Knowledge Transfer with Interactive Learning of Semantic Relationships

Supplementary Material

Smoothed Hinge Loss function $h_\rho(\cdot)$

In order to use the gradient descent optimization method at the peak points, we approximate them by smoothed versions as shown by the blue curves in Fig. 1 as in (Amit et al. 2007).

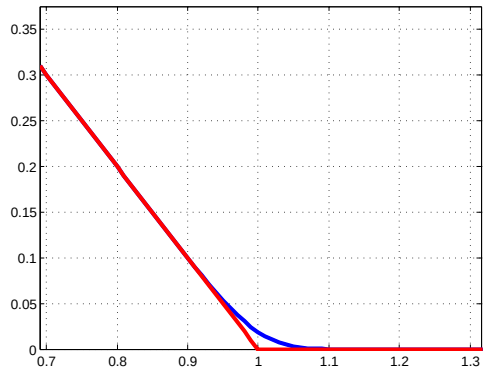

Figure 1: **Smoothed hinge loss.**

$h_\rho(\cdot)$ is the approximate hinge loss function that has no discontinuity:

$$h_\rho(z) = \begin{cases} 1-z & z < 1-\rho \\ \frac{-(1-z)^4}{16\rho^3} + \frac{3(1-z)^2}{8\rho} + \frac{(1-z)}{2} + \frac{3\rho}{16} & |1-z| \le \rho, \\ 0 & z > 1+\rho. \end{cases} \quad (1)$$

and its derivative with respect to z is

$$\frac{\partial h_\rho(z)}{\partial z} = \begin{cases} -1 & z < 1-\rho \\ \frac{(1-z)^3}{4\rho^3} - \frac{3(1-z)}{4\rho} - \frac{1}{2} & |1-z| \le \rho. \\ 0 & z > 1+\rho. \end{cases} \quad (2)$$

In our experiments we use $\rho = \sigma = 10^{-7}$.

Probability Mass Function

To compute the score by the entropy (Sec. 3.2 in the main paper), we define each entity's probability mass function by its classification confusion on validation set. Specifically, the probability of a label entity $\boldsymbol{u}_i$ to be a class label j is definedas:

$$P_{\boldsymbol{u}_i}(j) = \frac{\sum_{\boldsymbol{x}_k \in \mathcal{V}} \mathbb{1}(g(\boldsymbol{x}_k) = j)}{|\mathcal{V}|}, \quad (3)$$

where $g(\cdot)$ is the current classification model learned with $\boldsymbol{u}_i$ and $\boldsymbol{x}_k$ and $\mathcal{V}$ is a set of feature embeddings, $g(\boldsymbol{x}_k)$, in validation set. Thus $\mathbb{1}(g(\boldsymbol{z}_k) = j)$ equals to the number of feature embeddings whose obtained label by the current model is j. $|\cdot|$ denotes cardinality of a set. The ideal PMF is a delta function when $c = j$; $\delta(c = j)$. Note that the measure depends on the sample distribution under the current model. Thus, the entropy of an entity can be written as:

$$H(\boldsymbol{u}_i) = -\sum_{j \in \mathcal{C}} P_{\boldsymbol{u}_i}(j) \log P_{\boldsymbol{u}_i}(j), \quad (4)$$

where $\mathcal{C}$ is a set of all class labels. For the joint entropy, we need to derive a joint probability mass function of multiple label entities.

Joint Probability Mass Function of Multiple Entities

For computing a joint entropy, deriving a joint probability mass function (PMF) of multiple entities from Eq.(3) is straightforward. We start from the joint PMF of two entities, $P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2)$. Since the probability of $\boldsymbol{u}_i$ being label c_1 is dependent on the obtained labels of neighboring feature embeddings, $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N$, $P_{\boldsymbol{u}_i}(c_1)$ is actually a conditional probability as:

$$\begin{aligned} P_{\boldsymbol{u}_i}(c_1) &= P_{\boldsymbol{u}_i}(c_1|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) \\ &= P_{\boldsymbol{u}_i}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \mathcal{N}^i\}). \end{aligned} \quad (5)$$

We can write the joint PMF of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ as:

$$\begin{aligned} P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2) &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)P_{\boldsymbol{u}_j}(c_2) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2)P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \mathcal{N}^j\}) \\ &= P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})P_{\boldsymbol{u}_j}(c_2), \end{aligned} \quad (6)$$

the second to third line is because if $\boldsymbol{u}_j$ is given (or known), $\mathcal{N}^j$ are not necessary as conditioned variables; $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i \cup \mathcal{N}^j\}\}, c_2) =$

$P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|\{\boldsymbol{z}_k|\boldsymbol{z}_k \in \{\mathcal{N}^i - \mathcal{N}^j\} \cup c_2\})$. Then the conditional probability of $P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2)$ and the joint probability of $(\boldsymbol{u}_i, \boldsymbol{u}_j)$ can be written as:

$$P_{\boldsymbol{u}_i|\boldsymbol{u}_j}(c_1|c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \tag{7}$$

$$P_{\boldsymbol{u}_i,\boldsymbol{u}_j}(c_1, c_2) = \frac{\left(\sum_{\boldsymbol{z}_i \in \mathcal{N}^i - \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_1)\right) + \mathbb{1}(c_1 = c_2)}{|\mathcal{N}^i - \mathcal{N}^j| + 1} \cdot \frac{\sum_{\boldsymbol{z}_i \in \mathcal{N}^j} \mathbb{1}(g(\boldsymbol{z}_i) = c_2)}{|\mathcal{N}^j|}. \tag{8}$$

A joint PMF of more than three variables can be straightforwardly obtained by the chain rule.

Conditional Entropy

By the independence of variable for conditional entropy, we have the following equation:

$$\begin{gathered} H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}|\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) - H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = 0, \\ H(\boldsymbol{u}_{a_1}, \ldots, \boldsymbol{u}_{a_k}, \boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}) = H(\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_g}). \end{gathered} \tag{9}$$

Using Eq.(9) here, we can derive Eq.(6) in the main paper from Eq.(5) in the main paper as:

$$\begin{aligned} S(R, \boldsymbol{U}) &= H(\boldsymbol{u}_t, \boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}) \\ &= H(\boldsymbol{u}_t) - H(\boldsymbol{u}_{a_1}, \boldsymbol{u}_{a_2}). \end{aligned} \tag{10}$$

Computational Complexity

The computational complexity of Algorithm 1 depends on the complexity of training the model on the anchor and target classes, and generating a query pool.

First, the complexity of training the model on the anchor classes in Eq.(1) in the main paper for each $\boldsymbol{W}^{\boldsymbol{A}}$ and $\boldsymbol{U}^{\boldsymbol{A}}$ is $O(md(N^A + 1))$ and $O(m(dN^A + C^A))$ respectively, and the complexity of training the model for target categories in Eq.(2) in the main paper is $O(md(N^T + 2))$ and $O(m(dN^T + C^T + |\mathcal{R}|))$ for $\boldsymbol{W}$ and $\boldsymbol{U}$. It is dominated by $O(mdN^T)$ as $dN^T \gg C^T + |\mathcal{R}|$.

To generate a query pool (Sec.3.2), we first compute the probability mass function (PMF) for each label entity by $O(NC)$, where $N = N^A + N^T, C = C^A + C^T$ and a confusion matrix of label entities by its PMF with the complexity of $O(N^A C^A + N^T C^T)$. A naive way of enumerating all possible constraints takes $O(C^T {C^A}^2)$ but we generate a decent sized subset (k_p) to consider the most confusing entities' nearest neighboring label embeddings (C_r) by $O(k_p C_r^2)$. Thus, the complexity of generating the pool is $O(NC + N^A C^A + N^T C^T + k_p C_r^2)$. Re-scoring the pool using cross validation takes $O(k_p m d N^T)$. Finally, the outer loop of algorithm usually iterates few times and thus the total complexity of Algorithm 1 is $O(N^A(md + C^A) + N^T(k_p md + C^T) + NC + k_p C_r^2)$.

Test time complexity is $O(m(C^T + d))$, which is the same for all linear embedding methods.

Score Vector for Estimating Classification Improvement

The score vector $\boldsymbol{c}$ consists of confidence/confusion of t, a_1 and a_2 on both training set and validation set, geometric fitness $\left(\frac{\|\boldsymbol{u}_{a_2} - \boldsymbol{u}_t\|^2}{\|\boldsymbol{u}_{a_1} - \boldsymbol{u}_t\|^2}\right)$ and ball radius of sample distribution with respect to each class label prototype for t, a_1 and a_2.

Dataset Details

Low-Level Features. For visual features, we use the features provided by dataset authors (Lampert, Nickisch, and Harmeling 2014; Hwang, Grauman, and Sha 2013). The low-level features of both dataset is SIFT and other texture and color descriptors with PCA. In AWA dataset, we do PCA to reduce the dimensions to 300. In ImageNet-50, we use 1000 dimensional feature of same type of low level description to AWA dataset. We center the features by the sample mean.

Embedding Space Detail. For dimension of the embedding space, we choose 75, which is slightly bigger than the number classes (50) for encoding additional semantic information.

Animals with Attribute (AwA). There are 50 classes in total in AwA dataset (Lampert, Nickisch, and Harmeling 2014). Ten of them are target classes. The target classes of AwA dataset are 'Leopard', 'Pig', 'Hippopotamus', 'Seal', 'Persian Cat', 'Chimpanzee', 'Rat', 'Humpback Whale', 'Giant Panda' and 'Racoon'. The rest of the 40 classes of AwA serves as anchor classes.

ImageNet-50. There are 50 classes in total in the ImageNet-50 dataset (Hwang, Grauman, and Sha 2013). The 50 classes are randomly chosen from the entire ImageNet dataset. The 50 classes are: 'Kitfox', 'australianterrier', 'lesserpanda', 'egyptiancat', 'persiancat', 'cougar', 'badger', 'greatdane', 'scottishdeerhound', 'jaguar', 'blackfootedferret', 'skunk', 'corgi', 'weasel', 'colobus', 'orangutan', 'chimpanzee', 'gorilla', 'greyhound', 'hare', 'patas', 'baboon', 'macaque', 'tabby', 'raccoon', 'polecat', 'lion', 'cheetah', 'otter', 'sunflower', 'bonsai', 'strawberry', 'lamp', 'pooltable', 'acorn', 'drum', 'marimba', 'daisy', 'comb', 'rule', 'ferriswheel', 'rollercoaster', 'buckle', 'button', 'barnspider', 'gardenspider', 'bridge', 'featherboa', 'bathtub', 'basketball.'

Among them, we randomly choose ten of them are target classes. The target classes of ImageNet-50 dataset are 'cougar', 'weasel', 'colobus', 'gorilla', 'tabby', 'raccoon', 'pool-table', 'comb', 'roller-coaster', 'feather-boa'. The rest of the 40 classes of ImageNet-50 serves as anchor classes.

References

[Amit et al. 2007] Amit, Y.; Fink, M.; Srebro, N.; and Ullman, S. 2007. Uncovering Shared Structures in Multiclass Classification. In *ICML*.

[Hwang, Grauman, and Sha 2013] Hwang, S. J.; Grauman, K.; and Sha, F. 2013. Analogy-preserving semantic embedding for visual object categorization. In *International Conference on Machine Learning (ICML)*, 639–647.

[Lampert, Nickisch, and Harmeling 2014] Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans. on PAMI*.