't contribute at all to the ability of the algorithm to improve the accuracy over the test set.

To complete the experiments, we consider two possible augmentations for a train set of 500K pairs with a single annotation per pair. The first, adds 500K pairs with a single annotation for each pair. This keeps the train set uncertainty unchanged (0.27 on both). The second, adds only 79K pairs but the train set uncertainty is reduced to 0.24. It is important to stress that both augmentations have the same annotation budget (500M each). Table 3 summarizes the accuracy of the network before and after the augmentations. We can see that adding 500K pairs without reducing the uncertainty in the train set did not improve the accuracy. In contrast, adding a small number of pairs with lower uncertainty increased the accuracy by 1.8%. We can learn that adding more noisy labels may not improve the generalization of the algorithm, though adding clean labels does as predicted by our model.

# Pairs	Average annotations per pair	Train uncertainty	Train accuracy	Test accuracy
355K	8.55	< 0.3	0.902	0.727
466K	8.53	< 0.4	0.842	0.724

Table 2: Network accuracy on augmented base set with augmentation of pairs with uncertainties $< 0.3, 0.4$

# Pairs	Average annotations per pair	Train uncertainty	Train accuracy	Test accuracy
500K	1.0	0.27	0.700	0.705
1M	1.0	0.27	0.667	0.700
579K	1.7	0.24	0.707	0.718

Table 3: Network accuracy with augmentation of the same uncertainty and lower uncertainty

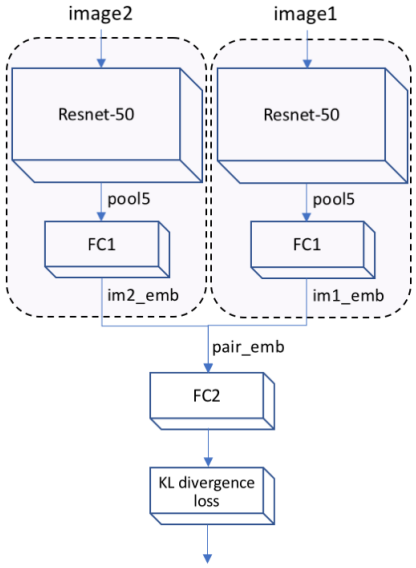


Figure 2: **Neural Network Architecture**

4.3 Learning with Uncertainty

In order to integrate our uncertainty model in to the learning process. We integrate it into a deep learning framework that instead of predicting the consensus class directly predicts the posterior distribution (4). The target of the network is to maximize the similarity between the posterior distribution and the output distribution from the neural network $\tilde{p}(c_i|A_i, B_i)$. This is achieved by training the network with a Kullback-Leibler divergence loss. Formally, the network loss for a single image pair is defined as:

$$loss = KL(p(c_i|\hat{c_i}, n_i, N)||\tilde{p}(c_i|A_i, B_i)) \quad . \tag{8}$$

The network architecture is a Siamese Resnet-50 [7] network with weight sharing between the branches. Each branch of the network is applied on a single image each image to obtain a per-image representation computed from the FC1 stacked on top of the pool5 layer (followed by Relu and Dropout). Both image representations are then concatenated to obtain a pair representation. This representation is connected to a fully connect layer with 10 output neurons followed by a Softmax normalization. The 10 neurons normalized output represents the estimated posterior distribution (4).

We trained our network using Stochastic Gradient Descent, with a batch size of 40. The network was initialized with pre-trained ImageNet model and trained for 200K iterations with a starting

Loss type	Train uncertainty	Test accuracy	RMSE
classification-10	0.20	0.703	0.358
classification-2	0.20	0.717	0.214
uncertainty	0.20	**0.727**	**0.206**
classification-10	0.15	0.705	0.251
classification-2	0.15	0.716	0.238
uncertainty	0.15	**0.735**	**0.202**

Table 4: Comparison of neural network performance

learning rate of 0.01. After each 60K iterations the learning rate was reduced by a factor of 10. We compared our approach (uncertainty) with two other methods, all using the KL-loss:

- 10 bins classification (classification-10) - we classify the bin of the sampled consensus using a one-hot ground truth vector (uniform sampling with 0.1 intervals).
- Binary classification (classification-2) - predicting the majority votes for each pair. The ground truth is either 0 or 1 and the number of output neurons is 2.

Table 4 summarizes the experiments on data sets with average uncertainty of 0.15 (531K pairs) and 0.2 (1M pairs) For each training set, we evaluate the performance of each one of the methods. In addition to binary accuracy, we calculated the root mean squared error (RMSE) for penalizing coarse errors more than fine ones.

From table 4 we can conclude that the uncertainty based loss achieves 2-3% increase in binary accuracy compared to other models. In addition, the granularity of the prediction is much more accurate as indicated by the 0.05-0.15 improvement in RMSE. The resulting improvements demonstrate the effectiveness of incorporating the posterior distribution of the label in the loss function.

5 SUMMARY

Predicting which outfit is more fashionable is a challenging task even for state-of-the-art deep learning architectures, trained on large amounts of labeled data. As we demonstrated, label noise plays an important role when training and evaluating an algorithm for this task. In this paper, we modeled the label noise as a statistical distribution conditioned on known factors. We have demonstrated that the ability of the algorithm to learn depends on the label uncertainty and not on the number of annotations per sample. In addition, we have shown that from a certain point adding more training examples without reducing the label uncertainty does not improve the ability of the algorithm to learn. Finally, we have shown how our model can be integrated into a learning algorithm by training a neural network that directly estimates the label uncertainty. This network was applied to the problem of selecting the better outfit out of a pair of outfits. For this task, integrating the model into the network yielded a performance gain compared to other loss functions.

REFERENCES

[1] Ritendra Datta, Jia Li, and James Z Wang. 2008. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 105–108.

[2] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (Jan. 2015), 98–136.

[5] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).

[6] Douglas Gray, Kai Yu, Wei Xu, and Yihong Gong. 2010. Predicting facial beauty without landmarks. In *European Conference on Computer Vision*. Springer, 434–447.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Amit Kagian, Gideon Dror, Tommer Leyvand, Daniel Cohen-Or, and Eytan Ruppin. 2007. A humanlike predictor of facial attractiveness. *Advances in Neural Information Processing Systems* 19 (2007), 649.

[9] Volodymyr Mnih and Geoffrey E Hinton. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. 567–574.

[10] Jafar Muhammadi, Hamid Reza Rabiee, and Abbas Hosseini. 2013. Crowd Labeling: a survey. *arXiv preprint arXiv:1301.2774* (2013).

[11] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual international conference on machine learning*. ACM, 889–896.

[12] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* (2014).

[13] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*. 2035–2043.

[14] Denny Zhou, Sumit Basu, Yi Mao, and John C Platt. 2012. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*. 2195–2203.

[15] Xingquan Zhu and Xindong Wu. 2004. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review* 22, 3 (2004), 177–210.