NoiseRank: Unsupervised Label Noise Reduction with Dependence Models

Karishma Sharma[1], Pinar Donmez[2], Enming Luo[2], Yan Liu[1], and I. Zeki Yalniz[2]

[1] University of Southern California, USA
[2] Facebook AI
krsharma@usc.edu,pinared@fb.com,eluo@fb.com,yanliu.cs@usc.edu,izy@fb.com

Abstract. Label noise is increasingly prevalent in datasets acquired from noisy channels. Existing approaches that detect and remove label noise generally rely on some form of supervision, which is not scalable and error-prone. In this paper, we propose NoiseRank, for unsupervised label noise reduction using Markov Random Fields (MRF). We construct a dependence model to estimate the posterior probability of an instance being incorrectly labeled given the dataset, and rank instances based on their estimated probabilities. Our method i) does not require supervision from ground-truth labels or priors on label or noise distribution, ii) is interpretable by design, enabling transparency in label noise removal, iii) is agnostic to classifier architecture/optimization framework and content modality. These advantages enable wide applicability in real noise settings, unlike prior works constrained by one or more conditions. NoiseRank improves state-of-the-art classification on Food101-N (∼20% noise), and is effective on high noise Clothing-1M (∼40% noise).

Keywords: Label noise, Unsupervised learning, Classification

1 Introduction

Machine learning has become an indispensable component of most applications across numerous domains, ranging from vision, language and speech to graphs and other relational data [22]. It has also led to an increase in the amount of training data required to effectively solve target problems. Labeled datasets, typically obtained through manual efforts, are prone to labeling errors arising from annotator biases, incompetence, lack of attention, or ill-formed and insufficient labeling guidelines. The likelihood of human errors increases in domains with high ambiguity [24, 30]. Additionally, there is an increasing dependence on automated data collection such as employing web-scraping, crowd-sourcing and machine-generated labeling [2, 29]. However, the cheap but noisy channels have made it imperative to deal with incorrectly labeled samples.

Existing literature either focus on training noise-robust classifiers [1, 7, 10, 11, 16], or attempt to reduce or correct label noise in the dataset generally with some form of supervision [15, 20, 28, 34]. However, attention is shifting towards

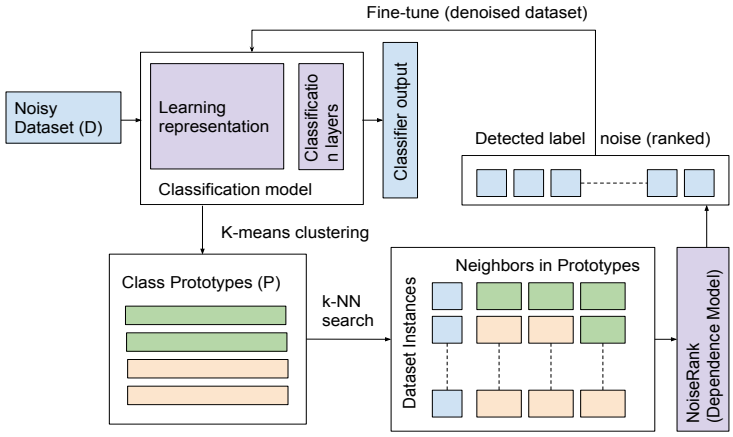


Fig. 1: Illustration of "NoiseRank" framework for unsupervised label noise reduction and iterative model training (interpretable, and agnostic to classification model architecture and optimization framework)

unsupervised label noise reduction due to obvious practical benefits. Most earlier methods use some form of supervision, either from verified or clean labels, or priors on the label/noise distribution, in order to guide the detection of mislabeled examples. In this work, we propose a fully unsupervised approach for label noise detection using Markov Random Fields (MRF), also known as dependence models, which provide a generic framework for modeling the joint distribution of a large set of random variables. We formulate a dependence model to estimate the posterior probability of an instance being incorrectly labeled, given the dataset, and rank instances based on the estimated posterior. We provide an iterative framework for label noise reduction using our dependence model for noise ranking, as shown in Fig. 1. The iterative framework is used to first learn instance representations from the noisy dataset, and detect label noise, then fine-tune on denoised (cleaned) subsets in order to improve classification and learned representations, which iteratively improve label noise detection.

Our approach addresses several shortcomings of existing methods. First, our proposed method "NoiseRank" removes dependence on supervision for label noise detection. This allows wider practical applicability of our method to real domains. In contrast, most supervised approaches dealing with label noise are error-prone and hardly scalable. Second, our proposed framework for label noise ranking and improving classification is agnostic to both the classifier architecture and its training procedure. The implication of this is that we can train classifiers on any domain (image, text, multi-modal, etc.) within the same framework, using any standard classification architecture and optimization framework. In comparison, methods such as [1, 7, 16, 11] require careful network initialization and regularization of the loss function for optimization. Lastly, NoiseRank's un-

derlying algorithm and its output are human interpretable by design. Our main contributions are summarized as follows:

- A fully unsupervised label noise detection approach which is a probabilistic dependence model estimating the likelihood of being mislabeled. It does not require ground-truth labels, or priors on label or noise distribution.
- The proposed framework is generic, i.e., independent of application domain and content modality, and applicable with any standard classifier model architecture and optimization framework, unlike many recent unsupervised approaches [38, 8, 1, 7, 25].
- Its underlying algorithm and output are human interpretable. Again many unsupervised methods do not incorporate interpretability [38, 25, 10], which reduces their transparency in label noise detection and ranking.
- Experiments on real noise benchmark datasets, Food101-N (∼20% noise) and Clothing-1M (∼40% noise) for label noise detection and classification tasks, which improved state-of-the-art classification on Food101-N.

2 Related Work

Robust and noise-tolerant classifiers: Methods that focus on training noise tolerant or robust classifiers attempt to directly modify the training framework for learning in the presence of label noise. [11] introduces a non-linear noise modeling layer in a text classifier architecture to encode the distribution of label noise. [1] fits a beta mixture model on the training loss distribution to estimate the likelihood of label noise and uses that to guide the classifier training with a carefully selected loss function based on bootstrapping [23] and mix-up data augmentation [39]. Approaches based on meta-learning and curriculum learning are also studied for modifying the training procedure, where training samples are either ordered based on learning difficulty or mixed with synthetic noise distributions [7, 16, 10]. However, these methods limit the choice of classifier architectures, and furthermore are known to work only with careful initialization [11] and regularization [1] needed for convergence.

Label noise reduction/correction: Other methods, including ours, are based on label noise reduction, that attempt to detect, and remove or correct label noise. Prominent approaches utilize supervision to guide label noise detection. [28, 34] require clean (ground-truth) labels for a subset of the data to learn a mapping from noisy to clean labels. [15] requires binary verification labels instead, which indicate whether the given label is correct or noisy, in order to train an attention mechanism that can select reference images as class prototypes, and learn to predict if a given label is noisy. Similar to [15], [8] uses prototypes (more than one per class) to generate corrected labels which are then employed to iteratively train a network. However, [8] does not rely on any supervision or assumptions on the label distribution. As compared to [8], our method uses standard cross-entropy loss, whereas their framework is based on self-supervised learning, limiting flexibility on classifier optimization framework.

Another iterative approach is of [38], which updates both network parameters and label distributions to iteratively correct the noisy labels. [20] relies on the availability of a noise transition matrix for loss correction when training a classifier, which specifies the noise distribution in terms of the probability of one class being mislabeled as another. [32] employs a deep learning based risk consistent estimator to fine-tune a noise transition matrix. One type of unsupervised approach is outlier removal [33, 21]. However, outliers are not necessarily mislabeled and removing them presents a challenge [5]. There are also several methods addressing instance selection for kNN classification, which retain a subset of instances that allow correct classification of the remaining instances [9, 6, 3, 4, 19], or remove instances whose labels are different from the majority labels of their nearest neighbors [31, 18, 14]. However, the proposed heuristics have been criticized for removing too many instances or keeping mislabeled instances [5]. Our approach is related to these methods but focuses on leveraging both label (in)consistencies to globally rank noisy candidates and more effectively detect mislabeled instances even without any supervision. Weakly-supervised methods based on classification filtering such as [27] remove samples misclassified by SVM trained on the noisy data. However, it could amount to removing non-noisy hard samples or not removing noisy samples that the classifier mistakenly fits.

3 Unsupervised Label Noise Reduction and Model Training Framework

Our ultimate goal is to learn an effective classifier from the noisy labeled dataset without any form of human supervision (i.e., label verification), prior knowledge on the target domain, or label/noise distribution. In this section, we elaborate our multi-step model training framework. As is illustrated in Fig. 1, we first describe vector representation, which is necessary for similarity measure and label prediction in our framework. Next, our proposed probabilistic dependence model "NoiseRank" is elaborated for ranking dataset examples based on their likelihood of being noisy. Finally, we discuss the iterative model training steps.

3.1 Vector Representation

The vector space representation (i.e., embeddings) is a core component of our design because we rely on the vector representations to determine content similarity between examples in the given noisy dataset. Our framework is agnostic to any modality as well as the solution for learning representations. However, a high-quality representation improves the similarity measure and thus our unsupervised method for label noise detection. In Sec. 3.3, we will discuss how we improve the representation through iterative training.

With the vector representation, we could determine the content similarity. More formally, let the noisy labeled dataset be denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^m$ is the vector representation for example i, and $y_i \in \{1, 2 \dots C\}$ is the given label (potentially incorrect) with $C \geq 2$ being the total number of

classes in the dataset. In this work, we define instance similarity in terms of Euclidean distance between x_i and x_j as $d(x_i, x_j) = \|x_i - x_j\|_2$.

3.2 Label Noise Detection

In this section, we first describe our process for generating class prototypes, that are representative instances selected for each of the C classes in the dataset; followed by our non-parametric approach to generate label predictions, y'_i, for each prototype i. Let $Y = \{y'_i\}_{i=1}^{P}$ denote the predictions. Next, we elaborate the proposed dependence model, named NoiseRank, to globally rank the dataset examples based on their likelihood of having incorrect labels given their vector representations, labels and predictions.

Generating class prototypes Each of the C classes in the dataset can be represented by a set of class prototypes, i.e. a representative subset of instances in that class. We select the prototypes using K-means clustering on the vector space representations of instances in each class, given by the noisy labels. As a rule of thumb [13], we select $\left\lfloor \sqrt{\rho/2} \right\rfloor$ cluster centroids per class, where ρ is the average number of instances per class in the dataset. Selecting class prototypes is beneficial towards improving scalability when the number of dataset instances grows. We find that it is also important for robustness in high noise datasets, and K-means based selection is effective compared to randomly selected prototypes.

Generating label predictions For each prototype instance i represented by vector x_i, we generate the predicted label y'_i by a weighted k nearest neighbor classifier, as specified in Eq. 1.

$$y'_i = \underset{v \in \{1,2 \ldots C\}}{\arg\max} \sum_{x_j \in \mathcal{N}(x_i)} \kappa(x_i, x_j) \mathbb{1}\{y_j = v\} \tag{1}$$

where $\mathbb{1}$ is the indicator function, and the distance kernel function $\kappa(x_i, x_j)$ is used to weigh the contribution of each neighbor x_j in the neighborhood $\mathcal{N}$ comprising the k nearest neighbors of x_i:

$$\kappa(x_i, x_j) = \frac{1}{b + d(x_i, x_j)^e} \tag{2}$$

where $d(x_i, x_j)$ is the distance function discussed in Sec. 3.1, and $b > 0$ and $e > 0$ are parameters for the bias and weight exponent, respectively. The kernel function is negatively correlated to the distance function. For example, when $e = 2$, the kernel will be inversely proportional to the squared distance between the instances. Since $0 \leq d(x_i, x_j) \leq \infty$, by setting a positive bias b, we can prevent $\kappa(x_i, x_j)$ from being undefined when $d = 0$.

Dependence Model Formulation Our formulation is to estimate the posterior probability $P(x_i, y_i|\mathcal{D}, Y)$ that indicates the likelihood of label noise for all examples (x_i, y_i) in the dataset $\mathcal{D}$ and rank them based on this estimate. For this purpose, we use Markov Random Fields (also known as MRFs or "dependence models" [17, 37]) which provide a generic framework for modeling the joint distribution of a large set of random variables.

In dependence models, conditional dependencies are defined only for certain groups of random variables called "cliques", and are represented with edges in an undirected graph. We represent the graph with G and the cliques in the graph as $C(G)$ in our formulations. For each type of clique $c \in C(G)$, we define a non-negative potential function $\phi(c; \Lambda)$ parameterized by Λ. The joint probabilities are estimated based on the Markov assumption as follows:

$$P(x_i, y_i, \mathcal{D}, Y) = \frac{1}{Z} \prod_{c \in C(G)} \phi(c; \Lambda) \tag{3}$$

where $Z = \sum_{x_i, y_i, \mathcal{D}, Y} \prod_{c \in C(G)} \phi(c; \Lambda)$ is a normalization term. Computing Z is very expensive due to the large number of summands. Since our aim is to rank examples in the dataset based on their posterior probabilities $P(x_i, y_i|\mathcal{D}, Y)$ and ignoring Z in this formulation does not change the ranking result, the posterior probability is estimated as follows:

$$\begin{aligned} P(x_i, y_i|\mathcal{D}, Y) &= \frac{P(x_i, y_i, \mathcal{D}, Y)}{P(\mathcal{D}, Y)} \\ &\overset{rank}{=} \log P(x_i, y_i, \mathcal{D}, Y) - \log P(\mathcal{D}, Y) \\ &\overset{rank}{=} \sum_{c \in C(G)} \log \phi(c; \Lambda) \end{aligned} \tag{4}$$

where $\overset{rank}{=}$ indicates rank equivalence. The formulation is a sum of logarithm of potential functions over all cliques. For simplification purposes, the potential function is assumed to be $\phi(c; \Lambda) = \exp(\lambda_c f(c))$, where $f(c)$ is the feature function over the clique c and λ_c is the weight for the feature function. The final ranking function is computationally tractable and linear over feature functions:

$$P(x_i, y_i|\mathcal{D}, Y) \overset{rank}{=} \sum_{c \in C(G)} \lambda_c f(c) \tag{5}$$

Depending on the choice of the feature functions and their corresponding weights, the final ranking score in 5 can be negative. In the next subsection, we elaborate our dependence model for the task of label noise detection by explicitly defining each clique, its feature function $f(c)$ and the corresponding weight λ_c.

Dependence Graph Construction In our formulation, we define cliques between all pairs of examples (i, j) where $i \neq j$ and $i \in \mathcal{D}, j \in P$ for dataset $\mathcal{D}$ with size N and set of all prototypes $P \subseteq \mathcal{D}$. There are $\mathcal{O}(N|P|)$ cliques defined in the

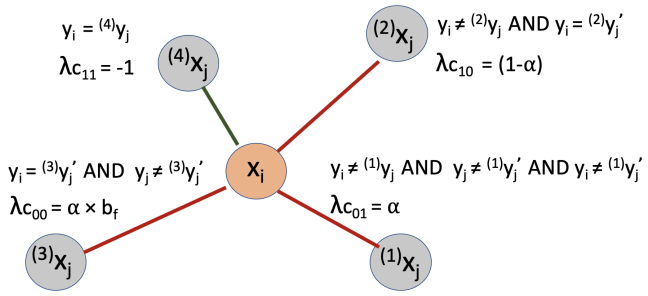


Fig. 2: The dependence graph illustrated for a given example i in a database containing five examples. Clique types and weights are determined based on the given y_i and y_j and predicted labels y'_j. Edge lengths indicate distance in the vector representation space

dependence graph. Each example is associated with its given label y_i and each prototype is associated with its given label y_j and predicted label y'_j, which are used for determining the clique weights as shown in Fig. 2 and explained below. All cliques are assumed to share the same feature function $f(c) = \kappa(x_i, x_j)$ as defined by the kernel function in Eq. 2.

We differentiate cliques into four types based on the values of the given and predicted labels of the examples. The first clique type, denoted by c_{11}, is for all pairs of examples (i, j) that share the same given label. If the examples share the same label (i.e., $y_i = y_j$), we assign a negative "blame" score (i.e., reward) weighted by $f(c)$ to example i so that it ranks lower in the final rank list of examples sorted by their overall MRF scores. For this clique type, we set the clique weight parameter as $\lambda_c = -1$.

The second clique type is denoted by c_{10}. It is defined for all pairs of examples (i, j) with different labels (i.e., $y_i \neq y_j$) where $y'_j = y_j$. In this case, example j blames example i for providing an incorrect prediction vote, even though the false vote did not change the prediction output y'_j which is consistent with example j's original label y_j. For this type, we set $\lambda_c = 1 - \alpha$, where α is a hyper-parameter defined in the range $[0.5, 1]$ to control the impact of incorrect vote (i.e., $y'_j \neq y_i$) on the blame score.

The third clique type, denoted by c_{01}, is for all pairs of examples (i, j) with different labels (i.e., $y_i \neq y_j$) where $y'_j \neq y_j$ and $y'_j \neq y_i$. In other words, the prediction output y'_j is different from both its own label y_j and example i's label y_i. While example i did not directly influence the mispredicted label, it did not contribute towards the correct prediction. By setting $\lambda_c = \alpha$, example j assigns a scaled blame score to example i.

The fourth clique type, denoted by c_{00}, is for all pairs of examples (i, j) with different labels (i.e., $y_i \neq y_j$) where $y'_j \neq y_j$ and $y'_j = y_i$. Example j blames example i strongly for supporting a prediction different from its own label y_j which is the same as its prediction y'_j. For this type, we set $\lambda_c = \alpha \times b_f$ where b_f in $[1, inf)$ is the "blame factor" and controls the strength of the blame.

NoiseRank Score Function The final ranking function in Eq. 6 is the sum of all blame and reward scores accumulated for example i according to Eq. 5.

$$\begin{aligned} P(x_i, y_i|\mathcal{D}, Y) \stackrel{rank}{=} & \sum_{(x_i,x_j)\in c_{11}} -\kappa(x_i, x_j) \\ + & \sum_{(x_i,x_j)\in c_{10}} (1-\alpha)\kappa(x_i, x_j) + \sum_{(x_i,x_j)\in c_{01}} \alpha\kappa(x_i, x_j) \\ + & \sum_{(x_i,x_j)\in c_{00}} (\alpha \times b_f)\kappa(x_i, x_j) \end{aligned} \tag{6}$$

The aggregate score $P(x_i, y_i|\mathcal{D}, Y)$ is the basis for ranking instances in the dataset. The rank reflects the relative likelihood of being mislabeled and the impact on mispredictions, accounted for in the penalty function. Since the score function is unbounded, detecting label noise given the ranked list requires threshold $\delta \geq 0$ to determine if an instance with detected label noise should be retained ($w = 1$) or removed ($w = 0$) from the dataset.

$$w(x_i) = \begin{cases} 0, & \text{if } P(x_i, y_i|\mathcal{D}, Y) > \delta \\ 1, & \text{otherwise} \end{cases} \tag{7}$$

It should be noted that as the dataset size increases, computing the ranking function over all $\mathcal{O}(N|P|)$ pairs is less efficient. Moreover, the value of the feature function $f(c)$ approaches zero as distances between pairs increase. We therefore limit the cliques to the k closest neighbors of example i for assigning blame and reward scores as defined above. This approximation is quite effective especially because the blame and reward scores diminish rapidly and approach zero as distances get larger. Another note is that we use the same k value in the score function and in the kernel function (Eq. 2) for both computing the aggregate rank score function and generating label predictions y'_j for simplification purposes.

3.3 Iterative Training

We provide a generic iterative framework to learn classifiers with label noise reduction. As described earlier, the vector space representations of examples in the dataset are used in determining content similarity for noise ranking. Initially, representations can be learned with the available (potentially noisy) labels. In order to improve the learned representations, we can iterate over representation learning, noise ranking and reduction, model training, in order.

The framework is agnostic to the model used for representation learning and classification, depending on the content modality. At a first step, we train the classifier model (eg. standard CNN with simple cross-entropy loss) with the available noisy dataset D. The classifier can be used to extract representations for examples in the dataset, which are used to run label noise detection with NoiseRank and remove the examples that are ranked as noisy (i.e., $w(x_i) = 0$). Finally, we fine-tune the trained model with the denoised subset of the dataset.

Table 1: Dataset Statistics. We use only the noisy (train) labels in NoiseRank. Verified labels (train/validation) are used in other supervised/weakly-supervised methods

Dataset	# Classes	# Train	# Verified (tr/va)	# Test
Food-101N	101	310K	55k/5k	25k
Clothing1M	14	1M	25k/7k	10k
YFCC100m	1000	99.2M	-/-	50k

4 Experiments

We report experiments on three public datasets: Food-101N, Clothing1M and YFCC100m on both label noise detection and classification.

Food-101N [15] and **Clothing-1M** [34]: These are **real noise** public datasets collected from noisy channels; which are used to study methods for learning in the presence of label noise. These datasets also contain additional verification/clean labels used for noise detection training and validation by supervised label noise reduction methods. Note that for NoiseRank, we do not use these additional verified/clean labels in training or validation. We only use the verified validation labels for evaluation of our method to report results on label noise detection recall and accuracy. These datasets also provide a clean test set with 25K and 10K examples respectively, used for evaluation of the classification task top-1 accuracy. **YFCC100m** [26]: This is a large-scale dataset with 99.2M images used in the semi-supervised learning setting in [36] and we combine it with NoiseRank for detecting label noise in machine-generated labels originated by the semi-supervised learning setup. We use NoiseRank to detect and remove mislabeled examples; and the rest are then leveraged to improve target ImageNet-1k classification. Dataset statistics are summarized in Table 1.

4.1 Experiment setup and hyper-parameters

For representation learning, we introduced a 256-dimensional bottleneck layer to the ResNet-50 model pre-trained on ImageNet1k. First, the pre-trained ResNet-50 is fully fine-tuned with the entire noisy dataset using learning rate 0.002 for [10,10,10] epochs and learning rate decay rate 0.1. The output of the bottleneck layer is L2 normalized and used for representing image content. We report results for NoiseRank which conducts label noise detection and removal only once; and iterative NoiseRank wherein after one round of noise removal we fine-tune the ResNet and repeat noise removal. For efficient nearest neighbor search, we use open-source library FAISS [12] which takes less than 10 minutes on one GPU for dataset of size 1M with the 256d vector representations.

Unsupervised hyper-parameter selection: The improvement in data quality can be directly measured (without supervision from verified labels) by the improvement in learnability of the classifier. We measure the training loss at epoch 10 on denoised subsets and select NoiseRank hyper-parameter setting that results in the least training loss. To reduce the parameter search, we

Table 2: Label noise detection accuracy. Left: average error rate over all the classes (%) Right: Label noise recall, F1 and macro-F1 (%). NoiseRank(I) is iterative NoiseRank

	Average error rate	
Method	Food-101N	Clothing-1M
Supervised		
MLP	10.42	16.09
Label Prop [35]	13.24	17.81
Label Spread [40]	12.03	17.71
CleanNet [15]	6.99	15.77
Weakly-Supervised		
Cls. Filt.	16.60	23.55
Avg. Base. [15]	16.20	30.56
Unsupervised		
DRAE [33]	18.70	38.95
unsup-kNN	26.63	43.31
NoiseRank	24.02	23.54
NoiseRank (I)	**18.43**	**22.81**

Method	Type	Recall	F1	MacroF1
Food-101N (19.66% estimated noise)				
CleanNet	sup.	71.06	**74.01**	**84.04**
Avg. Base.	weakly	47.70	59.57	76.08
unsup-kNN	unsup.	22.02	24.23	54.03
NoiseRank (I)	unsup.	**85.61**	64.42	76.06
Clothing-1M (38.46% estimated noise)				
CleanNet	sup.	69.40	**73.99**	**79.65**
Avg. Base.	weakly	43.92	55.14	67.65
unsup-kNN	unsup.	10.85	16.60	44.26
NoiseRank (I)	unsup.	**74.18**	71.74	76.52

first select and fix the best k (number of nearest neighbors) by grid search in $\{5, 10, 20, 50, 100, 250\}$ and then search $\alpha \in \{0.5, 0.6, 0.8\}$ and $b_f \in \{1.0, 1.5, 2.0\}$, and $b = e = 1$ in the distance kernel. The ranking cut-off $\delta = 0$.

4.2 Label Noise Detection Experiments

We report the effectiveness of our proposed method on detecting label noise in Table 2, in terms of i) averaged detection error rate over all classes in Food101-N and Clothing-1M, and ii) in terms of label noise recall and F1. Table 2 details the average error rate of label noise detection on the verified validation set compared against a wide range of baselines, as reported in [15]. The naive baseline predicts all samples as correctly labeled, and therefore its error rate approximates the true noise distribution assuming a random selection of the ground truth set. Clothing-1M has a significant amount of noise estimated at 38.46%. In this significantly noisy dataset, iterative NoiseRank even as an unsupervised method, strongly outperforms unsupervised outlier removal method DRAE [33] by a large margin of 16.15% (which is 40% error reduction) and weakly supervised Average Baseline (Avg. Base.) [15] by 7.75% (which is 25% error reduction) on avg error rate. This is state-of-the-art noise detection error rate among unsupervised alternatives on this dataset. Avg. Base. computes the cosine similarity between an instance representation and the averaged representation of a class; and although it does not use verified labels in training, it uses them to select the threshold on cosine similarity for label noise detection. On Food101-N the estimated noise is 19.66% and the avg error rate of iterative NoiseRank and DRAE are comparable.

However, since noise vs. clean instance distribution is imbalanced, we further measure recall, F1 and macro-F1 scores for label noise detection in Table 2. NoiseRank has state-of-the-art recall of 85.61% on Food-101N and 74.18% on Clothing-1M. NoiseRank F1/MacroF1 is competitive with the best supervised

Table 3: Image classification on Food-101N results in terms of top-1 accuracy (%). Train data (310k) and test data (25k). CleanNet is trained with an additional 55k/5k (tr/va) verification labels to provide the required supervision on noise detection

#	Method	Training	Pre-training	Top-1
1	None [15]	noisy train	ImageNet	81.44
2	CleanNet [15]	noisy(+verified)	ImageNet	83.95
3	DeepSelf [8]	noisy train	ImageNet	85.11
4	NoiseRank	cleaned train	ImageNet	85.20
5		cleaned train	noisy train #1	**85.78**

method in noise detection CleanNet [15] which requires verified labels in training and validation, and thus has a significant advantage compared to unsupervised and weakly-supervised methods. It should be noted that effective noise recall directly impacts classification, and is therefore an important evaluation metric for label noise detection and removal.

4.3 Classification Experiments

We conducted experiments to study the impact of data quality on the classification task using the ResNet-50 classifier pretrained on ImageNet and initially fine-tuned on noisy dataset and later on denoised subset with NoiseRank. In results table 3 and 4, in each row, the model is fine-tuned with the mentioned training examples on the specified pre-trained model (eg. "noisy train # 1" refers to the model # 1 referenced in the table that was trained using noisy training samples on ImageNet pre-trained Resnet-50). Similarly, in Table 4, "# 4" in the pre-training column, refers to model # 4 indicated in the table.

In Table 3 Food101-N, NoiseRank achieves state of the art 85.78% in top-1 accuracy compared to unsupervised [8]'s 85.11%, and 11% error reduction over supervised noise reduction method CleanNet. This can be attributed to the high noise recall on Food-101N as examined earlier. In Table 4 Clothing-1M, NoiseRank used to reduce label noise in noisy train ($\sim$40% estimated noise) is effective in improving classification from 68.94% to 73.82% (16% error reduction), even without supervision from clean set in high noise regime, and performs comparable to recent unsupervised [8] and marginally outperforms unsupervised PENCIL [38]. In contrast to [8] and [38], NoiseRank framework allows for flexible choice of classifier and optimization/loss function, and yet achieves comparable improvement due to noise reduction, using standard cross-entropy loss and standard training framework. This underlines the benefits of the proposed framework without compromising on classification improvements. Supervised baselines CleanNet and Loss Correction respectively utilize additional verified labels, and yet the performance gain from noise removal for ours is highly competitive, even in this high noise regime. Lastly, we also reported results of fine-tuning each method with an additional clean 50k set, as per the setting followed in [20]. [8] achieves best result of 81.16% with clean 50k sample set. We note that even

Table 4: Image classification on Clothing-1M results in terms of top-1 accuracy (%). Train data (1M) and test data (10k). CleanNet and Loss Correction are trained with an additional 25k/7k (train/validation) verification labels to provide required supervision on noise detection/correction

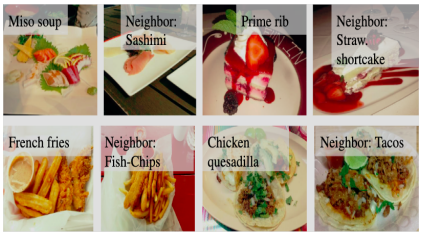
#	Method	Training	Pre-training	Top-1
1	None [20]	clean50k	ImageNet	75.19
2	None [20]	noisy train	ImageNet	68.94
3		clean50k	noisy train # 2	79.43
4	loss cor.[20]	noisy(+verified)	ImageNet	69.84
5		clean50k	# 4	80.38
6	Joint opt. [25]	noisy train	ImageNet	72.16
7	PENCIL [38]	noisy train	ImageNet	73.49
8	CleanNet [15]	noisy(+verified)	ImageNet	74.69
9		clean50k	# 8	79.90
10	DeepSelf [8]	noisy train	ImageNet	74.45
11		clean50k	# 10	**81.16**
12		cleaned train	ImageNet	73.77
13	NoiseRank	cleaned train	noisy train #2	73.82
14		clean50k	# 13	79.57

without noise correction, the inclusion of the clean set boosts accuracy from 68.94% to 79.43% and may shadow the benefit of noise removal; with CleanNet [15] at 79.90% and ours at 79.57% being comparable in this setting.

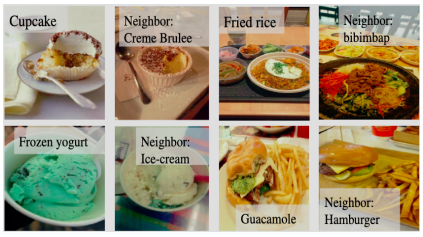
Table 5: Left: ImageNet benchmark top-1 accuracy (%). NoiseRank with removal of top x% ranked instances in noisy machine generated labels, against random removal. Right: Examples of noisy machine generated labels detected by NoiseRank (M: mislabeled instances, C: correctly labeled instances mistakenly identified by NoiseRank)

Method	Top-1 Accuracy
None [36]	79.06
Top 0.6% removed	79.13
Top 1.2% removed	79.12
Top 1.8% removed	**79.34**
Random 1.8% removed	78.96

In Table 5, we report semi-supervised learning results on large YFCC100m dataset, with and without label denoising. [36] is used to train a ResNet-101 to label images in YFCC100m into 1K ImageNet classes. 16K images from each class that have the most confident machine label predictions are retained. However, even after filtering, these labels contain noise as shown by examples detected using NoiseRank (right: Table 5). We run NoiseRank to remove label noise from the 16M machine labeled images. The denoised images are used to pre-train

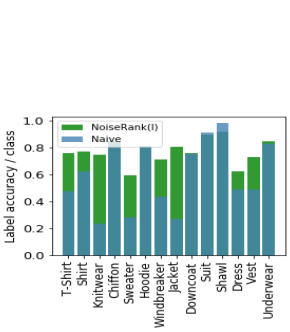


Examples correctly detected as label noise in Food-101N shown along with its nearest neighbor

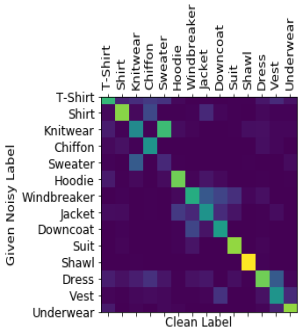


Examples incorrectly identified as label noise by NoiseRank, shown along with their nearest neighbors (top row); Examples identified as label noise by NoiseRank but are wrongly verified by humans (bottom row) in Food101-N

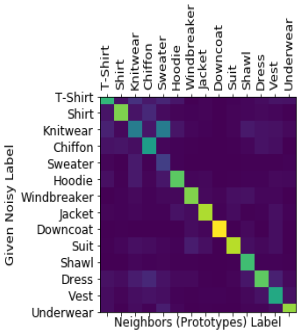
Fig. 4: Interpretability analysis of NoiseRank predictions on Food101-N



Accuracy per class for label noise detection



Noise transition matrix between classes, estimated on validation set



NoiseRank score function matrix between class types

Fig. 6: Interpretability analysis of NoiseRank predictions on Clothing-1M

ResNet-50, then fine-tuned with ImageNet-1K train set and evaluated on benchmark ImageNet-1K test set. The top-1 accuracy without noise removal is 79.06% and with noise removal is 79.34%, in comparison to removing the same number of random instances (78.96%) averaged over three runs. Note that in this setting, noise removal is not applied directly to the target classification task, but rather to the dataset used to pre-train the model, later trained on the target dataset.

4.4 Interpretability Analysis

Interpretability is useful in providing explanations about the predictions made by machine learning algorithms. NoiseRank is a transparent framework that can be easily used to provide human level analysis of why a given example was predicted as mislabeled or not mislabeled.

In Fig. 3a, we show sample images correctly identified by NoiseRank as label noise in the Food-101N dataset, along with their nearest neighbors. The nearest neighbors provide supporting visual evidence towards understanding the prediction made by NoiseRank. Interpretability is also useful for identifying hard instances; to support building better datasets and models. In Fig. 3b (top row) we show sample images incorrectly identified as label noise in Food-101N. As seen, these are tough examples with contradictory labels to their nearest neighbors, which provides insights into when and which instances might be confused with others. The bottom row shows sample images identified as label noise by NoiseRank but verified (seemingly incorrectly) by humans. Such samples further justify our belief that human label verification can also be prone to errors.

In Fig. 6, we provide class-wise analysis on the 14 Clothing-1M dataset class types. Fig. 5a reports noise detection accuracy per class (NoiseRank), compared to the original noise ratio in the dataset (Naive). Fig 5b provides the estimated probability of flipping a clean label of one class to another (noise transition matrix) estimated on the validation set. In Fig. 5c, we visualize how NoiseRank scores each example based on its neighboring prototypes. Each cell in 5c maps a given noisy label class and its prototype neighbors' class aggregated over the pairs used in the scoring function (Eq. 6), with weight equal to its contribution in the score function; and the matrix is then column-normalized for the distribution. It implicitly encodes the noise transition probability between class types without any knowledge of the clean labels, as seen from its similarity to 5b.

5 Conclusion

In this paper, we proposed an unsupervised label noise ranking algorithm based on Markov Random Fields. NoiseRank is unsupervised, interpretable, and agnostic to the downstream task or classifier architecture and training. It is applicable to any domain/modality with standard widely used deep learning models or classifiers, without constraining the choice of model, loss function, or optimizer.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N., Mcguinness, K.: Unsupervised label noise modeling and loss correction. In: International Conference on Machine Learning. pp. 312–321 (2019)
2. Corbiere, C., Ben-Younes, H., Ramé, A., Ollion, C.: Leveraging weakly annotated data for fashion image retrieval and label prediction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2268–2274 (2017)
3. Delany, S.J., Cunningham, P.: An analysis of case-base editing in a spam filtering system. In: European Conference on Case-Based Reasoning. pp. 128–141. Springer (2004)
4. Franco, A., Maltoni, D., Nanni, L.: Data pre-processing through reward–punishment editing. Pattern Analysis and Applications **13**(4), 367–381 (2010)
5. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems **25**(5), 845–869 (2013)
6. Gates, G.: The reduced nearest neighbor rule (corresp.). IEEE transactions on information theory **18**(3), 431–433 (1972)
7. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculumnet: Weakly supervised learning from large-scale web images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
8. Han, J., Luo, P., Wang, X.: Deep self-learning from noisy labels. In: 2019 IEEE international conference on computer vision. pp. 5138–5147. Ieee (2019)
9. Hart, P.: The condensed nearest neighbor rule (corresp.). IEEE transactions on information theory **14**(3), 515–516 (1968)
10. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. pp. 2309–2318 (2018)
11. Jindal, I., Pressel, D., Lester, B., Nokleby, M.: An effective label noise model for dnn text classification. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3246–3256 (2019)
12. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)
13. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in k-means clustering. International Journal **1**(6), 90–95 (2013)
14. Lallich, S., Muhlenbach, F., Zighed, D.A.: Improving classification by removing or relabeling mislabeled instances. In: International Symposium on Methodologies for Intelligent Systems. pp. 5–15. Springer (2002)
15. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5447–5456 (2018)
16. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5051–5059 (2019)
17. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 472–479. SIGIR '05, ACM, New York, NY, USA (2005). https://doi.org/10.1145/1076034.1076115, http://doi.acm.org/10.1145/1076034.1076115

18. Muhlenbach, F., Lallich, S., Zighed, D.A.: Identifying and handling mislabelled instances. Journal of Intelligent Information Systems **22**(1), 89–109 (2004)
19. Nanni, L., Franco, A.: Reduced reward-punishment editing for building ensembles of classifiers. Expert Systems with Applications **38**(3), 2395–2400 (2011)
20. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1944–1952 (2017)
21. Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., et al.: Estimating the support of a high-dimensional distribution (1999)
22. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M.P., Shyu, M.L., Chen, S.C., Iyengar, S.: A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR) **51**(5), 92 (2019)
23. Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. In: International Conference on Learning Representations (2015)
24. Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., Wojatzki, M.: Measuring the reliability of hate speech annotations: The case of the european refugee crisis. arXiv preprint arXiv:1701.08118 (2017)
25. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: 2018 IEEE conference on computer vision and pattern recognition. pp. 5552–5560. Ieee (2018)
26. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. arXiv preprint arXiv:1503.01817 (2015)
27. Thongkam, J., Xu, G., Zhang, Y., Huang, F.: Support vector machine for outlier detection in breast cancer survivability prediction. In: Asia-Pacific Web Conference. pp. 99–109. Springer (2008)
28. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 839–847 (2017)
29. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision **101**(1), 184–204 (2013)
30. Waseem, Z.: Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the first workshop on NLP and computational social science. pp. 138–142 (2016)
31. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics (3), 408–421 (1972)
32. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? In: NeurIPS (2019)
33. Xia, Y., Cao, X., Wen, F., Hua, G., Sun, J.: Learning discriminative reconstructions for unsupervised outlier removal. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1511–1519 (2015)
34. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2691–2699 (2015)
35. Xiaojin, Z., Zoubin, G.: Learning from labeled and unlabeled data with label propagation. Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University (2002)

36. Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)
37. Yalniz, I.Z., Manmatha, R.: Dependence models for searching text in document images. IEEE Trans. Pattern Anal. Mach. Intell. **41**(1), 49–63 (2019). https://doi.org/10.1109/TPAMI.2017.2780108, https://doi.org/10.1109/TPAMI.2017.2780108
38. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: 2019 IEEE conference on computer vision and pattern recognition. Ieee (2019)
39. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
40. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)