Deep k-NN for Noisy Labels

Dara Bahri [1] **Heinrich Jiang** [1] **Maya Gupta** [1]

Abstract

Modern machine learning models are often trained on examples with noisy labels that hurt performance and are hard to identify. In this paper, we provide an empirical study showing that a simple k-nearest neighbor-based filtering approach on the logit layer of a preliminary model can remove mislabeled training data and produce more accurate models than many recently proposed methods. We also provide new statistical guarantees into its efficacy.

1. Introduction

Machine learned models can only be as good as the data they were used to train on. With increasingly large modern datasets and automated and indirect labels like clicks, it is becoming ever more important to investigate and provide effective techniques to handle noisy labels.

We revisit the classic method of filtering out suspicious training examples using k-nearest neighbors (k-NN) (Wilson, 1972). Like Papernot & McDaniel (2018), we use a *deep k-NN*, in that we use a k-NN on learned intermediate representations of a preliminary model to identify suspicious examples for filtering. Recently, k-NN methods have been receiving renewed attention for their usefulness (Wang et al., 2018; Reeve & Kaban, 2019). Here, like Jiang et al. (2018), we use k-NN as an auxiliary method to improve modern deep learning.

The main contributions of this paper are both experimental and theoretical. *Experimentally,* we show that deep k-NN works as well or better than state-of-art methods for handling noisy labels, and it is robust to the choice of k. Furthermore, we show the method is effective both with and without access to a set of known cleanly labeled training examples. *Theoretically,* we show that k-NN's predictions will, asymptotically, only identify a training example as clean if its label is the Bayes-optimal label. We also provide finite-sample analysis in terms of the margin and how spread-out the corrupted labels are (Theorem 1), rates of convergence for the margin (Theorem 2) and rates under Tsybakov's noise condition (Theorem 3) with all rates matching minimax-optimal rates in the noiseless setting.

Our work shows that even though the preliminary neural network is trained with corrupted labels, it still yields intermediate representations that are useful for deep k-NN filtering. After identifying examples whose labels disagree with their neighbors, one can either automatically remove them, or send them to a human operator for further review. This strategy can also be useful in human-in-the-loop systems where one can warn the human annotator that a label is suspicious and automatically propose new labels based on its nearest neighbors.

In addition to strong empirical performance, deep k-NN filtering has a few advantages. Firstly, many methods require a clean set of samples whose labels can be trusted. We show that the k-NN based method is effective both in the presence and absence of a clean set of samples. Secondly, while k-NN does introduce the hyperparameter k, we will show that deep k-NN filtering is stable to the choice of k: such robustness to hyperparameters is highly desirable as optimal tuning for is often challenging for this problem.

2. Related Work

We review relevant prior work for training on noisy labels in Sec. 2.1 and related k-NN theory in 2.2.

2.1. Training with Noisy Labels

Methods to handle label noise can be classified into two main strategies: (1) explicitly identify and remove the noisy examples, and (2) indirectly handle the noise with robust training methods.

Data Cleaning: The proposed deep k-NN filtering fits into the broad family of *data cleaning* methods, in that our proposal detects and filters *dirty data* (Chu et al., 2016). Using k-NN to "edit" training data has been popular since Wilson (1972) used it to throw away training examples that were not consistent with their $k = 3$ nearest neighbors. The idea of using a preliminary model to help identify mislabeled ex-

[1] Google Research. Correspondence to: Dara Bahri <dbahri@google.com>, Heinrich Jiang <heinrichj@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

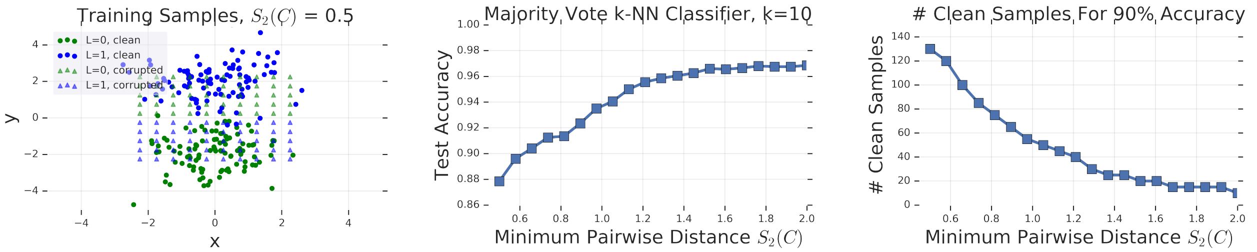

Figure 1. Impact of minimum pairwise distance $S_2(C)$ for binary classification on simulated Gaussian data. **Left:** Training samples. **Center:** Test accuracy improves as grid width, hence $S_2(C)$, increases. **Right:** With larger spread, fewer clean training samples are needed to achieve an accuracy of 90%.

amples dates to at least Guyon et al. (1994), who proposed using the model to compute an information gain for each example, and then considered examples with high gain to be suspect. Other early related work used a cross-validation set-up to (1) train a classifier on part of the data, (2) use it to make predictions for held-out training examples, and (3) remove any example whose label disagrees with the prediction (Brodley & Freidl, 1999).

Noise Corruption Estimation: For multi-class problems, a popular approach is to account for noisy labels by applying a confusion matrix after the model's softmax layer (Sukhbaatar et al., 2014). The confusion matrix, however, is often unknown and must be estimated. Patrini et al. (2017) suggest deriving it from the softmax distribution of the model trained on noisy data, but there are other alternatives (Goldberger & Ben-Reuven, 2017; Jindal et al., 2016; Han et al., 2018). Accurate estimates are generally hard to attain when only untrusted data is available. Hendrycks et al. (2018) achieve more accurate estimates in the setting where some amount of known clean, trusted data is available. EM-style algorithms have also been proposed to estimate the clean label distribution (Xiao et al., 2015; Khetan et al., 2017; Vahdat, 2017).

Noise-Robust Training: Natarajan et al. (2013) propose a method to make any surrogate loss function noise-robust given knowledge of the corruption rates. Ghosh et al. (2017) prove that losses like mean absolute error (MAE) are inherently robust under symmetric or uniform label noise while Zhang & Sabuncu (2018) show that training with MAE results in poor convergence and accuracy. They propose a new loss function based on the negative Box-Cox transformation that trades off the noise-robustness of MAE with the training efficiency of cross-entropy. Lastly, the ramp, unhinged, and savage losses have been proposed and theoretically justified to be noise-robust for support vector machines (Brooks, 2011; Van Rooyen et al., 2015; Masnadi-Shirazi & Vasconcelos, 2009). Amid et al. (2019) construct a noise-robust "bi-tempered" loss by introducing temperature in the exponential and log functions. Rolnick et al. (2017) empirically show that deep learning models are robust to noise when there are enough correctly labeled examples and when the model capacity and training batch size are sufficiently large. Thulasidasan et al. (2019) propose a new loss that enables the model to abstain from confusing examples during training.

Auxiliary Models: Veit et al. (2017) propose learning a label cleaning network on trusted data by predicting the differences between clean and noisy labels. Li et al. (2017) suggest training on a weighted average between noisy labels and distilled predictions of an auxiliary model trained on trusted data. Given a model pre-trained on noisy data, Lee et al. (2019) boost its generalization on clean data by inducing a generative classifier on top of hidden features from the model. The method is grounded in the assumption that the hidden features follow a class-specific Gaussian distribution.

Example Weighting: We make a hard decision about whether to keep a training example, but one can also adapt the weights on training examples based on the confidence in their labels. Liu & Tao (2015) provide an importance-weighting scheme for binary classification. Ren et al. (2018) suggest upweighting examples whose loss gradient is aligned with those of trusted examples at every step in training. Jiang et al. (2017) investigate a recurrent network that learns a sample weighting scheme to give to the base model.

2.2. k-Nearest Neighbor Theory

The theory of k-nearest neighbor classification has a long history (see e.g. Fix & Hodges Jr (1951); Cover (1968); Stone (1977); Devroye et al. (1994); Chaudhuri & Dasgupta (2014)). Much of the prior work focuses on k-NN's statistical consistency properties. However, with the growing interest in adversarial examples and learning with noisy labels, there have recently been analyses of k-nearest neighbor methods in these settings. Wang et al. (2018) analyze the robustness of k-NN classification and provide a robust

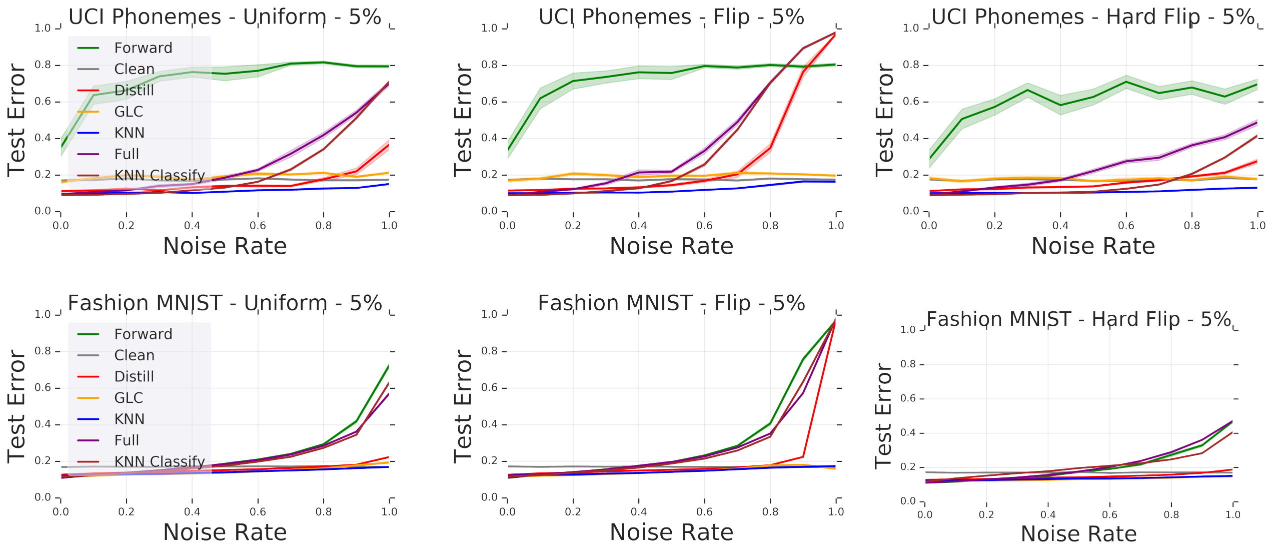

Figure 2. **Top row: UCI results, bottom row: Fashion MNIST results**. Test error for different amounts of noise applied to the labels of $\mathcal{D}_{\text{noisy}}$. $\mathcal{D}_{\text{clean}}$ contains 5% of the data. Rows correspond to datasets while columns correspond to corruption types. We see that the k-NN method consistently chooses the best examples to filter leading to lower error. More results are in the Appendix.

variant of 1-NN classification where their notion of robustness is that predictions of nearby points should be similar. Gao et al. (2016) provide an analysis of the k-NN classifier under noisy labels and like us, show that k-NN can attain similar rates in the noisy setting as in the noiseless setting. Gao et al. (2016) assume a noise model where labels are corrupted uniformly at random, while we assume an arbitrary corruption pattern and provide results based on a notion of how spread out the corrupted points are. Moreover, we provide finite-sample bounds borrowing recent advances in k-NN convergence theory in the noiseless setting (Jiang, 2019) while the guarantees of Gao et al. (2016) are asymptotic. Reeve & Kaban (2019) provide stronger guarantees on a robust modification of k-NN proposed by Gao et al. (2016). To the best of our knowledge, we provide the first finite-sample rates of consistency for the classic k-NN method in the noisy setting with few assumptions on the label noise.

3. Deep k-NN Algorithm

Recall the standard k-nearest neighbor classifier:

Definition 1 (k-NN)**.** *Let the k-NN radius of $x \in \mathcal{X}$ be $r_k(x) := \inf\{r : |B(x, r) \cap X| \geq k\}$ where $B(x, r) := \{x' \in \mathcal{X} : |x - x'| \leq r\}$ and the k-NN set of $x \in \mathcal{X}$ be $N_k(x) := B(x, r_k(x)) \cap X$. Then for all $x \in \mathcal{X}$, the k-NN classifier function w.r.t. X is*

$$\eta_k(x) := \arg\max_y \frac{1}{|N_k(x)|} \sum_{i=1}^{n} 1\left[y_i = y, x_i \in N_k(x)\right],$$

Our method is detailed in Algorithm 1. It takes a dataset $\mathcal{D}_{\text{noisy}}$ of examples with potentially noisy labels, along with a dataset $\mathcal{D}_{\text{clean}}$ consisting of clean or trusted labels. Note that we allow $\mathcal{D}_{\text{clean}}$ to be empty (i.e. in instances where no such trusted data is available). We also take as given a model architecture $\mathcal{A}$ (e.g. a 2-layer DNN with 20 hidden nodes). Algorithm 1 uses this sub-routine to select the initial train set:

Training Set Selection for Preliminary Model: To choose if the preliminary model used to compute the k-NN should be trained on $\mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{noisy}}$ or only on $\mathcal{D}_{\text{clean}}$, we split $\mathcal{D}_{\text{clean}}$ 70/30 into two sets: $\mathcal{D}_{\text{cleanTrain}}$ and $\mathcal{D}_{\text{cleanVal}}$, and then train two models: one on $\mathcal{D}_{\text{cleanTrain}}$ and one on $\mathcal{D}_{\text{cleanTrain}} \cup \mathcal{D}_{\text{noisy}}$. If the model trained with $\mathcal{D}_{\text{noisy}}$ performs better on the held-out $\mathcal{D}_{\text{cleanVal}}$, then output $\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{\text{clean}}$, otherwise output $\mathcal{D}_{\text{clean}}$.

Algorithm 1 Deep k-NN Filtering

```
Inputs: $\mathcal{D}_{\text{noisy}}$, $\mathcal{D}_{\text{clean}}$ (possibly empty), $k$, $\mathcal{A}$
1: Train a model $\mathcal{M}$ with architecture $\mathcal{A}$ on the output of
   Training Set Selection for Preliminary Model.
2: Let $\mathcal{N}_{\mathcal{M}}$ be the function that extracts activations on the
   logit layer of $\mathcal{M}$.
3: $\mathcal{D}_{\text{filtered}} := \{(x, y) \in \mathcal{D}_{\text{noisy}} : \eta_k(\mathcal{N}_{\mathcal{M}}(x)) = y\}$, where
   $\eta_k$ is computed w.r.t. $\mathcal{N}_{\mathcal{M}}(\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{\text{clean}})$.
4: Train final model with architecture $\mathcal{A}$ on $\mathcal{D}_{\text{filtered}} \cup \mathcal{D}_{\text{clean}}$.
```

4. Theoretical Analysis

For the theoretical analysis, we assume the binary classification problem with the features defined on compact set $\mathcal{X} \subseteq \mathbb{R}^D$. We assume that points are drawn according to distribution $\mathcal{F}$ as follows: the features come from distribution $\mathbb{P}_{\mathcal{X}}$ on $\mathcal{X}$ and the labels are distributed according to the measurable conditional probability function $\eta : \mathcal{X} \to [0, 1]$. That is, a sample (X, Y) is drawn from $\mathcal{F}$ as follows: X is drawn according to $\mathbb{P}_{\mathcal{X}}$ and Y is chosen according to $\mathbb{P}(Y = 1|X = x) = \eta(x)$.

The goal will be to show that given corrupted examples, the k-NN disagreement method is still able to identify the examples whose labels do not match the Bayes-optimal prediction.

We will make a few regularity assumptions for our analysis to hold. The first regularity assumption ensures that the support $\mathcal{X}$ does not become arbitrarily thin anywhere. This is a standard non-parametric assumption (e.g. Singh et al. (2009); Jiang (2019)).

Assumption 1 (Support Regularity)**.** *There exists $\omega > 0$ and $r_0 > 0$ such that Vol$(\mathcal{X} \cap B(x, r)) \geq \omega \cdot$ Vol$(B(x, r))$ for all $x \in \mathcal{X}$ and $0 < r < r_0$, where $B(x, r) := \{x' \in \mathcal{X} : |x - x'| \leq r\}$.*

Let $p_{\mathcal{X}}$ be the density function corresponding to $\mathbb{P}_{\mathcal{X}}$. The next assumption ensures that with a sufficiently large sample, we will obtain a good covering of the input space.

Assumption 2 ($p_{\mathcal{X}}$ bounded from below)**.** $p_{X,0} := \inf_{x \in \mathcal{X}} p_X(x) > 0$.

Finally, we make a smoothness assumption on η, as done in other analyses of k-NN classification (e.g. Chaudhuri & Dasgupta (2014); Reeve & Kaban (2019))

Assumption 3 (η Hölder continuous)**.** *There exists $0 < \alpha \leq 1$ and $C_\alpha > 0$ such that $|\eta(x) - \eta(x')| \leq C_\alpha |x - x'|^\alpha$ for all $x, x' \in \mathcal{X}$.*

We propose a notion of how spread out a set of points is based on the minimum pairwise distance between the points. This will be a quantity in the finite-sample bounds we will present. Intuitively, the more spread out a contaminated set of points is, the fewer clean samples are needed to overcome the contamination of that set.

Definition 2 (Minimum pairwise distance)**.**

$$S_2(C) := \min_{x, x' \in C, x \neq x'} |x - x'|.$$

Also define the Δ-interior region of $\mathcal{X}$ where there is at least Δ margin in the probabilistic label:

Definition 3. *Let $\Delta \geq 0$. Define $\mathcal{X}^\Delta := \{x \in \mathcal{X} : \left|\frac{1}{2} - \eta(x)\right| \geq \Delta\}$.*

We now state the result, which says that with high probability *uniformly* on $\mathcal{X}^\Delta$ when $\Delta > 0$ is known, we have that the label disagrees with the k-NN classifier if and only if the label is not the Bayes-optimal prediction. Due to space, all of the proofs have been deferred to the Appendix.

Theorem 1 (Fixed Δ)**.** *Let $\Delta, \delta > 0$ and suppose Assumptions 1, 2, and 3 hold. There exists constants $K_l, K_u > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from the $\mathcal{F}$ and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range:*

$$k \geq K_l \cdot \frac{1}{\Delta^2} \cdot \log^2(1/\delta) \cdot \log n,$$
$$k \leq K_u \cdot \min\{S_2(C)^D, \Delta^{D/\alpha}\} \cdot n.$$

Then the following holds uniformly over $x \in \mathcal{X}^\Delta$: the k-NN prediction computed w.r.t. X agrees with the label if and only if the label is the Bayes-optimal label $\eta^*(x) := 1[\eta(x) \geq \frac{1}{2}]$.

In the last result, we assumed that Δ was fixed. We next show how we can make a similar guarantee but show that we can take $\Delta \to 0$ as we choose $k, n \to \infty$ appropriately and provide rates of convergence.

Theorem 2 (Rates of convergence for Δ)**.** *Let $\delta > 0$ and suppose Assumptions 1, 2, and 3 hold. There exist constants $K_l, K_u, K > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from $\mathcal{F}$, and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range:*

$$K_l \cdot \log^2(1/\delta) \cdot n^{\frac{\alpha}{\alpha+D}} \leq k \leq K_u \cdot S_2(C)^D \cdot n,$$

then the following holds uniformly over $x \in \mathcal{X}^\Delta$: the k-NN prediction computed w.r.t. X agrees with the label if and only if the label is the Bayes-optimal label $\eta^*(x) := 1[\eta(x) \geq \frac{1}{2}]$ where

$$\Delta = K \cdot \left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n}\right)^{\alpha/D} \right).$$

Remark 1. *Choosing $k = O(n^{2\alpha/(2\alpha+D)})$ in the above result gives us $\Delta = \widetilde{O}(n^{-\alpha/(2\alpha+D)})$. This rate for Δ is the minimax-optimal rate for k-nearest neighbor classification on $\mathcal{X}^\Delta$ given a sample of size n (Chaudhuri & Dasgupta, 2014) in the uncorrupted setting. Thus, our analysis is tight up to logarithmic factors.*

We next give results with an additional margin assumption, also known as Tsybakov's noise condition (Mammen & Tsybakov, 1999; Tsybakov et al., 2004):

Assumption 4 (Tsybakov Noise Condition). *The following holds for some C_β and β and all $\Delta > 0$:*

$$\mathbb{P}_{\mathcal{X}}(x \notin \mathcal{X}^\Delta) \le C_\beta \cdot \Delta^\beta.$$

Theorem 3 (Rates under Tsybakov Noise Condition). *Let $\delta > 0$ and suppose Assumptions 1, 2, 3 and 4 hold. There exists constants $K_l, K_u, K, K' > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from the $\mathcal{F}$ and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range*

$$K_l \cdot \log^2(1/\delta) \cdot n^{\frac{\alpha}{\alpha+D}} \le k \le K_u \cdot S_2(C)^D \cdot n.$$

Then,

$$\begin{aligned} &\mathbb{P}\left(\eta_k(x) \neq \eta^*(x)\right) \le K \cdot \lambda^\beta, \textit{and} \\ &R_X - R^* \le K' \cdot \lambda^{\beta+1}, \textit{where} \\ &\lambda = \left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n}\right)^{\alpha/D}\right), \end{aligned}$$

$R_X := \mathbb{E}_{\mathcal{F}}[g_k(x) \neq y]$ and $R^* := \mathbb{E}_{\mathcal{F}}[g^*(x) \neq y]$ denote the risk of the k-NN method and Bayes optimal classifier, respectively.

Remark 2. *Choosing $k = O(n^{2\alpha/(2\alpha+D)})$ in the above gives us a rate of $\widetilde{O}(n^{-\alpha(\beta+1)/(2\alpha+D)})$ for the excess risk. This matches the lower bounds of (Audibert et al., 2007) up to logarithmic factors.*

4.1. Impact of Minimum Pairwise Distance

The minimum pairwise distance across corrupted samples, $S_2(C)$, is a key quantity in the theory presented in the previous section. We now empirically study its significance in a simulated binary classification task in 2 dimensions. Clean samples with label L are generated by sampling i.i.d from $\mathcal{N}(\mu_L, I_{2\times 2})$, where $\mu_0 = (0, -2)$ and $\mu_1 = (0, 2)$. The decision boundary is the line $y = 0$. We take 100 samples uniformly spaced on a square grid centered about $(0, 0)$ and corrupt them by flipping their true label. With this construction, $S_2(C)$ is precisely the grid width, which we let vary. The training set is a union of 100 clean samples and the 100 corrupted samples. Using 1000 clean samples as a test set we study the classification performance of a majority vote k-NN classifier, where $k = 10$. Results are shown in Figure 1. As expected, we see that as $S_2(C)$ decreases, so does test accuracy and we need more clean training samples to compensate.

5. Experiments With Clean Auxiliary Data

In this section we present experiments where a small set of relatively clean labeled data is given as side information. The methods in Section 6 do not assume such clean set is available.

5.1. Experiment Set-up

We run experiments for different sizes of $\mathcal{D}_{\text{clean}}$, different noise rates, and different label corruptions, as detailed. We randomly partition each dataset's train set into $\mathcal{D}_{\text{noisy}}$ and $\mathcal{D}_{\text{clean}}$, and we present results for 95/5, 90/10, and 80/20 splits. We corrupt the labels in $\mathcal{D}_{\text{noisy}}$ for a fraction of the examples - the experiment's "noise rate" - using one of three schemes:

Uniform: The label is flipped to any one of the labels (including itself) with equal probability.

Flip: The label is flipped to any *other* label with equal probability.

Hard Flip: With probability $1/2$, we flip the label m to $\pi(m)$ where π is some predefined permutation of the labels. The motivation here is to simulate confusion between semantically similar classes, as done in Hendrycks et al. (2018).

5.2. Comparison Methods

We compare against the following:

Gold Loss Correction (GLC): Hendrycks et al. (2018) estimates a corruption matrix by averaging the softmax outputs of the clean examples on a model trained on noisy data.

Forward: Patrini et al. (2017), similar in spirit to GLC, estimates the corruption matrix by training a model on noisy data and using the softmax output for prototype examples for each class. It does not require a clean dataset like other methods.

Distill: Li et al. (2017) assigns each example in the combined dataset a "soft" label that is a convex combination of its label and its softmax output from a model trained solely on clean data.

Clean: Train only on $\mathcal{D}_{\text{clean}}$.

Full: Train on $\mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{noisy}}$.

k-NN Classify: Train on $\mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{noisy}}$, but at runtime classify using k-NN evaluated on the logit layer (rather than the model decision). This is not a competing data cleaning method, but rather it double-checks the value of the logit layer as a metric space for k-NN.

5.3. Other Experiment Details

We report test errors and show the average across multiple runs with standard error bands shaded. Errors are computed on 11 uniformly distributed noise rates between 0 and 1

Uniform								
Dataset	% Clean	Forward	Clean	Distill	GLC	k-NN	k-NN Classify	Full
Letters	5	4.55	3.16	2.48	2.33	**2.05**	2.19	2.61
	10	4.28	2.51	2.05	1.91	**1.78**	1.79	2.23
	20	3.77	2.06	1.76	1.57	1.56	**1.34**	1.85
Phonemes	5	7.89	1.91	1.79	2.12	**1.26**	2.58	3
	10	7.86	1.54	1.53	1.67	**1.16**	2.28	2.85
	20	7.72	1.34	1.33	1.35	**1.13**	1.76	2.24
Wilt	5	5.18	0.56	0.85	0.54	**0.39**	0.93	1.86
	10	4.68	0.43	0.75	0.45	**0.32**	0.77	1.77
	20	4.31	0.36	0.86	0.35	**0.32**	0.57	1.5
Seeds	5	3.29	4.22	3.71	4.2	3.08	2.87	**2.71**
	10	3.43	3.04	2.84	2.99	**2.14**	2.74	2.57
	20	2.99	2.69	2.27	2.74	**1.72**	2.44	2.2
Iris	5	2.97	3.23	3.48	3.97	2.46	**1.72**	2.05
	10	2.89	2.48	2.59	2.25	1.32	**1.26**	1.55
	20	2.46	1.85	1.97	1.52	**0.6**	0.96	1.32
Parkinsons	5	5	3.46	3.26	4.22	3.4	**3.26**	3.76
	10	5.35	3.26	3.22	3.45	**3.21**	3.22	3.82
	20	4.88	3.01	3.08	3.1	2.98	**2.97**	3.52
MNIST	5	2.88	0.69	1.03	0.5	**0.4**	2.72	2.75
	10	2.57	0.5	0.85	0.41	**0.33**	2.42	2.45
	20	2.07	0.35	0.69	0.34	**0.27**	1.97	2.03
Fashion MNIST	5	2.76	1.88	1.73	1.59	**1.56**	2.53	2.54
	10	2.47	1.71	1.6	1.52	**1.52**	2.21	2.3
	20	2.07	1.56	1.48	1.45	**1.44**	1.95	2.05
CIFAR10	5	6.74	7	6.86	5.43	**5.03**	6.34	6.74
	10	6.58	6.58	6.32	5.39	**5.27**	6.11	6.55
	20	6.4	5.52	5.66	5.11	**4.57**	5.93	6.36
CIFAR100	5	10.8	10.22	9.98	9.59	9.57	**9.17**	9.29
	10	10.79	9.94	9.7	9.42	9.63	**9.09**	9.25
	20	10.78	9.38	9.15	9.06	9.23	**8.92**	9.07
SVHN	5	5.04	3.52	3.56	1.99	**1.62**	4.64	4.95
	10	4.98	2.2	3.2	2.27	**1.34**	4.51	4.82
	20	4.32	1.83	2.67	2.14	**1.2**	3.96	4.41

Table 1. **Area under the test error vs noise rate curve**. Each row corresponds to a dataset and size of $\mathcal{D}_{\text{clean}}$ pair, where the size is a percentage of the total training set (5%, 10%, 20%). Here we show results for Uniform noise; Flip and Hard Flip are in the Appendix. The k-NN method consistently outperforms the other methods under Uniform and Flip and outperforms the others under Hard Flip on the smaller datasets.

inclusive. For the results shown in the main text, we have that $\mathcal{D}_{\text{clean}}$ is randomly selected and is 5% of the data. In the Appendix, we show results over different sizes of $\mathcal{D}_{\text{clean}}$. We implement all methods using Tensorflow 2.0 and Scikit-Learn. We use the Adam optimizer (Kingma & Ba, 2014) with default learning rate 0.001 and a batch size of 128 across all experiments. We use $k = 500$ for all datasets except the UCI where we use $k = 50$ because some of the datasets have fewer than 500 samples. However, we find that the k-NN method's performance was quite stable to the choice of k, which we show in Section 5.7. We describe the permutations used for hard flipping in the Appendix.

5.4. UCI and MNIST Results

We show the results for one of the UCI datasets and Fashion MNIST in Figure 2. Due to space, results for MNIST and the remaining UCI datasets are in the Appendix. For UCI, we use a fully-connected neural network with a single hidden layer of dimension 100 with ReLU activations and we train for 100 epochs. For both MNIST datasets, we use a two hidden-layer fully-connected neural network where each layer has 256 hidden units with ReLU activations. We train the model for 20 epochs. The k-NN approach attains models with a low error rate across noise rates and either outperforms or is competitive with the next best method, GLC.

5.5. CIFAR Results

For CIFAR10/100 we use ResNet-20, which we train from scratch on single NVIDIA P100 GPUs. We train CIFAR10 for 100 epochs and CIFAR100 for 150 epochs. We show results for CIFAR10 in Figure 3 and results for CIFAR100 in the Appendix, due to space. We see that the k-NN method performs competitively. It generally outperforms on the Uniform and Flip noise types but performs worse for the Hard Flip noise type. It is not too surprising that k-NN would be weaker in the presence of Hard Flip noise (i.e. where labels are mapped based on a pre-determined mapping between labels) as the noise is much more structured in that case. This makes filtering by majority vote among neighbors more challenging. In other words, unlike the Uniform and Flip noise types, we are no longer dealing with *white label noise* in the Hard Flip noise type.

5.6. SVHN Results

We show the results in Figure 3. We train ResNet-20 from scratch on the GPUs for 100 epochs. As in the CIFAR experiments, we see that the k-NN method tends to be competitive under Uniform and Flip noise types but does slightly worse under Hard Flip.

5.7. Robustness to k

In this section, we show that our procedure is stable in its hyperparameter k. The theoretical results suggest that a wide range of k can give us statistical consistency guarantees, and in Figure 4 we show that a wide range of k gives us similar results for Algorithm 1. Such robustness in hyperparameter is highly desirable because optimal tuning is hard or impossible, especially when sufficient clean validation set is unavailable.

5.8. Summary of Results

We summarize the results in Table 1 by reporting the area under the test error vs. noise rate curve (on 11 points) for Uniform noise corruption. The best results (bolded) are generally with deep k-NN. Except in one case, when k-NN filtering does not produce the best results, the best results are using the same k-NN to directly classify (k-NN Classify), rather than re-training the deep model. One reason not to use k-NN Classify is its slow runtime. If we insist on re-training the model and only allow the use of k-NN to filter, then k-NN loses 6 of the 33 experiments, with Full (no filtering) being the second best method. Analogous tables for Flip and Hard Flip are in the Appendix and show similar results.

6. Experiments Without Clean Auxiliary Data

Next, we do not split the train set - that is, $\mathcal{D}_{\text{clean}} = \emptyset$. The experimental setup is otherwise as-described in Section 5.

6.1. Additional Comparison Methods

We compare against two more recently-proposed methods that do not use a $\mathcal{D}_{\text{clean}}$.

Bi-Tempered Loss: Amid et al. (2019) introduces two temperatures into the loss function and trains the model on noisy data using this "bi-tempered" loss. We use their code from `github.com/google/bi-tempered-loss` and the hyperparameters suggested in their paper: $t_1 = 0.8, t_2 = 1.2$, and 5 iterations of normalization.

Robust Generative Classifier (RoG): Lee et al. (2019) induces a generative classifier using a model pre-trained on noisy data. We implemented their algorithm mimicking their hyperparameter choices - see Appendix for details.

6.2. Results

Results for MNIST are shown in Figure 5. Due to space, we put results for all the other datasets presented in Section 5 in the Appendix. The results in Figure 5 are mostly representative of the other results. However, the results for CIFAR-100 are notable in that the logit space is 100-dimensional (since it's a 100-class problem), and we hypothesize this higher dimensional space befuddles k-NN a bit as it only does as well as most methods, whereas the RoG method is substantially better than most methods.

7. Conclusions and Open Questions

We conclude from our experiments and theory that the proposed deep k-NN filtering is a safe and dependable method to remove problematic training examples. While k-NN methods can be sensitive to the choice of k when used with small datasets (see e.g. Garcia et al. (2009)), we hypothesize that with today's large datasets one can blithely set k to a fixed medium-sized value (like $k = 500$, as we did) and get reasonable performance. Theoretically, we provided some new results for how well k-NN can identify clean versus corrupted labels. Open theoretical questions are whether there are alternate notions of how to characterize the difficulty of a particular configuration of corrupted examples and whether we can provide both upper and lower learning bounds under these noise conditions.

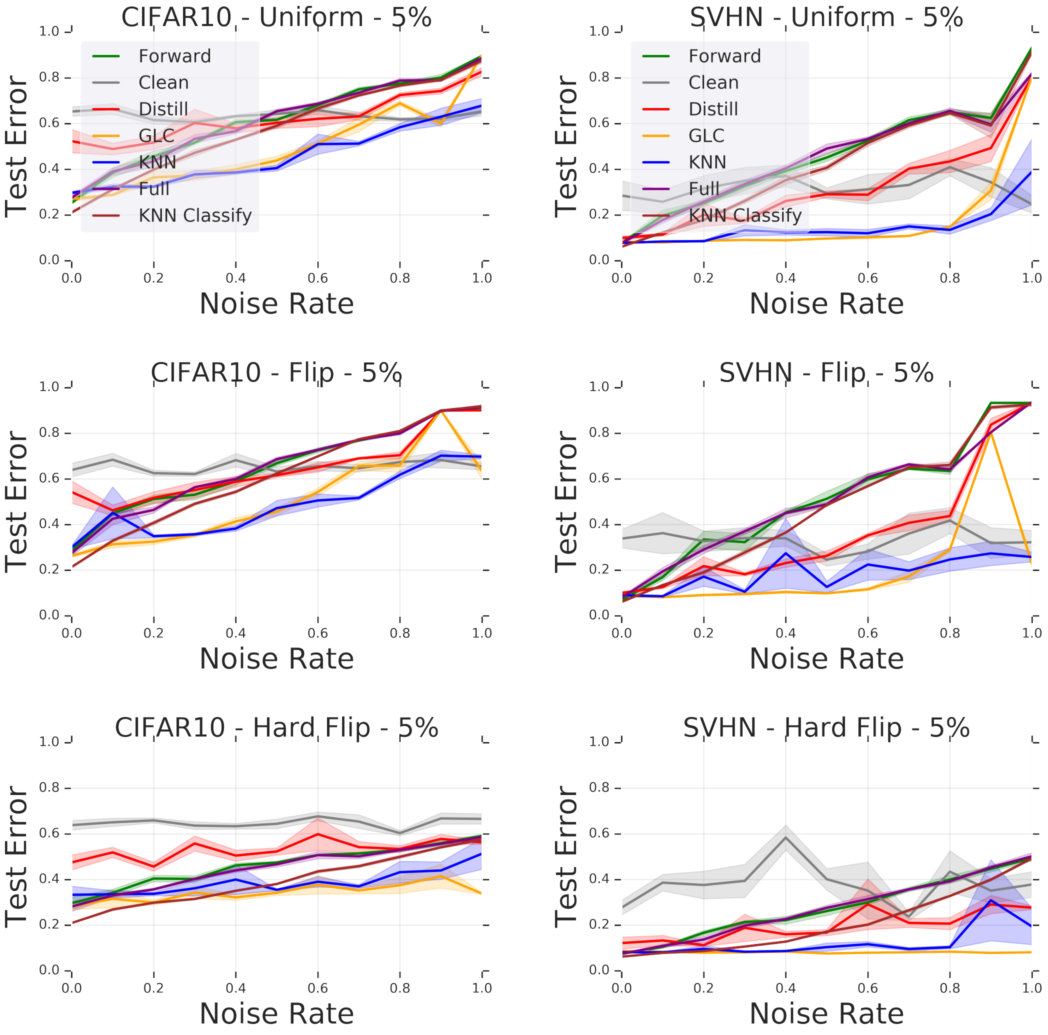

Figure 3. **CIFAR10 and SVHN**: Results under different corruption. We see that our k-NN method performs competitively or outperforms on the Uniform and Flip noise types but performs worse for the Hard Flip noise type. More results are in the Appendix.

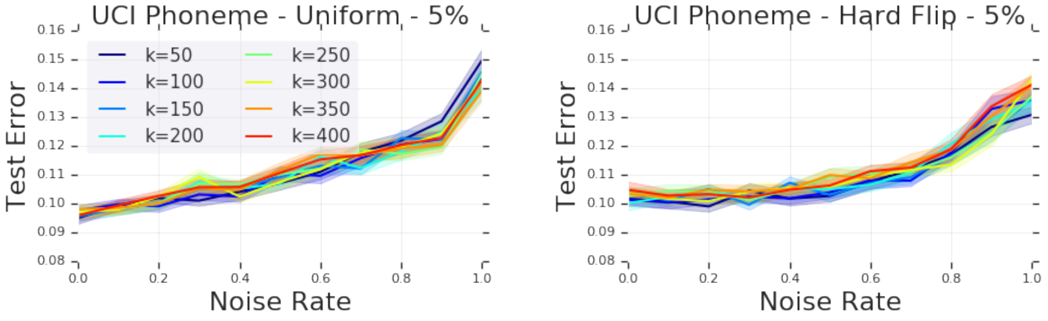

Figure 4. **Performance across different values of** k. We see that the performance of Algorithm 1 is stable to variations in its hyperparameter k for UCI Phoneme under Uniform and Hard Flip noise. The result is similar for Flip noise and can be found in the Appendix.

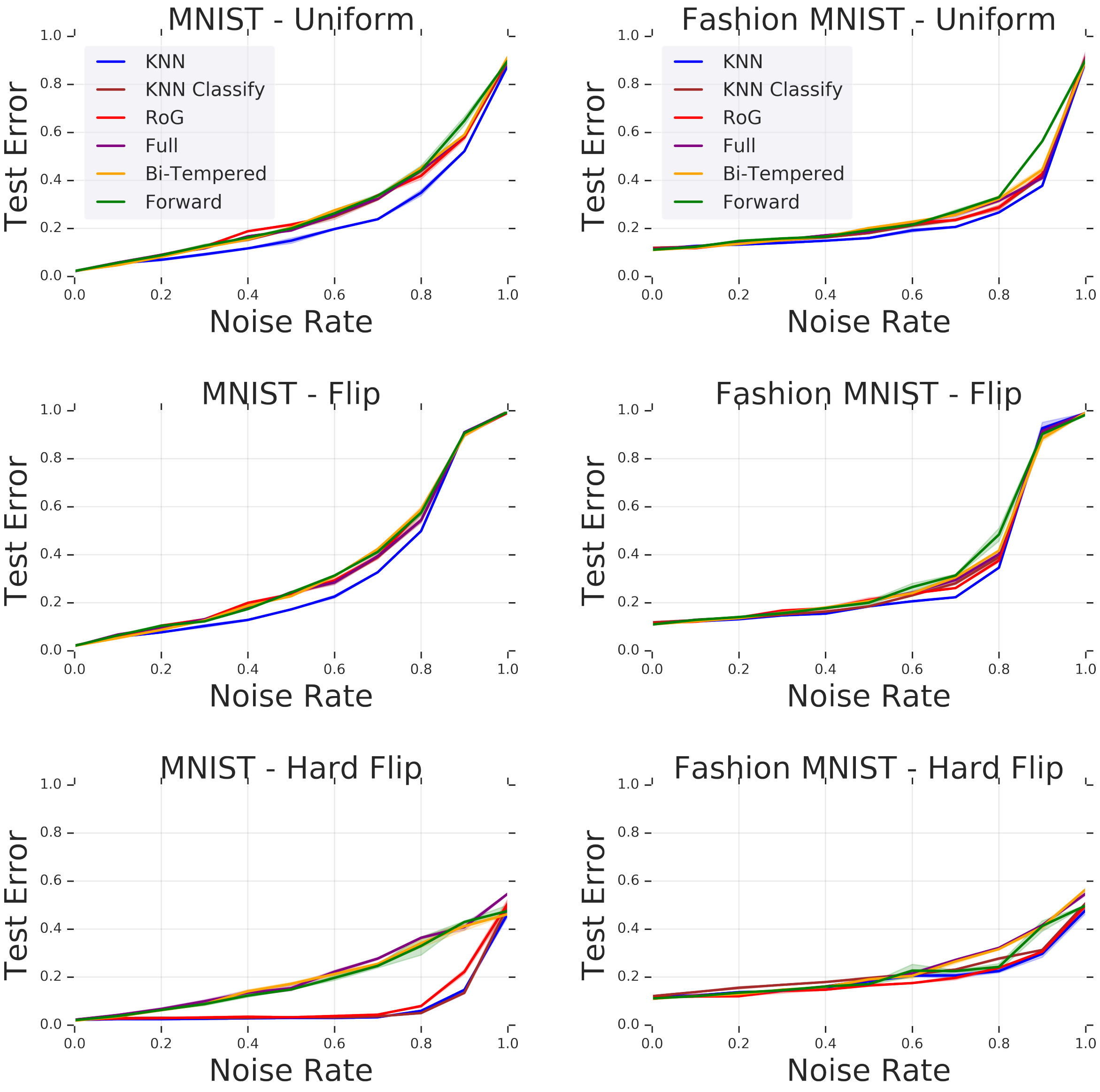

Figure 5. **MNIST and Fashion MNIST with $\mathcal{D}_{\text{clean}} = \emptyset$.** We find that on MNIST, k-NN consistently matches or outperforms the other methods across corruption types and noise levels.

References

Amid, E., Warmuth, M. K., Anil, R., and Koren, T. Robust bi-tempered logistic loss based on Bregman divergences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 14987–14996, 2019.

Audibert, J.-Y., Tsybakov, A. B., et al. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2): 608–633, 2007.

Brodley, C. E. and Freidl, M. A. Identifying mislabeled training data. *Journal Artificial Intelligence Research*, 1999.

Brooks, J. P. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2): 467–479, 2011.

Chaudhuri, K. and Dasgupta, S. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3437–3445, 2014.

Chu, X., Ilyas, I. F., Krishnan, S., and Wan, J. Data cleaning: Overview and emerging challenges. In *SIGMOD*, 2016.

Cover, T. M. Rates of convergence for nearest neighbor procedures. In *Proceedings of the Hawaii International Conference on Systems Sciences*, pp. 413–415, 1968.

Devroye, L., Gyorfi, L., Krzyzak, A., Lugosi, G., et al. On the strong universal consistency of nearest neighbor regression function estimates. *The Annals of Statistics*, 22(3):1371–1385, 1994.

Fix, E. and Hodges Jr, J. L. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley, 1951.

Gao, W., Niu, X.-Y., and Zhou, Z.-H. On the consistency of exact and approximate nearest neighbor with noisy data. *arXiv preprint arXiv:1607.07526*, 2016.

Garcia, E. K., Feldman, S., Gupta, M. R., and Srivastava, S. Completely lazy learning. *IEEE Trans. on Knowledge and Data Engineering*, 2009.

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. 2017.

Guyon, I., Matic, N., and Vapnik, V. Discovering informative patterns and data cleaning. In *AAAI Workshop on Knowledge Discovery in Databases*, 1994.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5836–5846, 2018.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10456–10465, 2018.

Jiang, H. Non-asymptotic uniform rates of consistency for k-NN regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3999–4006, 2019.

Jiang, H., Kim, B., Guan, M. Y., and Gupta, M. R. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Regularizing very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 4, 2017.

Jindal, I., Nokleby, M., and Chen, X. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 967–972. IEEE, 2016.

Khetan, A., Lipton, Z. C., and Anandkumar, A. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lee, K., Yun, S., Lee, K., Lee, H., Li, B., and Shin, J. Robust inference via generative classifiers for handling noisy labels. *arXiv preprint arXiv:1901.11300*, 2019.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1910–1918, 2017.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015.

Mammen, E. and Tsybakov, A. B. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

Masnadi-Shirazi, H. and Vasconcelos, N. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1049–1056, 2009.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1196–1204, 2013.

Papernot, N. and McDaniel, P. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and 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.

Reeve, H. W. and Kaban, A. Fast rates for a kNN classifier robust to unknown asymmetric label noise. *arXiv preprint arXiv:1906.04542*, 2019.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Singh, A., Scott, C., and Nowak, R. Adaptive Hausdorff estimation of density level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.

Stone, C. J. Consistent nonparametric regression. *The Annals of Statistics*, pp. 595–620, 1977.

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

Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., and Mohd-Yusof, J. Combating label noise in deep learning using abstention. In *International Conference on Machine Learning (ICML)*, 2019.

Tsybakov, A. B. et al. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166, 2004.

Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5596–5605, 2017.

Van Rooyen, B., Menon, A., and Williamson, R. C. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10–18, 2015.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and 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.

Wang, Y., Jha, S., and Chaudhuri, K. Analyzing the robustness of nearest neighbors to adversarial examples. In *International Conference on Machine Learning (ICML)*, pp. 5120–5129, 2018.

Wilson, D. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. on Systems, Man and Cybernetics*, 1972.

Xiao, T., Xia, T., Yang, Y., Huang, C., and 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.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8778–8788, 2018.