Tab 1

PRML Folder PRML
Medium blog :
https://medium.com/@aryan1113/long-tailed-visual-recognition-0bee1540394d
Presentation link PRML Long Tailed ProCo
Recording Long Tailed Visual Recognition Take1

Paper [in the folder] has highlighted portions telling the crux
Survey paper: A Survey on Long-Tailed Visual Recognition (also in folder)

AIM : Sample infinite contrastive pairs from estimated distribution

Long Tailed Distributions :

Data quality affects performance of our learner, and due to data being dominated from few classes, the learning of tail classes is severely underdeveloped.
Data in the tail classes is often insufficient to represent the true distribution. When a class is severely underrepresented, it becomes difficult to determine the decision boundary in our parameter/search space.

How to measure long tailed-ness of a distribution ?

1. **Imbalance factor** : ratio of maximum number of samples in a class and minimum number of samples in a class
2. **Standard deviation** : difficult to objectively express, as it is relative
3. **Mean / median ratio** : reflects skew
4. **Gini coefficient** : measure of inequality
   0 perfect equality ; 1 one class has all samples
   $\Sigma\ (x_i - x_j)\ /\ 2n^2\ *\ \mu$, differences btw i and j, n is the total number of samples and μ is the mean of the distribution

Using conventional methods for learning the distributions often result in poor performance as these assume that the data (both train and test) satisfy the I.I.D (independent, identically distributed) condition.

Naive methods to treat Long Tailed Distributions :

These methods are simple, data processing based, more of these are explored in the paper “A survey on Long Tailed Visual Recognition” by the Beijing University of Posts and Telecommunications.

Aim of these methods is to eliminate / minimize the imbalance between head and tail classes.

1. **Over sampling**

   Increase the instance number of the tail classes

   Class Aware sampling ensures that the probability of occurrence of each class is the same in each training batch. Probability of each class is 1/C, where C is the number of classes in our entire set.

2. **Under sampling**

   Decreases the instance number of head classes
   Random undersampling randomly removes instances of the head classes
   For long tailed distributions, we loose a lot of information because of the large difference between the head and tail classes.
   Drawbacks :
   a. May lead to overfitting, as we reduce overall size of our set
   b. Effects of noise / other defects are exaggerated
   c. If we remove too many instances, we risk not learning anything, due to under learning of the head classes as well.

3. **Data Augmentation**

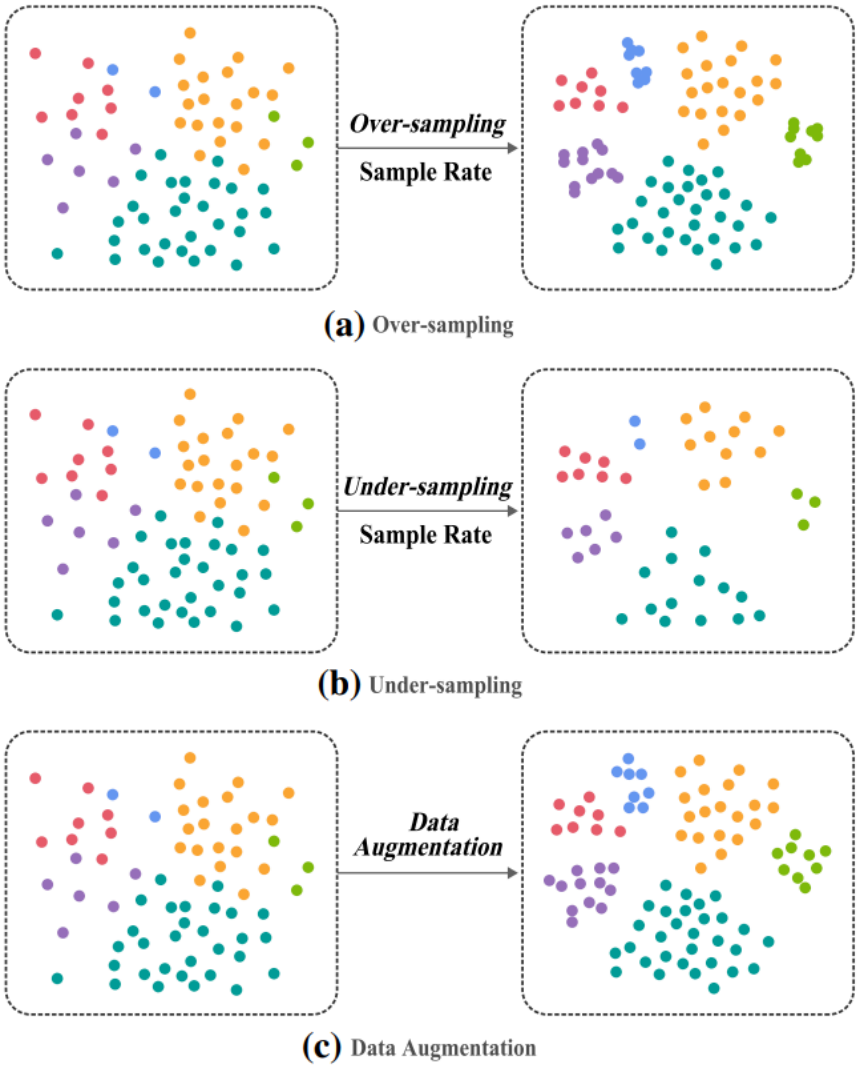
   Generate and synthesize new samples from the tail classes.
   Some common ways to augment image data are :
   1. Image flipping

2. Scaling (zoom)
3. Rotating
4. cropping

SMOTE is a common method for data synthesis for the minority class.

1. For each sample in the minority class, identify k-nearest neighbours within the same class based on distance.
2. Randomly choose a neighbour and interpolate a new datapoint using the formula, where λ is between 0 and 1

$$x_{newsample} = x + \lambda (x_{neighbour} - x)$$



What are Contrastive pairs ?

Samples used to help the learner make decisions, what's similar and what's different.
There could be positive samples, i.e similar to the class we are talking about. The learner tries to push these closer together in the feature space.

Negative samples are ones different from the current class, and the learner tries to push these apart in the feature space.

For Supervised Learning approaches,
Positive pairs: Any two samples from the same class label
Negative pairs: Any two samples from different class labels

Example :
If our anchor image is of a cat, another cat image belongs to the +ve class and a dog image is a -ve class sample.

Logit Adjustment :

modify the log loss function to account for frequency of the class as well

In the usual log loss, we treat all classes as equals and hence the learner is biased towards reducing loss majorly for the dominant class.
Factoring in the frequency of the classes help the learner also focus on rarer classes (present on the tail)

For class i, if the score at the output layer is z_i, the class probability is given by :

$$p_i = \frac{e^{z_i}}{\sum e^{z_i}}$$ For adjusted logit, use $z_i = z_i - \lambda log(f_i)$

von Mises Fisher vMF distribution

Equivalent of a Gaussian distribution; for data on a sphere/hypersphere.
All points lie on the surface of a unit sphere (length = 1)
Defined by two parameters:

- μ : mean direction
- kappa: concentration parameter; how tightly clustered is the data

If k = 0, vMF is a uniform distribution on the sphere
As k increases, the distribution becomes more concentrated around mean μ
Imagine spreading butter on a sphere:

- μ tells you the center point of where you're spreading
- κ tells you how widely you spread it
- Higher κ = more concentrated (like cold butter)
- Lower κ = more spread out (like melted butter)

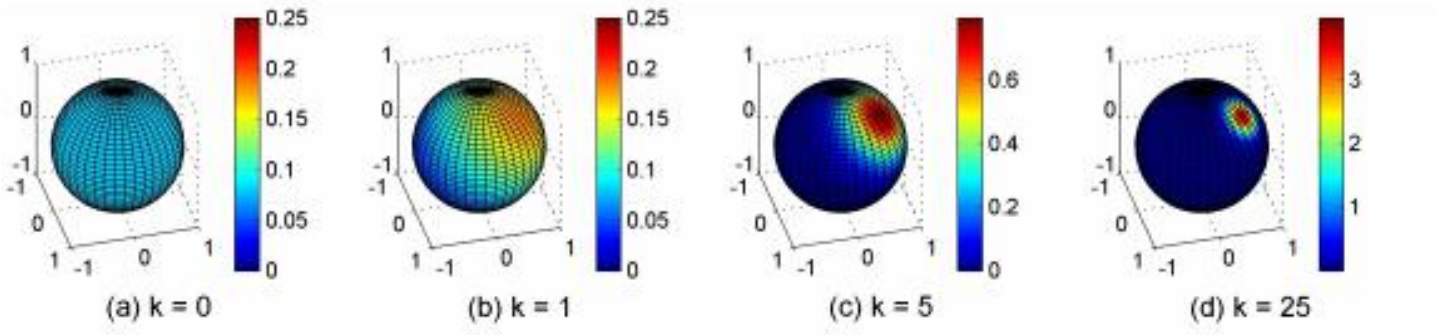
Notice the bessel function $I_{p/2-1}$ term in $C_p(k)$

$$f_p(\boldsymbol{z};\boldsymbol{\mu},\kappa) = \frac{1}{C_p(\kappa)} e^{\kappa \boldsymbol{\mu}^\top \boldsymbol{z}},$$

$$C_p(\kappa) = \frac{(2\pi)^{p/2} I_{(p/2-1)}(\kappa)}{\kappa^{p/2-1}},$$

Probability density function for a vMF distribution

Visual Analogy
A higher kappa shows data is concentrated in a small region



Parameter Estimation

Here f_p is the probabiliy distribution function of a vMF, and π_y ais the probability of class y. We estimate the mean and kappa params of the feature distribution.

$$P(z) = \sum P(y)\, P(z \mid y) = \sum \pi_y * f_p$$

proCO

SCL	PCL
Supervised Contrastive Learning	Probabilistic Contrastive Learning
Consider datapoints with the same labels as +ve examples,	Estimate feature distribution (estimate parameters of the vMF

and the rest are viewed as -ve examples.	distribution for features)
Limited by the actual samples in a batch	Use these estimates to generate contrastive pairs

Issues with Supervised Contrastive Learning

- If batch is too small, we might
    - not have examples with the same labels
    - miss out on important -ve examples
- Long Tailed distributions make it even harder to have +ve samples, as for rarer classes we might need very large batches to get adequate number of samples.
- Having a large batch cause
    - longer training time
    - higher memory usage
    - higher computational cost

Loss Function

The loss function for Supervised Learning is given by L_sup

$$L_{sup} = - log(\frac{\sum exp(z_i * z_p / \tau)}{\sum \sum exp(z_i . z_a / \tau)})$$

z_p feature vector of +ve samples, z_a feature vector of -ve samples
An anchor point is a reference point in contrastive learning

Why double summation in the denominator ?

outer : loops over all class j

inner : loops over all examples from -ve classes

τ scaling magnitude of similarity scores

- smaller τ enlarges differences between similarity scores, softmax output becomes sharper, as the next highest number is farther away
- larger τ compresses differences, smoother weighing for softmax

Goal of our loss function is to :

1. maximize similarity btw anchor z_i and with +ve examples z_p
2. maximize dis-similarity of anchor z_i and -ve class z_a, or in other words, penalizes similarity with -ve class.

For ProCo Probabilistic Contrastive Learning, z_p and z_a are replaced by expectations over the vMF distributions for +ve and -ve classes.

+ve term $A_p(k_{yi}) * \mu_{yi}$

Where μ_{yi} is the directional mean for class y_i and $A_p(k_{yi})$ is the contribution of class concentration k_{yi} to +ve similarity

-ve term $\sum \pi_j \frac{C_p(k_j)}{C_p(\kappa\sim_j)}$

Where π_j is the frequency of class j, represented in a fraction

$C_p(k_j)$ normalizes the vMF, ensuring it integrates to 1 over the unit hypersphere, where k is the concentration parameter controlling the shape of the distribution.

K~ is the adjusted concentration parameter
Large value of k~ implies strong alignment of anchor and mean direction for class j
Interaction between anchor sample z_i and vMF for class j is given by

$$k\sim_{j} = || k_{j}\pi_{j} + z_{i} / \tau ||_{2}$$

To compute higher order bessel functions, make use of recurrence relation; which is numerically unstable with lower values of kappa, the concentration parameter

$$I_{v+1}(k) = \frac{2v}{k} I_{v}(k) - I_{v-1}(k)$$

Two Branch Design

Imagine a
tree, with the root feeding into two branches

1. Classification Branch
   Uses Linear classifier, optimized with Adjusted Logit
   Learns to classify

2. Representation Branch
   Uses neural network, optimized with ProCO
   α is the strength parameter
   Learns what the good features are

The two branches handle rare cases without sacrificing performance for examples from common classes

Final Loss function $L = L_{adjusted\ logit} + \alpha L_{proCO}$