GistNet: a Geometric Structure Transfer Network for Long-Tailed Recognition - supplemental

Bo Liu
UC, San Diego
boliu@ucsd.edu

Haoxiang Li
Wormpex AI Research
lhxustcer@gmail.com

Hao Kang
Wormpex AI Research
haokheseri@gmail.com

Gang Hua
Wormpex AI Research
ganghua@gmail.com

Nuno Vasconcelos
UC, San Diego
nuno@ece.ucsd.edu

1. Rotation matrix implementation

A rotation matrix in d-dimensional space has d-by-d parameters, which is infeasible to learn. And as it only has $d-1$ degrees of freedom, further constraints have to be implemented if the matrix is directly learned. To avoid these difficulties, we define the structure parameters as d-dimensional vectors, and for each vector, the rotation matrix from a basic vector $[1, 0, 0, \ldots, 0]^T$ to the give vector is used as the rotation on class weights.

In a general case, given two vectors $\mathbf{x}$ and $\mathbf{y}$, we want to find the rotation matrix $\mathbf{R}$ from $\mathbf{x}$ to $\mathbf{y}$. One way to do this is to find the plane spanned by $\mathbf{x}$ and $\mathbf{y}$, and then with respect to this, consider the 2D rotation on the plane. With Gram–Schmidt process, we find the orthonormal basis as

$$\mathbf{u} = \frac{\mathbf{x}}{||\mathbf{x}||}, \quad \mathbf{v} = \frac{\mathbf{y} - <\mathbf{u}, \mathbf{y}> \mathbf{u}}{||\mathbf{y} - <\mathbf{u}, \mathbf{y}> \mathbf{u}||}. \tag{1}$$

Therefore, $\mathbf{P} = \mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T$ is a projection onto the space spanned by $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{Q} = \mathbf{I} - \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T$ is the projection onto complemented subspace. The rotation only takes place on the plain of $\mathbf{P}$. In result, we can map the vector onto the plain with basis $\mathbf{u}$ and $\mathbf{v}$, do the rotation on it, map this back, and add the invariant part from the complemented subspace. The whole rotation matrix is

$$\mathbf{R} = \mathbf{I} - \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T + [\mathbf{u}, \mathbf{v}]\mathbf{R}_\theta[\mathbf{u}, \mathbf{v}]^T. \tag{2}$$

$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is defined as the 2D rotation matrix between $\mathbf{x}$ and $\mathbf{y}$, with $\cos\theta = \frac{<\mathbf{x}, \mathbf{y}>}{||\mathbf{x}||||\mathbf{y}||}$.

Given a structure parameter vector $\mathbf{y} = \delta_j$, we set $\mathbf{u} = \mathbf{x} = [1, 0, 0, \ldots, 0]^T$, and the rotation matrix $\mathbf{R}_j$ is calculated with (2). The parameter constellations are implemented as

$$\mathbf{w}_{kj} = g(\mathbf{w}_k, \delta_j) = \mathbf{R}_j \mathbf{w}_k \tag{3}$$

Table 1. Results on the iNaturalist 2018. All methods are implemented with ResNet-50.

Method	Accuracy
CB-Focal [4]	61.1
LDAM+DRW [3]	68.0
Decoupling [8]	69.5
GistNet	**70.8**

2. Details of baseline results

In [Table 1, paper], results of Plain Model, Lifted Loss [12], Focal Loss [10], Range Loss [16], FSLwF [6] are copied from [11]. Results of OLTR [11], Distill [15], CB Expert [13] are copied from their papers respectively. Places-LT results of Decoupling [8] are copied from the paper. ImageNet-LT results of it are reproduced with the authors' code, because the detailed results with ResNet-10 on three splits are not provided in the paper.

3. iNaturalist 2018 Results

We further evaluate our methods on the iNaturalist 2018 dataset, with ResNet-50 [7], and compare to state-of-the-arts. Results are listed in Table 1.

4. Geometry of the Cross-Entropy Classifier

A popular architecture for classification is the softmax classifier. This consists of an embedding that maps images $\mathbf{x} \in \mathcal{X}$ into feature vectors $f_\phi(\mathbf{x}) \in \mathcal{F}$, implemented by multiple neural network layers, and a softmax layer that estimates class posterior probabilities according to

$$p(y = k|\mathbf{x}; \phi, \mathbf{w}_k) = \frac{\exp[\mathbf{w}_k^T f_\phi(\mathbf{x})]}{\sum_{k'} \exp[\mathbf{w}_{k'}^T f_\phi(\mathbf{x})]} \tag{4}$$

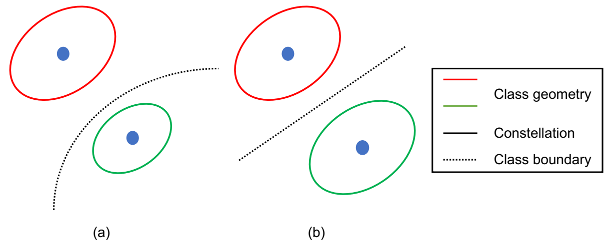

Figure 1. (a) The optimal classifier for two Gaussians of different covariance has quadratic boundary; (b) Linear boundary requires shared covariance.

where ϕ denotes the embedding parameters and $\mathbf{w}_k$ is the weight vector of the k^{th} class. The model is learned with a training set $\mathbb{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n^s}$ of n^s examples, by minimizing the cross entropy loss

$$\mathcal{L}_{CE} = \sum_{(\mathbf{x}_i, y_i)\in\mathbb{S}} -\log p(y_i|\mathbf{x}_i). \tag{5}$$

Recognition performance is evaluated on a test set $\mathbb{T} = \{(x_i, y_i)\}_{i=1}^{n^t}$, of n^t examples.

From Bayes rule

$$p(y=k|\mathbf{x}) = \frac{p(\mathbf{x}|y=k)p(y=k)}{\sum_{k'} p(\mathbf{x}|y=k')p(y=k')}, \tag{6}$$

the posterior distributions of (4) constrain the class-conditional distributions to the form

$$p(\mathbf{x}|y=k) \propto_{\mathbf{x}} \exp[\mathbf{w}_k^T f_\phi(\mathbf{x})], \tag{7}$$

where $\propto_{\mathbf{x}}$ means a proportional relationship that depends on $\mathbf{x}$. This implies that

$$p(\mathbf{x}|y=k) = h(f_\phi(\mathbf{x}))\exp[\mathbf{w}_k^T f_\phi(\mathbf{x}) - A(\mathbf{w}_k)] \tag{8}$$

where $h(.)$ is any non-negative function, and $A(.)$ a constant such that (8) integrates to 1. Hence, $p(\mathbf{x}|y=k)$ is an exponential family distribution of canonical parameter $\mathbf{w}_k$, sufficient statistic $f_\phi(\mathbf{x})$, cumulant function $A(\mathbf{w}_k)$ and underlying measure $h(.)$ [9]. This probability distribution can be uniquely expressed as [1]

$$p(\mathbf{x}|y=k) = u(f_\phi(\mathbf{x}))\exp[-d_\gamma(f_\phi(\mathbf{x}), \mu_k)], \tag{9}$$

where $\mu_k = \nabla A(\mathbf{w_k})$ is the mean of $p(\mathbf{x}|y=k)$, ∇A is the gradient of A, $u(.) = h(.)e^{\gamma(.)}$, and $d_\gamma(.,.)$ is the Bregman divergence [2] with respect to the function

$$\gamma(\mu_k) = \mathbf{w}_k^T\mu_k - A(\mathbf{w}_k). \tag{10}$$

Since the cumulant $A(.)$ defines $\gamma(.)$, it determines the distance function $d_\gamma(.,.)$ and thus the geometry of the embedding.

While the discussion above applies to any exponential family distribution, the equalities above are particularly simple to verify for the case where the class-conditional distributions are spherical Gaussians (covariances $\mathbf{\Sigma}_k = \sigma^2\mathbf{I}$). In this case

$$p(\mathbf{x}|y=k) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2}||f_\phi(\mathbf{x})-\mu_k||^2}, \tag{11}$$

which can be written as (9) with

$$d_\gamma(f_\phi(\mathbf{x}), \mu_k) = \frac{1}{2\sigma^2}e^{-||f_\phi(\mathbf{x})-\mu_k||^2}. \tag{12}$$

Similarly, they can be written in the form of (8), by expanding the 2-norm in the exponent and defining

$$h(f_\phi(\mathbf{x})) = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2}||f_\phi(\mathbf{x})||^2} \tag{13}$$

$$\mathbf{w} = \frac{1}{\sigma^2}\mu \tag{14}$$

$$A(\mathbf{w}) = \frac{\sigma^2}{2}||\mathbf{w}||^2. \tag{15}$$

From (10) it follows that $\gamma(\mu) = \frac{1}{2\sigma^2}||\mu||^2$, which generates the Bregman divergence of (12), leading to an Euclidean geometry for all classes and spherically Gaussian class conditionals.

The point of the discussion above is that the softmax form of (4) constrains the geometry of the embedding, by defining the distance $d_\gamma(.,.)$. The fact that (4) is a linear classifier, i.e. has *linear class boundaries,* places further constraints on this geometry. Consider the case where the Gaussian class-conditionals have different covariances Σ_k. In this case, the optimal classifier is a "softmax" of the form $p(y=k|\mathbf{z}) = (e^{-\mathbf{z}^T\Sigma_k^{-1}\mathbf{z}+\mathbf{w}_k^T\mathbf{z}-\mathbf{b}_k})/(\sum_j e^{-\mathbf{z}^T\Sigma_j^{-1}\mathbf{z}+\mathbf{w}_j^T\mathbf{z}-\mathbf{b}_j})$ [5], where $\mathbf{z} = f_\phi(\mathbf{x})$, and has *quadratic* boundaries, as shown in the left of Figure 1. The problem is that this classifier would require a different divergence

$$d_{\gamma_k}(f_\phi(\mathbf{x}), \mu_k) = \frac{1}{2\sigma^2}e^{-(f_\phi(\mathbf{x})-\mu_k)^T\mathbf{\Sigma}_k^{-1}(f_\phi(\mathbf{x})-\mu_k)} \tag{16}$$

per class, and this is not feasible under the discussion of the previous section. While the Gaussian of generic covariance can still be written in the exponential family form, this requires a quadratic transformation of $f_\phi(\mathbf{x})$.

It follows that the linear classifier is optimal *if and only if* the covariance is shared, i.e. $\Sigma_k = \Sigma, \forall k$, in which case all quadratic terms of $f_\phi(\mathbf{x})$ can be absorbed in $h(f_\phi(\mathbf{x}))$, as in (13), and thus cancel when (8) is inserted in (6). This is illustrated in the right of Figure 1. It follows that learning with the softmax model indirectly encourages the classes to have shared geometry. If the geometry were different, the CNN of (4) *could not* be optimal.

While learning with (5) produces a particular embedding geometry, which we denote the *natural* geometry for the training data, it is usually impossible to determine this geometry from the learned network parameters, because the cumulant $A(\mathbf{w}_k)$ is not observable in (4). On the other hand, it is possible to enforce a certain geometry through regularization. The simplest way to implement this *geometric regularization* is to implement the classifier with (9) and a pre-specified divergence $d_\gamma(.,.)$, e.g. the Euclidean distance. This encourages a desired geometry, e.g. Euclidean, and corresponding class-conditionals, e.g. Gaussian, even for a network trained with a few examples per class. Hence, it can improve generalization when training data is scarce. This type of regularization is leveraged by popular architectures for few-shot learning, e.g. the prototypical network [14].

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005. 2

[2] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. 2

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1565–1576, 2019. 1

[4] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 1

[5] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012. 2

[6] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020. 1

[9] Morton Kupperman et al. Probabilities of hypotheses and information-statistics in sampling from exponential-class populations. *The Annals of Mathematical Statistics*, 29(2):571–575, 1958. 2

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1

[11] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 1

[12] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016. 1

[13] Saurabh Sharma, Ning Yu, Mario Fritz, and Bernt Schiele. Long-tailed recognition using class-balanced experts. *arXiv preprint arXiv:2004.03706*, 2020. 1

[14] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 3

[15] Liuyu Xiang and Guiguang Ding. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. *arXiv preprint arXiv:2001.01536*, 2020. 1

[16] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017. 1