Adaptive Subspaces for Few-Shot Learning

Christian Simon, Piotr Koniusz,
Richard Nock, Mehrtash Harandi

CVPR 2020

Motivation

- Formulate FSL as generating dynamic classifier
  - Learning a universal feature extractor
  - Learning to generate a classifier dynamically from limited data.
- Method
  - Generate subspace from limited data
  - Singularity Vector Decomposition
  -

FSL review

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{W}_c^\top f_\Theta(\boldsymbol{q}))}{\sum_{c'} \exp(\boldsymbol{W}_{c'}^\top f_\Theta(\boldsymbol{q}))} = \frac{\exp(d_c(\boldsymbol{q}))}{\sum_{c'} \exp(d_{c'}(\boldsymbol{q}))},$$

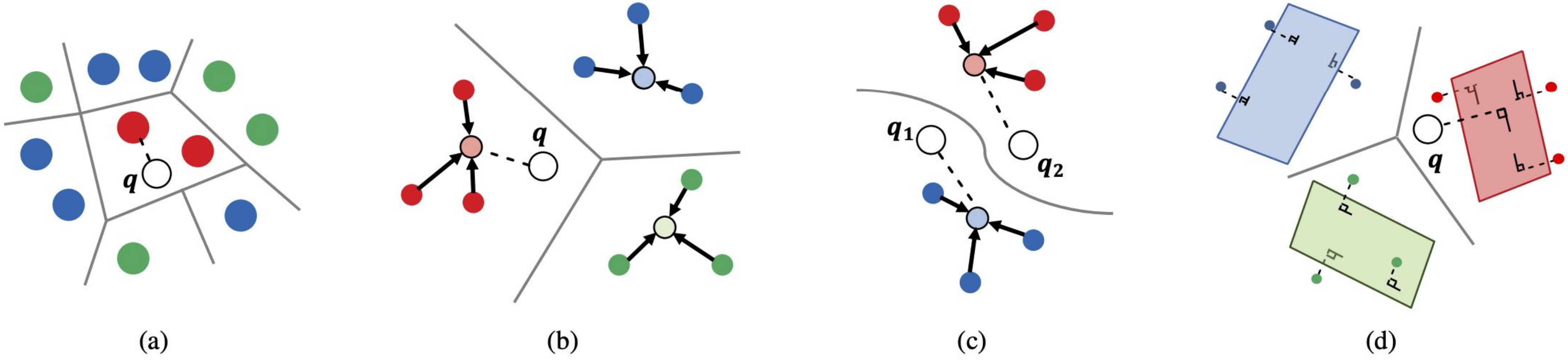

Figure 2: Various classifiers for few-shot classification. (a) Matching networks create pairwise classifiers. (b) Prototypical networks create mean classifiers based on the sample in the same class. (c) Relation networks produce non-linear classifiers. (d) Our proposed method creates classifiers using subspaces.

Subspace for few-shot

A subspace Z is represented by a basis. This paper try to generate the basis

$$\mathbb{R}^{D\times n} \ni \boldsymbol{B}_i = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n]; n \leq D.$$

A basis for a class can be compute by matrix decomposition (e.g.SVD)

$$\boldsymbol{B}_i^\top \boldsymbol{B}_i = \mathbf{I}_n$$

Generate subspace

Sample representation

$$\tilde{\boldsymbol{X}}_c = [f_{\Theta},(\boldsymbol{x}_{c,1}) - \boldsymbol{\mu}_c, \cdots, f_{\Theta}(\boldsymbol{x}_{c,K}) - \boldsymbol{\mu}_c]$$

where $\boldsymbol{\mu}_c = \frac{1}{K} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_c} f_{\Theta}(\boldsymbol{x}_i).$

Class representation as a subspace **Pc** from **Xc** by **SVD**

Subspace classifier

Given a query instance q, distance to subspace is

$$d_c(\boldsymbol{q}) = -\|(\mathbf{I} - \boldsymbol{M}_c)(f_\Theta(\boldsymbol{q}) - \boldsymbol{\mu}_c)\|^2$$

Distance distribution

$$p_{c,q} = p(c|\boldsymbol{q}) = \frac{\exp\left(d_c(\boldsymbol{q})\right)}{\sum_{c'} \exp\left(d_{c'}(\boldsymbol{q})\right)}$$

Discriminative Deep Subspace Network

Maximize subspace distance using Grassmannian geometry

$$\delta_p^2(\boldsymbol{P}_i, \boldsymbol{P}_j) = \left\| \boldsymbol{P}_i \boldsymbol{P}_i^\top - \boldsymbol{P}_j \boldsymbol{P}_j^\top \right\|_F^2 = 2n - 2\|\boldsymbol{P}_i^\top \boldsymbol{P}_j\|_F^2.$$

Final loss function

$$-\frac{1}{NM} \sum_c \log(p_{c,q}) + \lambda \sum_{i \neq j} \|\boldsymbol{P}_i^\top \boldsymbol{P}_j\|_F^2.$$

Mean refinement

Make use of unsupervised data

$$\tilde{\boldsymbol{\mu}}_c = \frac{K\boldsymbol{\mu}_c + \sum_i m_i f_\Theta(\boldsymbol{r_i})}{K + \sum_i m_i}$$

Where

$$m_i = \frac{\exp(-\|f_\Theta(\boldsymbol{r}_i) - \boldsymbol{\mu}_c)\|^2)}{\sum_{c'} \exp(-\|f_\Theta(\boldsymbol{r}_i) - \boldsymbol{\mu}_{c'})\|^2)}$$

Algorithm

Algorithm 1 Train Deep Subspace Networks

Input: Each episode $\mathcal{T}_i$ with S and Q

$\Theta_0 \leftarrow$ random initialization
for t in $\{\mathcal{T}_1, ..., \mathcal{T}_{N_\mathcal{T}}\}$ **do**
    for k in $\{1, ..., N\}$ **do**
        $\tilde{\boldsymbol{X}}_c \leftarrow \boldsymbol{S}_c$
        Calculate the average of the class
        Calculate mean refinement (MR) using Eq. 6
        Subtract $\tilde{\boldsymbol{X}}_c$ with an offset
        $[\mathcal{U}, \Sigma, \mathcal{V}^\top] \leftarrow \text{Decompose}(\tilde{\boldsymbol{X}}_c)$
        $\boldsymbol{P}_c \leftarrow$ Truncate $\mathcal{U}_{1,\ldots,n}$
        for $\boldsymbol{q}$ in Q **do**
            Compute $d_c(\boldsymbol{q})$ using Eq. 2
        end for
    end for
    Compute final loss $\mathcal{L}_t$ using Eq. 5
    Update Θ using $\nabla \mathcal{L}_t$
end for

Result

Model	Backbone	1-shot	5-shot
Matching Nets [4]	Conv-4	43.56 ± 0.84	55.31 ± 0.73
MAML [7]	Conv-4	48.70 ± 1.84	63.11 ± 0.92
Reptile [48]	Conv-4	49.97 ± 0.32	65.99 ± 0.58
R2-D2 [49]	Conv-4	48.70 ± 0.60	65.50 ± 0.60
Prototypical Nets [20]	Conv-4	44.53 ± 0.76	65.77 ± 0.66
Relation Nets [14]	Conv-4	50.44 ± 0.82	65.32 ± 0.70
DSN	Conv-4	$\mathbf{51.78 \pm 0.96}$	$\mathbf{68.99 \pm 0.69}$
DSN-MR	Conv-4	$\mathbf{55.88 \pm 0.90}$	$\mathbf{70.50 \pm 0.68}$