Supplementary Materials to "DAT:Training Deep Networks Robust to Label-Noise by Matching the Feature Distributions"

Anonymous CVPR 2021 submission

Appendix A
Code

The algorithmic description of DAT without clean set is shown in Algorithm 1. To illustrate how DAT works, we also provide the code on the MNIST and CIFAR-10 datasets. The provided code is in the DAT-master folder, and the github url will be released after the review procedure.

Algorithm 1 DAT-Algorithm without clean set

Input: noisy training set D_ρ, α and β, learning rate η, epoch T, iteration N.

```
for t = 1, 2, 3, ..., T do
    Shuffle training set D_ρ
    Sample a subset D_s from D_ρ
    for n = 1, 2, 3, ..., N do
        Fetch mini-batch ρ̄ from D_ρ
        Fetch mini-batch S̄ from D_s
        Calculate L_{cce~} on ρ̄, L_dis on S̄
        Update θ_{h,h̀,g} = θ_{h,h̀,g} − ∇_{θ_{h,h̀,g}} L_{cce~}
        Update θ_h̀ = θ_h̀ + α∇_{θ_h̀} L_dis
        Update θ_g = θ_g − β∇_{θ_g} L_dis
```

1. **for** $t = 1, 2, 3, \ldots, T$ **do**
    - Shuffle training set D_ρ
    - Sample a subset D_s from D_ρ
    - **for** $n = 1, 2, 3, \ldots, N$ **do**
        - Fetch mini-batch $\bar{\rho}$ from D_ρ
        - Fetch mini-batch $\bar{S}$ from D_s
        - Calculate $\mathcal{L}_{\widetilde{cce}}$ on $\bar{\rho}$, $\mathcal{L}_{dis}$ on $\bar{S}$
        - Update $\theta_{h,\grave{h},g} = \theta_{h,\grave{h},g} - \nabla_{\theta_{h,\grave{h},g}} \mathcal{L}_{\widetilde{cce}}$
        - Update $\theta_{\grave{h}} = \theta_{\grave{h}} + \alpha \nabla_{\theta_{\grave{h}}} \mathcal{L}_{dis}$
        - Update $\theta_g = \theta_g - \beta \nabla_{\theta_g} \mathcal{L}_{dis}$

Output: $\theta_{h,\grave{h},g}$

APPENDIX B
THEORETICAL DERIVATION

In this section, we show the proof of Theorem 1 and the reason that $h\triangle\mathcal{H}$-divergence has a tighter upper bound. For ease of reference, we restate the definition of $h\triangle\mathcal{H}$-divergence and Theorem 1.

Definition 1: Given two feature distribution $D_\rho^{\mathcal{Z}}$ and $D_c^{\mathcal{Z}}$ extracted by a fixed g, and a hypothesis class $\mathcal{H}$ which is a set of binary classifiers. Through a given classifier h, $h\triangle\mathcal{H}$-divergence between $D_\rho^{\mathcal{Z}}$ and $D_c^{\mathcal{Z}}$ is:

$$d_{h\triangle\mathcal{H}}(D_\rho^{\mathcal{Z}}, D_c^{\mathcal{Z}}) = 2\sup_{\grave{h}\in\mathcal{H}}\left\{\Pr_{z\sim \mathrm{D}_c^{\mathcal{Z}}}\left[h(z)\neq \grave{h}(z)\right] - \Pr_{z\sim \mathrm{D}_\rho^{\mathcal{Z}}}\left[h(z)\neq \grave{h}(z)\right]\right\}. \tag{1}$$

The following Theorem 1 can be stated through the $h\triangle\mathcal{H}$-divergence.

Theorem 1: Let g be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$, $\mathcal{H}$ be the hypothesis class of Vapink-Chervonenkis dimension d. If random noisy samples of size m is generated by applying g from D_ρ-i.i.d., then with probability at least $1-\delta$, the generalized bound of the clean risk $\epsilon_c(h)$:

$$\epsilon_c(h) \leq \epsilon_\rho^m(h) + \frac{1}{2}d_{h\triangle\mathcal{H}}\left(D_\rho^{\mathcal{Z}}, D_c^{\mathcal{Z}}\right) + \lambda, \tag{2}$$

where

$$\lambda = \epsilon_c(h^*) + \epsilon_\rho(h^*) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})}, \tag{3}$$

$$h^* = \mathrm{argmin}_{h\in\mathcal{H}}\ \epsilon_c(h), \tag{4}$$

$$\epsilon_\rho^m(h) = \frac{1}{m}\sum_{i=1}^{m}|\hat{f}_\rho(z) - h(z)|. \tag{5}$$

Proof 1: For a classifier h, let $\mathcal{Z}_h \subseteq \mathcal{Z}$ be the characteristic subset for whose characteristic function is h. The parallel notation $\mathcal{Z}_{h^*}$ and $\mathcal{Z}_{\grave{h}}$ are used for classifier h^* and $\grave{h}$. Through the characteristic subset, we make $\Pr_c[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}] = \Pr_{z\sim\mathrm{D}_c^{\mathcal{Z}}}[h(z)\neq h^*(z)]$, and the parallel notation $\Pr_\rho$ is used.

$$\begin{aligned}
\epsilon_c(h) &\leq \epsilon_c(h^*) + \Pr_c[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}] && (6)\\
&\leq \epsilon_c(h^*) + \Pr_\rho[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}] + \{\Pr_c[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}] - \Pr_\rho[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}]\} && (7)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + \{\Pr_c[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}] - \Pr_\rho[\mathcal{Z}_h\triangle\mathcal{Z}_{h^*}]\} && (8)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + \sup_{\grave{h}\in\mathcal{H}}\left\{\Pr_c\left[\mathcal{Z}_h\triangle\mathcal{Z}_{\grave{h}}\right] - \Pr_\rho\left[\mathcal{Z}_h\triangle\mathcal{Z}_{\grave{h}}\right]\right\} && (9)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + \frac{1}{2}d_{h\triangle\mathcal{H}}\left(D_\rho^{\mathcal{Z}}, D_c^{\mathcal{Z}}\right) && (10)
\end{aligned}$$

InEq. (6) and InEq. (8) relies on the triangle inequality for classification error [1]. According to the standard Vapnik-Chervonenkis theory [2], we can then bound the true $\epsilon_\rho(h)$ by its empirical estimate $\epsilon_\rho^m(h)$:

$$\epsilon_\rho(h) \leq \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \epsilon_\rho^m(h) \tag{11}$$

in summary:

$$\epsilon_c(h) \leq \epsilon_\rho^m(h) + \lambda + \frac{1}{2}d_{h\triangle\mathcal{H}}\left(D_\rho^{\mathcal{Z}}, D_T^{\mathcal{Z}}\right) \tag{12}$$

Before explaining why $h\triangle\mathcal{H}$-divergence has a tighter upper bound, we give a definition of $\mathcal{H}\triangle\mathcal{H}$-divergence [3] (the same analysis type is suitable for $\mathcal{H}$-divergence):

Definition 2: Given two feature distribution $D_\rho^{\mathcal{Z}}$ and $D_c^{\mathcal{Z}}$ extracted by a fixed g, and a hypothesis class $\mathcal{H}$ which is a set of binary classifiers. Through a given classifier h, $h\triangle\mathcal{H}$-divergence between $D_\rho^{\mathcal{Z}}$ and $D_c^{\mathcal{Z}}$ is:

$$d_{\mathcal{H}\triangle\mathcal{H}}(D_\rho^{\mathcal{Z}}, D_c^{\mathcal{Z}}) = 2\sup_{h,\grave{h}\in\mathcal{H}}\left|\Pr_{z\sim \mathrm{D}_c^{\mathcal{Z}}}\left[h(z)\neq \grave{h}(z)\right] - \Pr_{z\sim \mathrm{D}_\rho^{\mathcal{Z}}}\left[h(z)\neq \grave{h}(z)\right]\right|. \tag{13}$$

Assuming that the $h\triangle\mathcal{H}$-divergence is replaced by the $\mathcal{H}\triangle\mathcal{H}$-divergence in Theorem 1, the proof becomes of the following form.

Proof 2:

$$
\begin{aligned}
\epsilon_c(h) &\leq \epsilon_c(h^*) + \Pr_c[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}] && (14)\\
&\leq \epsilon_c(h^*) + \Pr_\rho[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}] + |\Pr_c[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}] - \Pr_\rho[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}]| && (15)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + |\Pr_c[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}] - \Pr_\rho[\mathcal{Z}_h \triangle \mathcal{Z}_{h^*}]| && (16)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + \sup_{h,\grave{h}\in\mathcal{H}} \left|\Pr_c\left[\mathcal{Z}_h \triangle \mathcal{Z}_{\grave{h}}\right] - \Pr_\rho\left[\mathcal{Z}_h \triangle \mathcal{Z}_{\grave{h}}\right]\right| && (17)\\
&\leq \epsilon_c(h^*) + \epsilon_\rho(h^*) + \epsilon_\rho(h) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}\left(D_\rho^{\mathcal{Z}}, D_c^{\mathcal{Z}}\right) && (18)
\end{aligned}
$$

Compared to InEq. (7), InEq. (15) add an additional absolute value, which is an absolute value inequality that allows the upper bound of the clean error rate $\epsilon_c(h)$ to be amplified. In addition, InEq. (17) searches both h and $\grave{h}$ in $\mathcal{H}$ to maximize the probability difference, which also amplifies the upper bound of $\epsilon_c(h)$ even more compared to InEq. (9). As a result, $h\triangle\mathcal{H}$-divergence has a tighter generalized upper bound.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, pp. 137–144, 2006.
[2] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.
[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.