Contrastive Coding for Active Learning under Class Distribution Mismatch

1 Appendix

1.1 The propertity of lipschitz continuity

Before starting the proof, we will state that CNN is $\lambda^l - Lipschitz\ continuous$. We have the following definition [4] of *Lipschitz continuous* with high dimensional x.

Definition 1 *A function $f : \mathcal{R}^n \to \mathcal{R}^m$ is called Lipschitz continuous if there exists a constant L such that*

$$\forall x, y \in \mathcal{R}^n, ||f(x) - f(y)||_2 \leq L||x - y||_2.$$

The smallest L for which the previous inequality is true called the Lipschitz constant of f and will be denoted $L(f)$.

Then we can conclude the following two Lemmas.

Lemma 1 *The Softmax function is λ^s-Lipschitz continuous.*

Proof 1 *We can solve the Lipschitz constant of Softmax function by maximizing of the Frobenius norm of the Jacobian matrix. Softmax function is defined as*

$$f_i(x) = \frac{exp(x_i)}{\sum_{j=1}^{C} exp(x_j)} \triangleq f_i \quad i = 1, 2, ..., C.$$

Its Jacobian matrix is

$$J = \begin{bmatrix} f_1(1-f_1) & -f_1 f_2 & ... & -f_1 f_C \\ -f_2 f_1 & f_2(1-f_2) & ... & -f_2 f_C \\ ... & ... & ... & ... \\ -f_C f_1 & -f_C f_2 & ... & -f_C(1-f_C) \end{bmatrix}.$$

And the Frobenius norm will be

$$||J||_{\mathcal{F}} = \sqrt{2\sum_{i=1}^{C}\sum_{j>i}^{C} f_i^2 f_j^2 + \sum_{i=1}^{C} f_i^2(1-f_i)^2}.$$

Now, we need to solve the optimal solution for $||J||_{\mathcal{F}}$*, and we use the Lagrange multiplier method [1]. In our task, the constrains are* $f_1 + f_2 + ... + f_C = 1$*. So, we can get the unconstrained function as,*

$$F(x) = \sqrt{2\sum_{i=1}^{C}\sum_{j>i}^{C} f_i^2 f_j^2 + \sum_{i=1}^{C} f_i^2(1-f_i)^2} + \lambda(1 - f_1 - f_2 - ... - f_C),$$

where λ *is a Lagrange multiplier. Then, we obtain several equalities as follows.*

$$\begin{cases} \sum\limits_{j\neq i}^{C} f_i f_j^2 + 2f_i(1-f_i)(1-2f_i) = 0, \quad \forall i = 1, 2, ..., C \\ f_1 + f_2 + ... + f_C = 1 \end{cases}$$

The solution is $f_i = \frac{1}{C}$ *,*$\forall i = 1, 2, ..., C$*. Hence, we get Lipschitz constant* $L(f) = \frac{\sqrt{C-1}}{C} \triangleq \lambda^s$*.*

Lemma 2 *The fully-connected layer is* λ^c*-Lipschitz continuous.*

Proof 2 *With two inputs* x_i*,* $x_j \in \mathcal{R}^n$*, their output at one fully-connected layer is* y_i*,* $y_j \in \mathcal{R}^n$*. The function* $f : \mathcal{R}^n \to \mathcal{R}^n$ *can be defined as* $f(x) = y = \mathbf{W}x$*. By similar steps in above proof, we get the* $\lambda^c = \tau$*, if assuming* $||\mathbf{W}||_{\mathcal{F}} \leq \tau$*.*

Considering that the max pooling layer and Convolutional layer are special fully-connected layers, both are λ^c*-Lipschitz continuous.*

Lemma 3 *ReLU is* 1*-Lipschitz continuous.*

Proof 3 *ReLU*$(\cdot)$ *is defined as* $max(0, \cdot)$*, and we have*

$$||ReLU(x_i) - ReLU(x_j)||_2 = ||max(0, x_i) - max(0, x_j)||_2 \leq ||x_i - x_j||_2.$$

Thus, ReLU is 1*-Lipschitz continuous.*

Combing Lemma1 &2 &3, we can state that CNN is λ^l*-Lipschitz continuous.*

Lemma 4 *If one Convolutional Neural Networks consists of* n_c *Convolutional layers,* n_p *max pooling layers,* n_r *ReLU and a softmax function, the CNN is* λ^l*-Lipschitz continuous, where* $\lambda^l = \lambda^s\{\lambda^c\}^{(n_c+n_p)}$*.*

Proof 4 *We define the function of d^{th} layer as $h_d(x) = \mathbf{W_d}x$, so the CNN will be*

$$CNN(x) = \mathbf{W_{n_c+n_p+n_r+1}} \cdots \mathbf{W_2 W_1} x.$$

Then, we obtain,

$$||CNN(x_i) - CNN(x_j)||_2 \leq \lambda^s \{\lambda^c\}^{(n_c+n_p)} ||x_i - x_j||_2 \triangleq \lambda^l ||x_i - x_j||_2.$$

Thus, the CNNs are λ^l-Lipschitz continuous.

Because the loss function can be rewritten as,

$$\begin{aligned} &|l(x_i, y_i; \mathbf{w}) - l(x_j, y_j; \mathbf{w})| \\ &\quad = \big| ||CNN(x_i) - y_i||_2 - ||CNN(x_j) - y_j||_2 \big| \\ &\quad \leq ||CNN(x_i) - CNN(x_j)||_2 \\ &\quad \leq \lambda^l ||x_i - x_j||_2. \end{aligned}$$

Hence, we proof that the loss function is also λ^l-Lipschitza continuous. In the theoretical study, we use the l_2 loss instead of the widely applied cross-entropy loss for the classification problem. Our experiments show that CCAL is very effective for cross-entropy loss, although our theoretical study does not extend to it.

1.2 The proof for the upper bound of active learning error

A bound for valid query error

With the propertity of lipschitz continuity, we can further proof an effective upper bound for the valid query error. As we assume that the training error can be reduced to zero, we have

$$\left| \frac{1}{p} \sum_{i=1}^{p-d} l(x_i^{ID\setminus re}, y_i; \mathbf{w}) - \frac{1}{q} \sum_{j=1}^{q} l(x_j^{tr}, y_j; \mathbf{w}) \right| \leq \left| \frac{1}{p} \sum_{i=1}^{p} l(x_i^{ID\_C}, y_i; \mathbf{w}) \right|.$$

Then, we will calculate the upper bound of expectation of above average, denoted as $E_{y\sim\tau(x)}\big[l(x, y; \mathbf{w})\big]$.

Before start, we state a lemma which tells us that if two vectors are close to each other then their labels are likely to be the same.

Lemma 5 *(Ex.3 of Chapter 19 in [3]) Fix some $p, p' \in [0, 1]$ and $y' \in \{0, 1\}$. We use the notation $y \sim p$ as a shorthand for that y is a Bernoulli random variable with expected value p, and denote $p = \frac{1}{k}\sum_i p_i$ and $p' = \frac{1}{k}\sum_{i=1}^{k} CNN_i$. Show that*

$$\mathcal{P}_{y\sim p}[y \neq y'] \leq \mathcal{P}_{y\sim p'}[y \neq y'] + |p - p'|.$$

Proof 5 *If $y' = 0$, we have*

$$\mathcal{P}_{y\sim p}[y \neq y'] = \mathcal{P}_{y\sim p}[y = 1] = p = p' + p - p' \leq \mathcal{P}_{y\sim p'}[y \neq y'] + |p - p'|.$$

And if $y' = 1$, we have

$$\mathcal{P}_{y\sim p}[y \neq y'] = 1 - \mathcal{P}_{y\sim p}[y = 0] = 1 - p = 1 - p' + p' - p \leq \mathcal{P}_{y\sim p'}[y \neq y'] + |p - p'|.$$

In CLAL framework, the simplified score of distinctiveness is defined as,

$$S_{dis} = cos < x'_j, x_i > - cos < x'_k, x_i > + cos < x'_j, x'_k >,$$

where x'_j, x'_k represents the 2-nearest neighbor in labeled pool of x_i. And x_i is the feature vector learning by contrastive learning of the i^{th} point, with $|x_i| = 1$. The simplification of the score does not influence the performance of our algorithm. We assume that we can successfully solve the classification task by using the feature vectors. Then, we can calculate the Euclidean distance, denoted as d_i, between x_i, x'_j.

$$\begin{aligned} d_i &= \sqrt{|x_i| + |x'_j| - 2|x_i||x'_j| cos < x'_j, x_i >} \\ &= \sqrt{2 - 2cos < x'_j, x_i >} \\ &= \sqrt{2} \cdot \sqrt{1 - (S_{dis} + cos < x'_k + x_i > - cos < x'_j, x'_k >)} \\ &\leq \sqrt{2} \cdot \sqrt{3 - S_{dis}} \\ &= \sqrt{2} \cdot \sqrt{3 - \alpha}. \ (\alpha \leq 3, \ \forall x_i \in X^U) \end{aligned}$$

The population distribution of x_j is $\tau(x_j)$, and the empirical distribution of x_j is $\tau_k(x'_j)$, where x'_j is its nearest neighbor in labeled set.

$$\begin{aligned} & E_{y_i\sim\tau(x_i)}[l(x_i, y_i; \mathbf{w})] = \sum_{k\in C} p_{y_i\sim\tau_k(x_i)}(y_i = k) l(x_i, y_i; \mathbf{w}) \\ & \leq \left[\sum_{k\in C} p_{y_i\sim\tau_k(x'_j)}(y_i = k) l(x_i, y_i; \mathbf{w}) + \sum_{k\in C} |\tau_k(x_i) - \tau_k(x'_j)| l(x_i, y_i; \mathbf{w}) \right] \end{aligned}$$

For $\sum_{k\in C} |\tau_k(x_i) - \tau_k(x'_j)| l(x_i, y_i; \mathbf{w})$, we obtain

$$\sum_{k\in C} |\tau_k(x_i) - \tau_k(x'_j)| l(x_i, y_i; \mathbf{w}) = \sum_{k\in C} d_i \cdot l(x_i, y_i; \mathbf{w}) \leq \sqrt{6 - 2\alpha} \cdot \lambda^\mu LK.$$

For $\sum_{k\in C} p_{y_i\sim\tau_k(x'_j)}(y_i = k) l(x_i, y_i; \mathbf{w})$, we have

$$\begin{aligned} & \sum_{k\in C} p_{y_i\sim\tau_k(x'_j)}(y_i = k) l(x_i, y_i; \mathbf{w}) \\ & = \sum_{k\in C} p_{y_i\sim\tau_k(x'_j)}(y_i = k) \{l(x_i, y_i; \mathbf{w}) - l(x_j, y_j; \mathbf{w})\} + \sum_{k\in C} p_{y_i\sim\tau_k(x'_j)}(y_i = k) l(x_j, y_j; \mathbf{w}) \\ & \leq \lambda^l \sqrt{6 - 2\alpha}. \end{aligned}$$

Hence, $E_{y_i\sim\tau(x_i)}[l(x_i, y_i; \mathbf{w})] \leq \sqrt{6-2\alpha} \cdot (\lambda^{\mu} LK + \lambda^{l})$.

We further use the Hoeffding inequality [2]

$$\mathcal{P}\left\{ \left| \frac{1}{p} \sum_{i=1}^{p} l(x_i^{ID\text{-}C}, y_i; \mathbf{w}) - E_{y\sim\tau(x)}\left[l(x, y; \mathbf{w}) \right] \right| \geq t \right\} \leq exp(\frac{-2pt^2}{T^2}).$$

Let $exp(\frac{-2pt^2}{T^2}) = \gamma$. *With probability* $1-\gamma$, *we have*

$$\left| \frac{1}{p} \sum_{i=1}^{p-d} l(x_i^{ID\backslash re}, y_i; \mathbf{w}) - \frac{1}{q} \sum_{j=1}^{q} l(x_j^{tr}, y_j; \mathbf{w}) \right| \leq \sqrt{6-2\alpha}(\lambda^{l} + \lambda^{\mu} TK) + \sqrt{\frac{T^2 log(1/\gamma)}{2p}}.$$

A bound for invalid query error

With the boundary of the loss function $l(\cdot; \mathbf{w})$, *we have,*

$$\left| \frac{1}{p} \sum_{i=1}^{d} l(x_i^{re}, y_i; \mathbf{w}) \right| \leq \frac{dT}{p}$$

A bound for active learning error

Therefore, with probability $1-\gamma$, *we have*

$$\begin{aligned} &\left| \frac{1}{p} \sum_{i=1}^{p-d} l(x_i^{ID\backslash re}, y_i; \mathbf{w}) - \frac{1}{q} \sum_{j=1}^{q} l(x_j^{tr}, y_j; \mathbf{w}) \right| + \left| \frac{1}{p} \sum_{i=1}^{d} l(x_i^{re}, y_i; \mathbf{w}) \right| \\ &\leq \sqrt{6-2\alpha}(\lambda^{l} + \lambda^{\mu} TK) + \sqrt{\frac{T^2 log(1/\gamma)}{2p}} + \frac{dT}{p}. \end{aligned}$$

References

[1] *Dimitri P Bertsekas.* Constrained optimization and Lagrange multiplier methods. *Academic press, 2014.* 2

[2] *Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In* The Collected Works of Wassily Hoeffding, *pages 409–426. Springer, 1994.* 5

[3] *Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks.* arXiv preprint arXiv:1410.1141, *2014.* 3

[4] *Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In* Proceedings of the 32nd International Conference on Neural Information Processing Systems, *pages 3839–3848, 2018.* 1