Label Error Correction and Generation Through Label Relationships

Zijun Cui,[1] **Yong Zhang,**[2] **Qiang Ji**[1]
[1]Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute
[1]{cuiz3, jiq}@rpi.edu
[2]Tencent AI Lab
[2]zhangyong201303@gmail.com

Abstract

For multi-label supervised learning, the quality of the label annotation is important. However, for many real world multi-label classification applications, label annotations often lack quality, in particular when label annotation requires special expertise, such as annotating fine-grained labels. The relationships among labels, on other hand, are usually stable and robust to errors. For this reason, we propose to capture and leverage label relationships at different levels to improve fine-grained label annotation quality and to generate labels. Two levels of labels, including object-level labels and property-level labels, are considered. The object-level labels characterize object category based on its overall appearance, while the property-level labels describe specific local object properties. A Bayesian network (BN) is learned to capture the relationships among the multiple labels at the two levels. A MAP inference is then performed to identify the most stable and consistent label relationships and they are then used to improve data annotations for the same dataset and to generate labels for a new dataset. Experimental evaluations on six benchmark databases for two different tasks (facial action unit and object attribute classification) demonstrate the effectiveness of the proposed method in improving data annotation and in generating effective new labels.

Introduction

The performance of supervised multi-label learning heavily relies on the quality and the quantity of label annotations. Most existing learning algorithms assume the correctness of the annotated labels which is not always true. Real-world datasets are annotated manually by human experts. Label error is defined as the discrepancy between the actual labels and the assigned labels. The presence of bias and inconsistency with human annotation has been demonstrated in (Beigman Klebanov and Beigman 2010; Passonneau and Carpenter 2014; Torralba and Efros 2011). Three factors contribute to incorrect annotations, including the imperfect evidence, confusion among similar patterns and perceptual errors, in particular for fine grain level annotations.

To illustrate how label errors are introduced in real-world datasets, we consider the annotation of facial action units (AUs) as an example. According to the Facial Action Coding System (FACS) (Friesen and Ekman 1978), AUs are defined as a contraction or relaxation of one or a group of facial muscles. Since the physical movements of muscles are continuous, there are boundary regions where the appearance of AUs is close to both inactive and active states. The final label depends on the personal interpretation, which varies with experts. An annotation error is illustrated in Figure 1. The consequences of label errors on the performance and on the complexity of classifiers have been widely studied, including theoretical analysis (Bootkrajang and Kabán 2012) and empirical assessment (Pantic et al. 2005). Besides labelling errors, another issue is inadequate annotation. Since manual annotation is often time-consuming and requires expertise, data is often labelled sparsely or not labelled at all.

Different methods have been proposed to deal with label errors. The natural approach is to correct the errors before learning. It first detects incorrect labels and then either removes (Segata et al. 2010) or flips them (Xiao et al. 2015). Another approach is to design robust models to handle label errors, either using an error tolerant loss function (Natarajan et al. 2013) or explicitly modeling label errors with probabilistic models (Ruiz et al. 2008). However, these methods are only applicable to a single binary classification problem.

In this work, we propose a method to exploit relationships among labels at different levels to reduce the annotation errors for multi-label learning problems. Specifically, we propose to use a Bayesian network to capture the structural relationships among labels at two different levels, i.e. object-level (meta-level) and property-level. The object-level labels capture the overall class such as object category, while the property-level labels represent specific local object properties or attributes. Our goal is to systemically capture the inherent dependencies between object-level and property-level labels and leverage such dependencies to correct the bias and inconsistencies in manual annotations. The underlying assumption is that despite the labeling errors, the dominant relationships among labels at different levels remain accurate and robust to labeling errors, and that the object-level labelling is easier to label and hence is less susceptible to la-

(a) (b) (c)

Figure 1: Instances of AU24 (Lip Pressor). (a) A positive template of AU24 defined in FACS. (b) A positive instance of AU24 in CK+ (Lucey et al. 2010) (the 5th sequence of the 136th subject). (c) A negative instance of AU24 in CK+ (the 10th sequence of the 127th subject). (c) is a label error.

belling errors. From the captured label relationships, a subset of most consistent and stable label relationships are identified and are used to improve label annotation at property-level and to generate labels.

The main contributions of the paper are as follows:

- We are the first to propose methods to systematically capture the structural relationships among labels at two different levels, to identify a subset of most robust and consistent label relationships, and to use them to improve the data annotation for existing datasets and to generate labels for new datasets.
- Introduce methods to evaluate the performance of the proposed model in label correction on real-world datasets without access to ground truth labels.
- Extensive empirical evaluations on six real-world benchmark datasets against state-of-the-art methods for two different computer vision tasks demonstrate the effectiveness of our method for both label annotation improvement and generation.

Related Work

The existing learning algorithms dealing with label errors can be categorized into two groups. Algorithms in the first group intend to detect samples with incorrect labels and then remove or flip them before classifier learning. (Khoshgoftaar and Rebours 2004) combined predictions of multiple classifiers to identify mis-labelled instances jointly. (Segata et al. 2010) proposed to use local maximal margin model trained on the k-Nearest Neighbors to determine whether to reject incorrect instances.

Algorithms in the second group try to learn a model to handle label errors. Many error tolerant methods have been proposed based on basic classifiers, including robust boosting (Bootkrajang and Kabán 2013), robust SVM (Xiao et al. 2015), NHERD algorithm (Crammer and Lee 2010), AROW (Crammer, Kulesza, and Dredze 2009), kernel Fisher discriminant (Lawrence and Schölkopf 2001) and soft confidence weighting (Hoi, Wang, and Zhao 2012). For example, (Bootkrajang and Kabán 2013) proposed the robust boosting by using an error tolerant basic learner and designing a new objective function. Label noise is also modeled as a random variable in Bayesian approaches (Ruiz et al. 2008) and the knowledge on the noise is added as prior.

Recently there are related works on applying deep models to multi-label problems to handle label errors. For example, (Sukhbaatar et al. 2014) was trying to match the noise distribution by introducing a constrained linear 'noise' layer to the NN model. (Veit et al. 2017) used millions of data with noisy annotations together with a small set of cleanly-annotated data to train a model that was able to handle annotation errors. These deep model based methods focus on developing robust features to handle noise, while our proposed approach focuses on improving label errors. So these two approaches are fundamentally different and complementary. In addition, the proposed model is feature independent that can work with hand-crafted as well as deep model learned features. Hence, we focus our comparison with methods that improve annotations instead of improving features.

Algorithms discussed before are considering label errors only for single label classification problem. Besides handling single noisy label for each sample, there are also works (Dawid et al. 1979; Karger, Oh, and Shah 2011; Parisi et al. 2014) that handle multiple noisy labels for each sample. These labels may be generated by crowdsourcing workers or by pre-trained classifiers. They are therefore noisy and contain errors. Different methods are introduced, including the EM method and belief propagation, to combine the multiple noisy labels to infer the true labels for each sample. These work require multiple noisy labels for each sample, while we focus on single label for each sample.

All existing methods failed to explicitly capture and exploit relationships among labels. These relationships provide useful information to improve label annotation. In this paper, we introduce a method to reduce label annotation errors by exploiting relationships among multiple labels.

Label Relationships Modeling and Inference

Problem Setup

Suppose that we are given a training set $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i, z_i\}_{i=1}^{l}$. Each $\mathbf{x}_i \in \mathbb{R}^d$ represents the features of the ith instance. $\mathbf{y}_i = \{y_{i,1}, ..., y_{i,K}\} \in \{+1, -1\}^K$ is the label vector consisting of K property-level labels while $z_i \in \{1, 2, ..., C\}$ is the corresponding object-level (meta-level) label that can be accurately acquired. C refers to the number of object-level labels. Since the property-level labels $\mathbf{y}_i$ are difficult to annotate as mentioned previously, the given training set $\mathcal{D}$ is polluted with label errors on $\mathbf{y}_i$ and z_i is assumed to be correct since it is relatively easy to annotate. The goal is to find a function (Eq. 1) that maps the noisy training set $\mathcal{D}$ to a better one $\mathcal{D}^*$ which has fewer errors for each property-level label under each object-level label.

$$f : \mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i, z_i\}_{i=1}^{l} \Rightarrow \mathcal{D}^* = \{\mathbf{x}_i, \mathbf{y}_i^*, z_i\}_{i=1}^{l} \quad (1)$$

To achieve such a purpose, we propose to leverage relationships among labels to improve the label annotation and to generate new labels. The relationships can be divided into two groups, including the relationships between the object-level label and the property-level labels, and the relationships among property-level labels. We learn a BN to capture these relationships encoded as probabilistic dependencies. We first learn the structure and parameters of the model, and we then infer a subset of most stable and consistent property-level label configuration, given only the

object-level label. The inferred property-level label configurations are then used to correct existing labels and to generate new labels. Details are described below.

Label Relationship Learning

In taxonomy, an instance can be assigned with multiple labels at different levels and many of them are closely related to each other, especially labels at different levels. For example, a facial expression image categorized by a certain expression can also be described by AUs. And AUs often depend on each other to form a coherent and meaningful facial expression. Some pairs of AUs co-occur frequently under global facial expressions while some never show up together. For example, AU6 (Cheek Raiser) and AU12 (Lip Corner Puller) often show up together to form a happy expression. AU5 (Upper Lid Raiser) and AU7 (Lid Tightener) never co-occur since anatomically, it is impossible to contract and relax the same muscle simultaneously. The relationships among the labels are hence useful to improve label annotation by considering all labels simultaneously.

To automatically capture relationships, we adopt a BN to learn dependencies among these labels. A BN is a direct acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes nodes and $\mathcal{E}$ for edges. The edges $\mathcal{E}$ between nodes represent the dependencies. The parameters of BN are used to represent the conditional probability distribution of each node given its parents. In a BN, Let $\mathbf{Y} = [Y_1, Y_2, ..., Y_K]$ and Z denote the property-level labels and the object-level label respectively. We want to learn a BN $\mathcal{G}$ to capture dependencies between $\mathbf{Y}$ and Z as well as relationships among Ys.

Structure learning We use the Bayesian Information Criterion (BIC) (Schwarz and others 1978) score function to search for an optimal model in the structure space $\mathcal{G}_s$:

$$Score(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D}|\hat{\theta}_{\mathcal{G}}, \mathcal{G}) - \frac{d(\hat{\theta}_{\mathcal{G}})}{2} \log N \qquad (2)$$

where the first term is the log-likelihood of data $\mathcal{D}$ under the structure $\mathcal{G}$ and the parameters $\hat{\theta}_{\mathcal{G}}$, representing the fitness of $\mathcal{G}$ to data $\mathcal{D}$. The second term is a penalty on the complexity of $\hat{\theta}_{\mathcal{G}}$, which is related to the number of free parameters $d(\hat{\theta}_{\mathcal{G}})$ and the number of training instances N. Here $\mathcal{D}$ contains two levels of labels. $\hat{\theta}_{\mathcal{G}}$ is obtained through a Bayesian parameter learning given one candidate structure $\mathcal{G}$. The Branch and Bound algorithm (De Campos and Ji 2011) is adopted to exactly learn the structure of the BN maximizing the BIC score (Eq. 2). The method returns a globally optimal solution $\mathcal{G}^*$.

Bayesian parameter learning Since $\mathbf{Y}$ and Z are discrete, we use Conditional Probability Table (CPT) to represent the conditional distribution of each node given its parents. To deal with insufficient data for certain parameters, we propose to employ a Bayesian method to learn the BN parameters instead of using the conventional maximum likelihood method,i.e.,

$$\theta^* = \mathbb{E}_{P(\theta|\mathcal{G},\mathcal{D},\alpha)}[\theta] = \int \theta P(\theta|\mathcal{G}, \mathcal{D}, \alpha) d\theta \qquad (3)$$

where α are the parameters for the Dirichlet distribution that specify the prior distribution of the parameters θ.

We denote one node as X_i and its parents as $Pa(X_i)$. The parameter θ_{ij} represents the conditional distribution of X_i where j represents the state of $Pa(X_i)$. The posterior distribution of θ_{ij} is $P(\theta_{ij}|\mathcal{G}, \mathcal{D}, \alpha) = Dir(\alpha_{ij1} + N_{ij1}, ..., \alpha_{ijr_i} + N_{ijr_i})$, where r_i is the number of states of X_i. α_{ijr_i} is the hyper-parameter, which choose to be 1 for uniform distribution. Since θ_{ij} is the parameters of a multinomial distribution and the posterior of θ_{ij} is Dirichlet distribution that is conjugate to multinomial distribution, we can get the analytical solution for Eq. 3,

$$\theta^*_{ijk} = \mathbb{E}_{P(\theta_{ijk}|\mathcal{G},\mathcal{D},\alpha)}[\theta_{ijk}] = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \qquad (4)$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.

Label Relationship Inference

Given the learned structure and parameters, our goal is to leverage the relationships encoded in the model to improve label annotation. For this, we consider the largest subset of property-level label $\mathbf{Y}$, whose relationships are most consistent and stable for a given object-level label Z. Such relationships can subsequently be used to improve label annotation and to generate new labels. To obtain such relationships, we propose a constrained MAP inference as,

$$\mathbf{Y'}^*_Z = \arg \max_{\mathbf{Y'}_{max} \subseteq \mathbf{Y}} P(\mathbf{Y'}_{max}|Z, \mathcal{G}^*, \theta^*) \geq \eta \qquad (5)$$

where $\mathbf{Y'}_{max}$ is the maximum subset of $\mathbf{Y}$,
$P(\mathbf{Y'}_{max}|Z, \mathcal{G}^*, \theta^*)$ is the probability of the property-level labels $\mathbf{Y'}$, given the object-level label Z, the BN structure G^* and parameters θ^*. η is a pre-defined confidence level.

To efficiently explore the subspace of $\mathbf{Y}$ to identify the largest subset, we start from the whole set of property-level labels $\mathbf{Y}$, then at each time reduce the size of the property-level label set by one until we obtain a subset $\mathbf{Y'}$ whose most possible configuration produced by Eq. 5 satisfy

$$P(\mathbf{Y'}^*|Z, \mathcal{G}^*, \theta^*) \geq \eta \qquad (6)$$

The constrained MAP inference can be performed for each value of the property-level label Z, yielding $\mathbf{Y'}^*_z$, i.e., the optimal property-level label relationships for each object level label value z.

Label Correction and Generation

We assume $\mathbf{Y'}^*_Z$ are robust to labelling errors and are true across datasets. They therefore represent the most stable and consistent property-level label relationships and can hence be used to correct label annotations and to generate new labels. For a dataset with existing annotations, label corrections can then be performed by examining the labels for each sample and correcting them if they are inconsistent with $\mathbf{Y'}^*_Z$. For a dataset with missing property-level annotations but with object-level annotations, we can then use $\mathbf{Y'}^*_z$ to produce property-level labels for each sample, given its object-level label value z.

Experiments

We demonstrated the performance of our method on six real-world benchmark databases for two computer vision tasks: facial action unit(AU) recognition and object attribute prediction. For AU recognition, facial expression corresponds to the object-level label and AUs correspond the property-level labels. Similarly, for object attribute prediction, object category corresponds to the object-level label and attributes correspond to property-level labels.

Datasets: For facial expressions, the Extended Cohn-Kanande (CK+) (Lucey et al. 2010) database, the M&M Initiative facial expression database(Pantic et al. 2005)(MMI), BP4D-Spontaneous database(BP4D)(Zhang et al. 2013) and EmotionNet dataset(Matthews and Baker 2004) are four widely used databases for AU recognition. CK+ and MMI are posed expression databases while BP4D is spontaneous. EmotionNet dataset is collected in the wild with annotations automatically generated by existing algorithms, and contains significant annotation errors. 309 sequences for 109 subjects with both annotated AUs and the major six expressions were collected from the CK+ database, while 196 sequences for 27 subjects were collected from the MMI database. The six expressions consist of anger, disgust, fear, happiness, sadness and surprise. Only the apex frame in each sequence was used for training or testing. In our experiments, we considered the recognition of the 10 most frequent AUs 1,2,6,7,9,12,17,23,24 and 25 in both CK+ and MMI. For BP4D, we extracted 732 apex frames of 41 subjects under 5 expressions from the raw sequences by following the construction procedures of CK+. The expressions include anger, disgust, fear, happiness and surprise except for sadness since only very few frames of sadness satisfy the rules in CK+. And we considered the recognition of 11 AUs such as AU 1,2,6,7,10,12,14,15,17,23 and 24. For EmotionNet dataset, all 24,556 images with AU annotations were collected. Only around 2,000 images have expression annotations. We considered the recognition of 11 AUs. They are AU1,2,4,5,6,9,12,17,20,25 and 26. To evaluate the performance, we performed 5 fold subject independent cross validation with F1-score as measurement. Each experiment was run 10 times, and the average F1-score was reported.

For object attribute prediction, the a-Pascal database and the a-Yahoo database (Farhadi et al. 2009) are used for attribute prediction. The a-Pascal database (Farhadi et al. 2009) contains 6340 instances for training and 6355 for testing. It has totally 20 object categories. The a-Yahoo database contains 2644 instances belonging to 12 object categories that have no intersection with the a-Pascal database. Each instance is annotated for 64 attributes in both databases. To evaluate the performance on attribute, we used G-Mean as the measurement (Wang and Ji 2013).

Features: For AU recognition, face images were firstly normalized to 200×200 according to two eye centers. 51 inner facial landmarks were extracted by (Matthews and Baker 2004), around which Local Binary Patterns (LBP) (Ojala, Pietikäinen, and Mäenpää 2002) features were computed from regions of 32×32 pixels. Finally, PCA was adopted to reduce the feature dimensionality to 150. For attribute prediction, 9751 dimension features are used for each image provided by both a-Pascal and a-Yahoo databases.

hyper-parameter η: For all experiments, the confidence level η is determined through a validation dataset.

(a) (b)

Figure 2: Two examples for successful correction from CK+. For (a), AU9(Nose Wrinkler) is corrected from 'OFF' to 'ON' for expression Disgust. For (b), AU23(Lips Tightener) is corrected from 'ON' to 'OFF' for expression Surprise.

Label Correction Evaluation with GT Annotations

To compare the annotation error rate of labels before and after correction, we manually selected a subset of annotations from CK+ and verified their correctness based on the Facial Action Coding System(FACS) (Friesen and Ekman 1978). According to FACS, because of underlying facial anatomy and the need to produce meaningful facial expressions, certain AUs are always present for certain facial expressions. For example, AU6(Cheek Raiser) and AU12(Lip Corner Puller) are always present for happy expression. We follow these FACS rules to obtain ground truth labels for certain AUs under certain facial expressions. The verified subset of annotations serves as our ground truth and only the error rate for annotations that have ground truth labels are reported. As shown in Table 1, annotation errors for different AUs under different expressions are corrected. For AU1, annotation errors of 12%, 7.1% and 2.4% are corrected respectively for expression Fear, Sadness and Surprise. Similar label improvements are obtained for AU9, AU12, AU23, and AU24. In Figure 2, we show two visual correction examples corresponding to the results in Table 1.

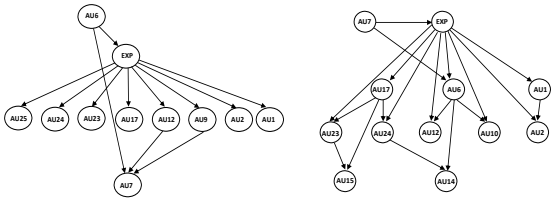
Label Correction Evaluation with Property-level Label Classification

As the provided labels for the testing samples are also noisy, it does not make sense to compare the corrected labels with the provided labels for the testing samples. To evaluate the proposed method on noisy real datasets, we train property-level label classifiers on the training dataset with original labels and with corrected labels respectively. We then compare their classification performance on the same testing dataset.

We first trained two baseline classifiers by using the original noisy labels (NLB) and the improved labels (MAPLB) respectively, and compared their performances on the same testing datasets. Then, we evaluated the performance of several state-of-the-art algorithms by training them using corrected labels and original labels respectively. Thirdly, we compared our method with a set of state-of-the-art algorithms that can handle label errors.

Error rate	AU1	AU9	AU12	AU23	AU24
Angry	– –	– –	0.022 **/0.000**	0.000 /0.000	0.267 **/0.000**
Disgust	– –	0.017 **/0.000**	– –	– –	– –
Fear	0.120 **/0.000**	– –	– –	– –	– –
Happy	– –	– –	0.029 **/0.000**	– –	– –
Sadness	0.071 **/0.000**	– –	– –	– –	– –
Surprise	0.024 **/0.000**	– –	– –	0.012 **/0.000**	– –

Table 1: Comparison of the annotation error rate of original labels and corrected labels on CK+



(a) (b)

Figure 3: AU recognition experiment on CK+ and BP4D. (a) Structure of the learned BN on CK+; (b) Structure of the learned BN on BP4D

For the methods we compare with, we apply the same hand-crafted features in our experiments(except for the experiments with deep models) and thus we can demonstrate that any performance improvement is because of label correction. For the performance on the benchmark datasets, we use simple baseline classifiers as the goal of this work is not to produce the best model for a particular task but to improve existing methods through label corrections.

Basic classifiers with improved labels We firstly conducted experiments on CK+ and BP4D. We used a regularized logistic regression model(LR) and SVM as two basic classifiers. Each classifier is trained with the improved labels and the original labels respectively. Then, the trained classifiers are used for AU prediction on the same testing set. Classifiers are trained to classify each AU independently. The results are illustrated in Figure 3, Table 2 and Table 3.

As shown in Table 2, on CK+, MAPLB outperforms NLB by 4.5% with LR, and by 5.5% with SVM on average of all AUs. The F1-score of AU7, AU24 have been improved significantly, especially for AU7 (Lid Tightener) which is hard to annotate only according to the local appearance. The relationships can improve noisy labels, especially for the labels that are difficult to annotate. For BP4D as shown in Table 3, the performance of MAPLB is 3.0% better than NLB with a LR classifier, and 3.1% better with a SVM classifier. The improvement is also significant though not as much as CK+ since the BP4D is a spontaneous database. BP4D has larger facial appearance variance, and the relationships among expression and AUs are more complex. We also compare our method to (Wang, Gan, and Ji 2017) which uses BN to jointly predict the AU labels and achieves 81.51% on CK+ and 66.68% on BP4D. Our method outperforms (Wang, Gan, and Ji 2017) by 1.49% on CK+ and by 2.02% on BP4D with a LR classifier.

Method		AU1	AU2	AU6	AU7	AU9	AU12
LR	NLB	0.936	**0.912**	0.787	0.480	0.908	0.897
	MAPLB	0.936	0.911	**0.804**	**0.701**	**0.923**	**0.913**
SVM	NLB	**0.935**	0.890	0.787	0.450	0.899	0.891
	MAPLB	0.932	**0.899**	**0.803**	**0.674**	0.899	**0.910**
Method		AU17	AU23	AU24	AU25	MEAN	
LR	NLB	**0.873**	0.585	0.525	0.947	0.785	
	MAPLB	0.866	**0.613**	**0.681**	**0.948**	**0.830**	
SVM	NLB	0.877	0.611	0.386	**0.950**	0.767	
	MAPLB	**0.881**	**0.629**	**0.649**	0.946	**0.822**	

Table 2: Comparison of the improved and the original labels for AU recognition performance on CK+

Method		AU1	AU2	AU6	AU7	AU10	AU12
LR	NLB	0.546	0.394	0.764	0.826	0.833	0.858
	MAPLB	**0.605**	**0.488**	**0.787**	**0.870**	**0.855**	0.858
SVM	NLB	0.550	0.418	0.771	0.827	0.830	0.864
	MAPLB	**0.598**	**0.500**	**0.796**	**0.870**	**0.853**	0.864
Method		AU14	AU15	AU17	AU23	AU24	MEAN
LR	NLB	0.663	0.532	0.726	0.549	0.534	0.657
	MAPLB	**0.737**	**0.537**	0.726	**0.562**	0.534	**0.687**
SVM	NLB	0.639	0.519	0.731	0.532	0.525	0.655
	MAPLB	**0.732**	0.519	0.731	**0.560**	0.525	**0.686**

Table 3: Comparison of the improved and the original labels for AU recognition performance on BP4D

To address the concern that label errors in CK+ and BP4D can be insignificant, we conducted the experiment on EmotionNet dataset. Because provided expression labels are limited, we captured relationships among 11 AUs without expressions. Due to occlusion, many images have partial AU labels. The learned AU relationships are firstly applied to generate the most probable property-level labels for the missing AU annotations. Then AU classifiers are trained with the completed AU labels and evaluated on the testing set. SVM and a standard 3-layer CNN are applied as the baseline classifiers and the results are reported in Table 4. As shown in Table 4, MAPLB outperforms NLB by 3.3% on average of 11 AUs with SVM. With CNN, the F1-score of AU25 is increased by 1.5% and the average performance is improved by 0.6%, which show that the proposed method can work with not only hand-craft features but also deep features. The improvement with CNN is not as significant as with SVM because the CNN can better handle label errors. These results demonstrate that our model still works well for datasets that contain significant label errors and without object-level labels for both shallow and deep models.

For attribute prediction, we used linear SVM (Chang and Lin 2011) as the basic classifier. We trained the classifier on a-Yahoo dataset and tested its performance on a-Pascal dataset. As shown in the Table 5, the performance is improved by 1.0%. The performance improvement for attribute recognition is not as significant as AU recognition, which may be due to the fact that AUs are harder to annotate than most attributes, and hence are more prone to labelling errors.

Method		AU1	AU2	AU4	AU5	AU6	AU9
LR	NLB	0.473	0.425	0.523	0.427	0.586	0.430
	MAPLB	**0.485**	**0.493**	0.523	**0.490**	**0.586**	**0.495**
CNN	NLB	**0.486**	0.525	**0.646**	0.486	0.785	0.644
	MAPLB	0.482	**0.537**	0.641	**0.500**	**0.797**	**0.648**
Method		AU12	AU17	AU20	AU25	AU26	MEAN
LR	NLB	0.644	0.417	0.412	0.583	**0.476**	0.491
	MAPLB	0.644	**0.495**	**0.499**	0.583	0.475	**0.524**
CNN	NLB	**0.855**	0.545	0.499	0.793	0.561	0.620
	MAPLB	0.852	**0.546**	**0.499**	**0.808**	**0.572**	**0.626**

Table 4: Comparison of the improved and the original labels for AU recognition on EmotionNet

Method	NLB	MAPLB
a-Pascal	0.743	**0.753**

Table 5: Comparison of the improved labels and original labels for attribute prediction on a-Pascal

State-of-the-art multi-label learning models with improved labels We train state-of-the-art multi-label learning models by using the original labels and the corrected labels respectively. We then compare the performance of the models trained using the improved labels to the performance of the models trained using the original labels.

Totally we consider four multi-label learning algorithms: ML-KNN(Zhang and Zhou 2007), LEAD(Zhang and Zhang 2010), LIFT(Zhang and Wu 2015), and MLTSVM(Chen et al. 2016). These algorithms exploit the relationships among labels to improve classifiers. The results are shown in Tabel 6. Performances of these four methods get improved by using the corrected labels during training. MLTSVM achieves 9.3% improvement on CK+ and 8.8% improvement on BP4D which are significant. For attribute prediction, we trained SVM with the improved labels to provide attribute measurements in (Wang and Ji 2013) instead of using the original noisy labels. We use the measurements to learn the BN, and apply it for prediction. The performance of using the improved labels is 79.4%, compared to the original 79.0% in (Wang and Ji 2013).

Comparison with state-of-the-art methods We compare basic classifiers that are trained with corrected labels with eight SoA methods that are proposed to handle label errors, including ALFASVM(Xiao et al. 2015), LSVM(Segata et al. 2010), rLR(Bootkrajang and Kabán 2012), rBoost(Bootkrajang and Kabán 2013), NKFD (Lawrence and Schölkopf 2001), NHERD(Crammer and Lee 2010), SCW(Hoi, Wang, and Zhao 2012), and AROW (Crammer, Kulesza, and Dredze 2009). We conduct experiments on the CK+, MMI and BP4D. We apply regularized Logistic Regression (BN-LR) and linear SVM (BN-SVM) as two basic classifiers. For several AUs, MMI contains very few positive instances. LSVM and rBoost can not handle such cases. Since AU annotations in the MMI is very sparse and the instances are not enough to capture the relationships precisely, we apply the BN learned on CK+ to infer the labels for MMI.

As shown in Table 7, BN-LR and BN-SVM outperform other methods on CK+. NKFD is a kernel fisher discriminant method, whose performance is the same with BN-SVM on CK+. However, NKFD uses RBF kernel while SVM is a linear method. On MMI and BP4D, BN-SVM achieves the highest F1-score. These results further demonstrate the effectiveness of using the relationships to handle label errors.

Label Correction Evaluation without GT Annotations

The previous experiments assume testing labels are good which may not be the case since the annotations of the testing data are subject to the same errors as the training data. We conducted two experiments to evaluate the effectiveness of the proposed method without access to the GT labels.

Evaluation with prediction uncertainty We evaluate the performance of the classifier by calculating the entropy of its predictions. The assumption is that a better classifier should produce less average uncertainty on the unlabelled testing samples. We used regularized logestic regression model (LR) as the base classifier and computed the entropy with the output probabilistic distribution as

$$H(y) = -\sum_{i=1}^{N} P(y_i)\log_2(P(y_i)) \tag{7}$$

where $N = 2$ as each AU is of binary states. Prediction entropy is calculated for each instance. We then take average entropy over all testing instances which is treated as the classifier uncertainty. A classifier that produces predictions with lower entropy is less uncertain about its predictions, and a good classifier should provide accurate predictions with low uncertainty. We compared the prediction uncertainty of LR classifiers that are trained with the improved labels(MAPLB) and the original labels(NLB) respectively.

As shown in Table 8, on both CK+ and BP4D, MAPLB produces predictions with significantly less uncertainty. On AU7, MAPLB achieves 63.4% less average entropy on CK+ and 64.1% less average entropy on BP4D compared to NLB. The reduction in prediction uncertainty demonstrates again the improvement of original noisy labels.

Evaluation through surrogate task We evaluate the effectiveness of our method on a surrogate (meta-level) task. For AU recognition, instead of directly evaluating the AU recognition on the testing set, we indirectly evaluate the label correction performance by studying its impact on facial expression recognition. The expression labels are assumed to be reliable, and thus improvement on expression classification accuracy can validate the correction of AU annotations. During training, we train an expression classifier to map AU labels to expression and also train AU classifiers to map image features to AU labels. This is done for both original and corrected AU labels. During testing, we predict the AU labels for the unseen image and use the predicted AUs as input of the expression classifier for expression recognition. We used regularized logestic regression model(LR) and SVM as two basic classifiers and performed evaluation on the original labels and the improved labels respectively. The results are shown in Table 9. On CK+, MAPLB outperforms NLB by 6.5% with LR classifier and by 6.1% with SVM

Method		ML-KNN (Zhang and Zhou 2007)	LEAD (Zhang and Zhang 2010)	LIFT (Zhang and Wu 2015)	MLTSVM (Chen et al. 2016)
CK+	NLB	0.700	0.755	0.711	0.691
	MAPLB	**0.752**	**0.817**	**0.776**	**0.784**
BP4D	NLB	0.611	0.626	0.638	0.609
	MAPLB	**0.659**	**0.695**	**0.676**	**0.697**

Table 6: Improvements of state-of-the-art methods by using the improved labels on CK+ and BP4D

Method	ALFASVM (Xiao et al. 2015)	LSVM (Segata et al. 2010)	rLR (Bootkrajang and Kabán 2012)	rBoost (Bootkrajang and Kabán 2013)	NKFD (Lawrence and Schölkopf 2001)
CK+	0.783	0.812	0.803	0.803	0.820
MMI	0.416	-	0.482	-	0.494
BP4D	0.653	0.651	0.669	0.661	0.668
Method	NHERD (Crammer and Lee 2010)	AROW (Crammer, Kulesza, and Dredze 2009)	SCW (Hoi, Wang, and Zhao 2012)	BN-LR	BN-SVM
CK+	0.779	0.791	0.777	**0.826**	0.820
MMI	0.412	0.396	0.368	0.514*	**0.532***
BP4D	0.657	0.663	0.658	0.684	**0.688**

Table 7: Comparison with label error tolerant state-of-art methods on CK+, MMI, and BP4D

Dataset		AU1	AU2	AU6	AU7	AU9
CK+	NLB	0.181	0.181	0.313	0.317	0.085
	MAPLB	**0.136**	**0.147**	**0.074**	**0.116**	**0.081**
BP4D	NLB	0.300	0.298	0.303	0.284	-
	MAPLB	**0.235**	**0.212**	**0.132**	**0.102**	-
Dataset		AU10	AU12	AU14	AU15	AU17
CK+	NLB	-	0.130	-	-	0.257
	MAPLB	-	**0.074**	-	-	**0.131**
BP4D	NLB	0.299	0.221	0.446	0.279	0.329
	MAPLB	**0.132**	**0.132**	**0.119**	**0.213**	**0.246**
Dataset		AU23	AU24	AU25	MEAN	
CK+	NLB	0.126	0.151	0.196	0.194	
	MAPLB	**0.109**	**0.109**	**0.124**	**0.110**	
BP4D	NLB	0.293	0.232	-	0.299	
	MAPLB	**0.246**	**0.205**	-	**0.179**	

Table 8: Comparison of the improved and the original labels for AU recognition uncertainty on CK+ and BP4D

classifier. On BP4D dataset, SVM achieves 3.9% improvement on expression classification accuracy and LR achieves 3.2% improvement. Evaluations on the surrogate task further show that the original labels are improved.

Method		LR	SVM
CK+	NLB	0.820	0.825
	MAPLB	**0.885**	**0.886**
BP4D	NLB	0.425	0.426
	MAPLB	**0.457**	**0.465**

Table 9: Evaluation through expression recognition

Label Generation Evaluation

In the previous sections, we evaluate our method in terms of its capability in label correction. In this section, we evaluate its ability in generating new labels for another new dataset. For this, we perform the cross-database annotation generation experiment. The relationships among object-level label and property-level labels exist in different databases for the same task. We use the label relationships learned on the source database to generate labels for the target database. On the target database, given the object-level label, the label relationships $\mathbf{Y'}_Z^*$ learnt in the source domain are used to generate the most probable property-level labels. Specifically, we learn the BN structure and parameters on the CK+ database and use the learned BN to generate AU labels for training samples in the MMI database given the expression. AU classifiers are trained with original AU labels(NLB) and the generated AU labels(MAPLB) respectively. The trained classifiers are then evaluated on the same testing set. The average F1-score over AUs is reported in Table 10. The performance of using the generated AU labels achieves 5.0% improvement over using the original labels with SVM classifier and 4.9% improvement with LR classifier.

Method		LR	SVM
MMI	NLB	0.465	0.482
	MAPLB	**0.514**	**0.532**

Table 10: Cross-database annotation generation

Discussion

Contribution of object-level labels To evaluate the contribution of object-level labels to label correction, we conduct an ablation study on CK+ with setting a regularized logistic regression model(LR) as the basic classifier. We train the basic classifier with the original noisy labels(NLB), improved labels(MAPLB) by using relationships among AUs and expressions, and improved labels(mMAPLB) by using relationships among AUs only. The experiments settings remain the same. As shown in Table 11 below, object-level labels are important for effective label correction.

Conclusion

Label errors are always present. For effective supervised learning, label errors should be dealt with to mitigate their side effects. In this paper, we proposed a novel method to capture and leverage relationships among labels at two different levels to improve both label annotation and to generate new labels. We extensively evaluated our method on

Method	AU1	AU2	AU6	AU7	AU9	AU12
NLB	0.936	**0.912**	0.787	0.480	0.908	0.897
MAPLB	0.936	0.911	**0.804**	**0.701**	**0.923**	**0.913**
mMAPLB	0.936	**0.912**	0.787	0.608	0.908	0.897
Methods	AU17	AU23	AU24	AU25	MEAN	
NLB	**0.873**	0.585	0.525	0.947	0.785	
MAPLB	0.866	**0.613**	**0.681**	**0.948**	**0.830**	
mMAPLB	**0.873**	0.585	0.624	0.947	0.809	

Table 11: The contribution of object-level labels

benchmark datasets with several state-of-the-art methods for the two computer vision tasks, including AU recognition and attribute prediction. The experimental results show the effectiveness of the proposed method in both improving label annotation and in generating new labels. The proposed method can also generalize to other object recognition tasks where there exist strong relationships among object labels.

Acknowledgment

The work described in this paper is supported in part by a DARPA grant FA, and in part by the US National Science Foundation award CNS $\#1629856$.

References

Beigman Klebanov, B., and Beigman, E. 2010. Some empirical evidence for annotation noise in a benchmarked dataset. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 438–446. Los Angeles, California: Association for Computational Linguistics.

Bootkrajang, J., and Kabán, A. 2012. Label-noise robust logistic regression and its applications. In *Proceedings of the 2012th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'12. Berlin, Heidelberg: Springer-Verlag. 143–158.

Bootkrajang, J., and Kabán, A. 2013. Boosting in the presence of label noise. In *UAI*.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *TIST*.

Chen, W.-J.; Shao, Y.-H.; Li, C.-N.; and Deng, N.-Y. 2016. Mltsvm: A novel twin support vector machine to multi-label learning. *PR*.

Crammer, K., and Lee, D. D. 2010. Learning via gaussian herding. In *NIPS*.

Crammer, K.; Kulesza, A.; and Dredze, M. 2009. Adaptive regularization of weight vectors. In *NIPS*.

Dawid, P.; Skene, A. M.; Dawidt, A. P.; and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics* 20–28.

De Campos, C. P., and Ji, Q. 2011. Efficient structure learning of bayesian networks using constraints. *JMLR*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *CVPR*.

Friesen, E., and Ekman, P. 1978. Facial action coding system: A technique for the measurement of facial movement. *Palo Alto*.

Hoi, S. C.; Wang, J.; and Zhao, P. 2012. Exact soft confidence-weighted learning. In *ICML*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, 1953–1961.

Khoshgoftaar, T. M., and Rebours, P. 2004. Generating multiple noise elimination filters with the ensemble-partitioning filter. In *IRI*.

Lawrence, N. D., and Schölkopf, B. 2001. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*.

Lucey, P.; Cohn, J. F.; Kanade, T.; Saragih, J.; Ambadar, Z.; and Matthews, I. 2010. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *CVPRW*.

Matthews, I., and Baker, S. 2004. Active appearance models revisited. *IJCV*.

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. In *NIPS*.

Ojala, T.; Pietikäinen, M.; and Mäenpää, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*.

Pantic, M.; Valstar, M.; Rademaker, R.; and Maat, L. 2005. Web-based database for facial expression analysis. In *ICME*.

Parisi, F.; Strino, F.; Nadler, B.; and Kluger, Y. 2014. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences* 111(4):1253–1258.

Passonneau, R. J., and Carpenter, B. 2014. The benefits of a model of annotation. *TACL*.

Ruiz, M.; Girón, F.; Pérez, C.; Martín, J.; and Rojano, C. 2008. A bayesian model for multinomial sampling with misclassified data. *Journal of Applied Statistics*.

Schwarz, G., et al. 1978. Estimating the dimension of a model. *The annals of statistics* 6(2):461–464.

Segata, N.; Blanzieri, E.; Delany, S. J.; and Cunningham, P. 2010. Noise reduction for instance-based learning with a local maximal margin approach. *Journal of Intelligent Information Systems*.

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR*.

Veit, A.; Alldrin, N.; Chechik, G.; Krasin, I.; Gupta, A.; and Belongie, S. J. 2017. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 6575–6583.

Wang, X., and Ji, Q. 2013. A unified probabilistic approach modeling relationships between attributes and objects. In *ICCV*.

Wang, S.; Gan, Q.; and Ji, Q. 2017. Expression-assisted facial action unit recognition under incomplete au annotation. *pr*.

Xiao, H.; Biggio, B.; Nelson, B.; Xiao, H.; Eckert, C.; and Roli, F. 2015. Support vector machines under adversarial label contamination. *Neurocomputing*.

Zhang, M.-L., and Wu, L. 2015. Lift: Multi-label learning with label-specific features. *TPAMI*.

Zhang, M.-L., and Zhang, K. 2010. Multi-label learning by exploiting label dependency. In *SIGKDD*.

Zhang, M.-L., and Zhou, Z.-H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *PR*.

Zhang, X.; Yin, L.; Cohn, J. F.; Canavan, S.; Reale, M.; Horowitz, A.; and Liu, P. 2013. A high-resolution spontaneous 3d dynamic facial expression database. In *FG workshop*.