Some thoughts about loss factorization and centroid estimation

1.using sample reweighting method to get centroid estimation

Denote $\mathcal{D}$ the clean sample distribution, $\tilde{\mathcal{D}}$ the noisy sample distribution, $\tilde{\mathcal{S}}_m \sim \tilde{\mathcal{D}}$ the noisy training set with $\tilde{\mathcal{S}}_m = m$. $(x, y) \sim \mathcal{D}$ where $x \in \mathcal{R}^d$, $y_c \in \{1, \cdots, C\}$ is the label of x and $y \in \triangle^{C-1}$ is the one-hot encoding of y_c in the (c-1)-dimensional simplex. Similarly ,y_i and $\tilde{y}_i$ is the one-hot encoding of $y_{i,c}$ and $\tilde{y}_{i,c}$. I define $Tr(X)$ as the trace of matrix X and e_j the j th column of an identity matirx $I_{C\times C}$. Then the expected loss on clean distribution $\mathcal{D}$ is:

$$\begin{aligned}
\mathcal{R}(\mathcal{D}; W) &= \mathbb{E}_{(x,y)\sim\mathcal{D}}[||W^T x - y||_2^2] \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}}[1 + x^T W W^T x - 2 \cdot x^T W y] \\
&= 1 + \mathbb{E}_x[x^T W W^T x] - 2 \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}}[x^T W y] \\
&= 1 + \mathbb{E}_x[x^T W W^T x] - 2 \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathrm{Tr}(y^T W^T x)] \\
&= 1 + \mathbb{E}_x[x^T W W^T x] - 2 \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathrm{Tr}(W^T x y^T)] \\
&\overset{(1)}{=} 1 + \mathbb{E}_x[x^T W W^T x] - 2 \cdot \mathrm{Tr}[W^T \mathbb{E}_{(x,y)\sim\mathcal{D}}(x y^T)]
\end{aligned}$$

Where (1) holds due to $\mathbb{E}(\mathrm{Tr(X)}) = \mathrm{Tr(E(X))}$ and $\mathbb{E}_x(Ax) = A\mathbb{E}_x(x)$, which is easy to prove.

I denote $\mu(\mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}(xy^T)$ as the centroid of clean data distribution and $\mu(\tilde{\mathcal{D}}) = \mathbb{E}_{(x,\tilde{y})\sim\tilde{\mathcal{D}}}(x\tilde{y}^T)$ as the centroid of noisy distribution $\tilde{\mathcal{D}}$. Then we get:

$$\begin{aligned}
\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy^T] &= \int_{\mathcal{X}} \sum_{i=1}^{C} P_{\mathcal{D}}(X = x, Y = e_i) x e_i^T dX \\
&= \int_{\mathcal{X}} \sum_{i=1}^{C} P_{\tilde{\mathcal{D}}}(X = x, \tilde{Y} = e_i) \frac{P_{\mathcal{D}}(X = x, Y = e_i)}{P_{\tilde{\mathcal{D}}}(X = x, \tilde{Y} = e_i)} x e_i^T dX \\
&= \int_{\mathcal{X}} \sum_{i=1}^{C} P_{\tilde{\mathcal{D}}}(X = x, \tilde{Y} = e_i) \frac{P_{\mathcal{D}}(Y = e_i | X = x)}{P_{\tilde{\mathcal{D}}}(\tilde{Y} = e_i | X = x)} x e_i^T dX \\
&= \mathbb{E}_{(x,\tilde{y})\sim\tilde{\mathcal{D}}}[\frac{P_{\mathcal{D}}(Y|X = x)_{\tilde{y}_c}}{P_{\tilde{\mathcal{D}}}(\tilde{Y}|X = x)_{\tilde{y}_c}} x\tilde{y}^T]
\end{aligned}$$

Assume that we already have the well-estimated transition matrix $T(X) = T$ where $T_{ij} = P(\tilde{y} = e_j | y = e_i)$.Then $P_{\tilde{\mathcal{D}}}(\tilde{Y} = e_i | X = x) = \sum_{j=1}^{C} P_{\tilde{\mathcal{D}}}(\tilde{Y} = e_i, y = e_j | X = x)$=$\sum_{j=1}^{C} P_{\mathcal{D}}(y =$

$e_j|X=x)P(\tilde{y}=e_i|y=e_j)$=$(T^T p(y|x))_i$, where α_i means the i th element of a vector α and $p(y|x)\in\mathcal{R}^C$ is the probability distribution of ground-true label y.Denote $p(y|x)\in\mathcal{R}^C$ the noisy label distribution, then we get:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy^T]=\mathbb{E}_{(x,\tilde{y})\sim\tilde{\mathcal{D}}}\Big[\frac{p(y|x)_{\tilde{y}_c}}{(T^T p(y|x))_{\tilde{y}_c}}x\tilde{y}^T\Big]$$

So we get the following result:

$$\mathcal{R}(\mathcal{D};W)=1+\mathbb{E}_x[x^TWW^Tx]-2\cdot\mathrm{Tr}[W^T\mathbb{E}_{(x,\tilde{y})\sim\tilde{\mathcal{D}}}\Big[\frac{p(y|x)_{\tilde{y}_c}}{(T^T p(y|x))_{\tilde{y}_c}}x\tilde{y}^T\Big]]$$

Note that the second term is **label independent**, and the third term depends only on noisy label, so we can approximate the second and third term by empirical value:

$$\begin{aligned}\mathcal{R}_1(X;W)&=\frac{1}{n}\sum_{i=1}^{n}||W^Tx_i||_2^2\\ \mathcal{R}_2(X,\tilde{Y};W)&=\mathrm{Tr}[W^T\frac{1}{n}\sum_{i=1}^{n}\Big[\frac{p(y|x_i)_{\tilde{y}_{i,c}}}{(T^T p(y|x_i))_{\tilde{y}_{i,c}}}x_i\tilde{y}_i^T\Big]]\\ &=\mathrm{Tr}[\frac{1}{n}\sum_{i=1}^{n}\Big[\frac{p(y|x_i)_{\tilde{y}_{i,c}}}{(T^T p(y|x_i))_{\tilde{y}_{i,c}}}W^Tx_i\tilde{y}_i^T\Big]]\end{aligned}$$

By neural network, we can get $p(y|x)$ and learn network parameters by the following objective, which is composed of noisy label dependent term and noisy label independent term.

$$W^*=\arg\min_W\mathcal{R}(\mathcal{D};W)=\mathcal{R}_1(X;W)+\mathcal{R}_2(X,\tilde{Y};W)$$

2.using conditional expectation to get (clean)label-independent loss

First, we have expected loss:

$$\begin{aligned}\mathcal{R}(\mathcal{D};W)&=\mathbb{E}_{(x,y)\sim\mathcal{D}}[||W^Tx-y||_2^2]\\ &=\mathbb{E}_x\mathbb{E}_{y|x}||W^Tx-y||_2^2\end{aligned}$$

The conditional expectation is:

$$
\begin{aligned}
\mathbb{E}_{y|x}||W^Tx-y||_2^2 &= \mathbb{E}_{y|x}[1+x^TWW^Tx-2\cdot x^TWy] \\
&= 1+x^TWW^Tx-2\cdot\sum_{i=1}^{C}P_{\mathcal{D}}(y=e_i|X=x)x^TWe_i \\
&= 1+x^TWW^Tx-2\cdot x^TW\sum_{i=1}^{C}P_{\mathcal{D}}(y=e_i|X=x)e_i \\
&= 1+x^TWW^Tx-2\cdot x^TWP_{\mathcal{D}}(Y|X=x)
\end{aligned}
$$

Where $P_{\mathcal{D}}(Y|X=x)$ is the clean label distribution. As $T^TP_{\mathcal{D}}(Y|X=x)=P_{\tilde{\mathcal{D}}}(\tilde{Y}|X=x)$, we can reweight the conditional expectation as :

$$
\mathbb{E}_{y|x}||W^Tx-y||_2^2 = 1+x^TWWx-2\cdot x^TWT^{-T}P_{\tilde{\mathcal{D}}}(\tilde{Y}|X=x)
$$

So the final expected loss based on conditional expectation is:

$$
\begin{aligned}
\mathcal{R}(\mathcal{D};W) &= \mathbb{E}_x\mathbb{E}_{y|x}||W^Tx-y||_2^2 \\
&= 1+\mathbb{E}_x||W^Tx||_2^2-2\mathbb{E}_x(x^TWT^{-T}P_{\tilde{\mathcal{D}}}(\tilde{Y}|X=x))
\end{aligned}
$$

Then we can also get the optimal parameters by empirical loss minimization as we did in **1**:

$$
W^* = \arg\min_W \mathcal{R}(\mathcal{D};W)
$$

3.extend loss factorization & centroid estimation to openset setting and give the model ability to reject during training and testing phase

Assume true label $y\in\mathcal{R}^{C+1}$ where C classes are observed and the last element of y is an Out-of-Distribution indicator. The observed noisy label $\tilde{y}\in\mathcal{R}^C$ contain only C observed classes. We aim to train a neural network to get the clean label distribution $P(Y|X)\in\mathcal{R}^{C+1}$, and if $\arg\max_Y P(Y|X)=C+1$, then sample $(X,\tilde{Y})\sim\tilde{\mathcal{D}}$ is an Out-of-Distribution sample. We denote $\pi_i=P(y=e_i)$.

Similar to [1], we have:

$$
\mathbb{E}_{(x,\tilde{y})}[X\tilde{Y}^T|(X,Y)] = \sum_{i=1}^{C+1}P(Y=e_i)\mathbb{E}_{\tilde{Y}}[X\tilde{Y}^T|(X,Y=e_i)]
$$

If $Y = e_i \in \mathcal{R}^{C+1}$ and $\tilde{Y} = e_j \in \mathcal{R}^C$, we can get the following relation between Y and $\tilde{Y}$:

$$\tilde{Y} = \begin{bmatrix} I_C \\ 0^T \end{bmatrix}^T E_{ij} Y$$

where I_C is an identity matrix, E_{ij} is a permutation matrix to exchange i th row and j th row of Y with $E_{ij}^T = E_{ij}$. $0^T \in \mathcal{R}^C$ is a row zero-vector. Then we denote $S = \begin{bmatrix} I_C \\ 0^T \end{bmatrix}$, due to its column-orthogonal property, we can obtain $S^\dagger = S^T = \begin{bmatrix} I_C & 0 \end{bmatrix}$, which is useful for the next derivation.

We can easy derive the following result:

$$\begin{aligned} \mathbb{E}_{\tilde{Y}}[X\tilde{Y}^T|(X,Y)] &= \sum_{i=1}^{C+1} P(Y=e_i) \sum_{j=1}^{C} T_{ij} XY^T E_{ij} S \\ &= \sum_{i=1}^{C+1} \pi_i \sum_{j=1}^{C} T_{ij} XY^T E_{ij} S \\ &= XY^T (\sum_{i=1}^{C+1} \pi_i \sum_{j=1}^{C} T_{ij} E_{ij}) S \end{aligned}$$

Similar to [1], we denote $M = \sum_{i=1}^{C+1} \pi_i \sum_{j=1}^{C} T_{ij} E_{ij}$, as $\mathbb{E}_{(X,Y)}\left[\mathbb{E}_{\tilde{y}}[X\tilde{Y}^T|(X,Y)]\right] = \mathbb{E}_{(X,\tilde{Y})}\left[X\tilde{Y}^T\right] = \mathbb{E}_{(X,Y)}\left[XY^T\right]MS$, so $\mathbb{E}_{(X,Y)}\left[XY^T\right] = \mathbb{E}_{(X,\tilde{Y})}\left[X\tilde{Y}^T\right]S^T M^\dagger$, which is because $(AB)^\dagger = B^\dagger A^\dagger$ and $S^\dagger = S^T$.

Finally, we can get the optimal parameters of neural network by empirical risk minimization on the mean square loss:

$$W^* = \arg\min_W \hat{R}(\mathcal{D}; W) = 1 + \frac{1}{n} \sum_{i=1}^{n} x^T W W^T x - 2 \cdot \mathrm{Tr}(W \hat{\mu}(\tilde{\mathcal{D}}) S^T M^\dagger)$$

where $\hat{\mu}(\tilde{\mathcal{D}}) = \frac{1}{n} \sum_{i=1}^{n} X_i \tilde{Y}_i^T$ is the empirical centroid of noisy data distribution $\tilde{\mathcal{D}}$.

The transition matrix $T \in \mathcal{R}^{(C+1)\times C}$ can estimated using the similar method like[3].

4.rethinking noisy centroid and get distribution-reletive centroid estimation

Now we think deep into the derivation of 3. We can find that we use $\hat{\mu}(\tilde{\mathcal{D}}) = \frac{1}{n}\sum_{i=1}^{n} X_i \tilde{Y_i}^T$ to estimite the empirical centroid of noisy distribution. But now we can give noisy data a distribution so these training samples get different weights.

To estimate $\mu(\tilde{\mathcal{D}})$, we must get joint distribution of noisy data, i.e. $P(x, \tilde{y})$, which can get with noisy training.Assuming for class $i \in \{1, 2, \ldots, C\}$, there are m noisy prototypes $\{m_j^i\}_{j=1}^m$ and these prototypes divide up the sample space that belongs to class i.
Then we can derive the following equations:

$$\begin{aligned}\hat{P}(X = x_i, \tilde{Y} = \tilde{y}_i) &= \sum_{j=1}^{m} P(x_i, x_i \in m_j^{\tilde{y}_{i,c}}, \tilde{Y} = \tilde{y}_i) \\ &= \sum_{j=1}^{m} P(\tilde{Y} = \tilde{y}_i) P(x_i \in m_j^{\tilde{y}_{i,c}} | \tilde{Y} = \tilde{y}_i) P(x_i | \tilde{Y} = \tilde{y}_i, x_i \in m_j^{\tilde{y}_{i,c}}) \\ &= \tilde{\pi}_{\tilde{y}_{i,c}} \sum_j \mathrm{Softmax}(\frac{x_i^T m^{\tilde{y}_{i,c}}}{\sqrt{d}})_j \cdot C \exp\left(\frac{-||x_i - m_j^{\tilde{y}_{i,c}}||^2}{2\sigma^2}\right)\end{aligned}$$

Where $\mathrm{Softmax}(\frac{x_i^T m^{\tilde{y}_i}}{\sqrt{d}})$ is an attention vector[2], which gives different attention to these prototypes and d is the dimention of embedding space/feature space. C is a constant, which is useless for our analysis. Assume that $x_i \in \mathcal{B}^d$ and $m_j^{\tilde{y}_i} \in \mathcal{B}^d$ where $\mathcal{B}^d$ is a d-dimentional ball, so we can derive:

$$\hat{P}(X = x_i, \tilde{Y} = \tilde{y}_i) = \tilde{\pi}_{\tilde{y}_{i,c}} \sum_j \mathrm{Softmax}(\frac{x_i^T m^{\tilde{y}_{i,c}}}{\sqrt{d}}) \cdot C \exp\left(\frac{x_i^T m_j^{\tilde{y}_{i,c}}}{\tau}\right)$$

where $\tau = \sigma^2$ is a hyperparameter.Then normalize $\hat{P}(X = x_i, \tilde{Y} = e_{\tilde{y}_i}), \forall (x_i, \tilde{y}_i) \in \tilde{D}$ can get the final per sample weight:

$$P(X = x_i, \tilde{Y} = \tilde{y}_i) = \frac{\hat{P}(X = x_i, \tilde{Y} = \tilde{y}_i)}{\sum_j \hat{P}(X = x_j, \tilde{Y} = \tilde{y}_j)}$$

So we can estimate the noisy centroid by:

$$\mu(\tilde{\mathcal{D}}) = \sum_i P(X = x_i, \tilde{Y} = \tilde{y}_i) X_i \tilde{Y}_i^T$$

[1]Multi-class Label Noise Learning via Loss Decomposition and Centroid Estimation

[2]PMAL:Open Set Recognition via Robust Prototype Mining

[3]Provably End-to-end Label Noise Learning without Anchor Points