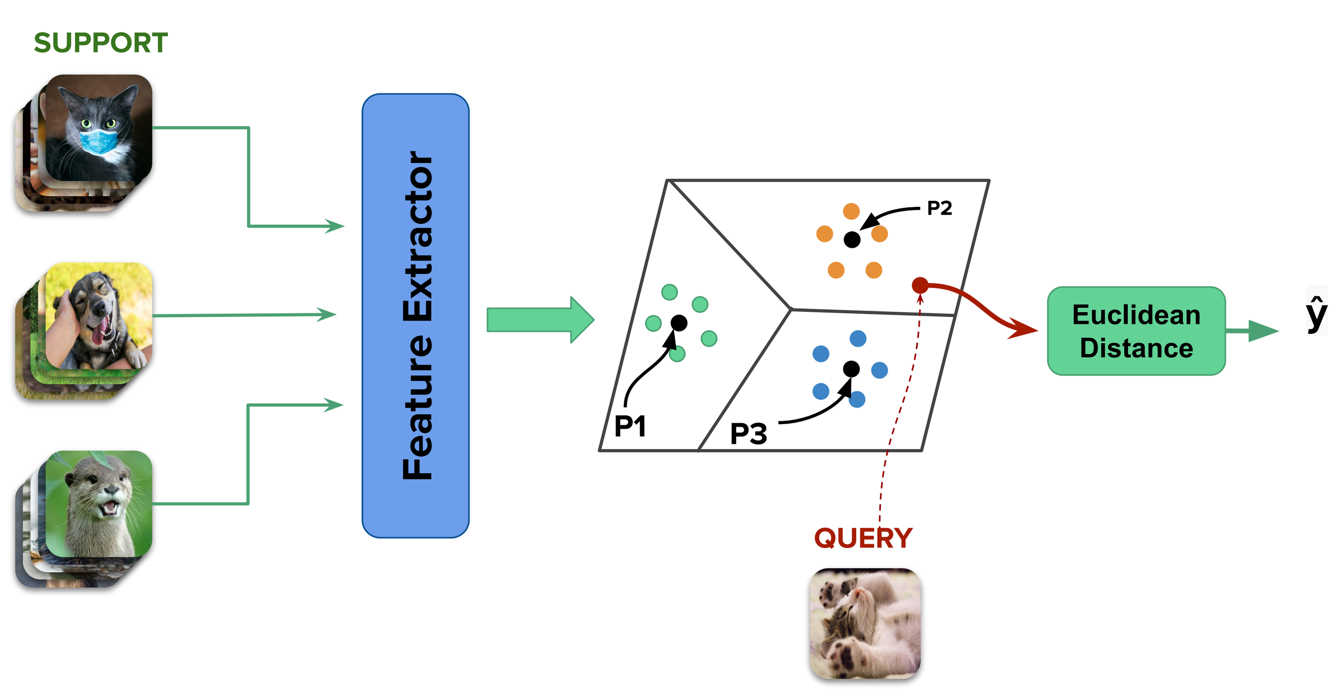
Finding Significant Features for Few-Shot Learning using Dimensionality Reduction Techniques

Mauricio Mendez Ruiz, Ivan Garcia, Andres Mendez-Vazquez and Gilberto Ochoa-Ruiz

1. Context and Motivation

Few-shot learning is a fairly new technique that specialize in problems where we have **little amount of data.** The goal of this method is to classify categories that hasn't been seen before with just a handful of samples.

Recent approaches, such as **metric learning**, adopt the meta-learning setting in which we have episodic tasks conformed by support (training) data and query (test) data. Metric learning methods has demonstrated that simple models can achieve good performance, by learning a similarity function to compare the support and the query data.

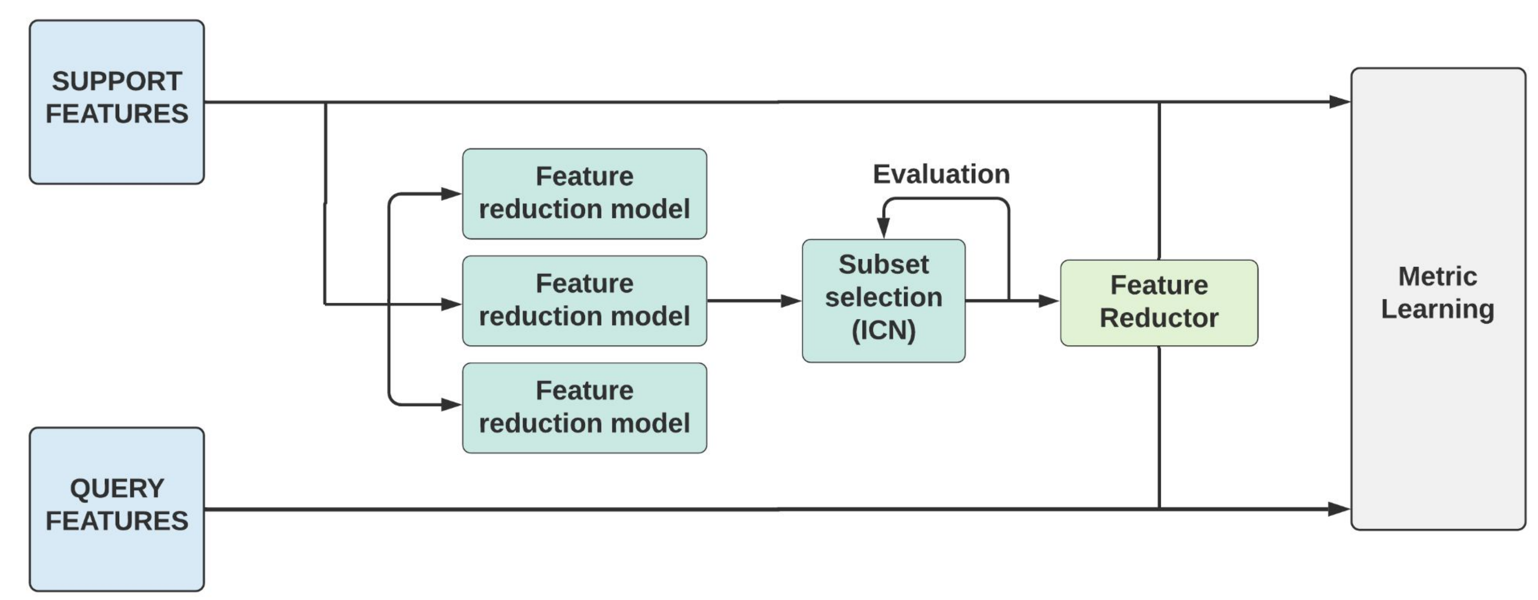


However, the feature space learned by the metric learning **may not exploit the information given by a specific few-shot task.** In this work, we explore the use of dimension reduction techniques as a way to find task-significant features.

We believe that finding those **relevant features for each task** is important, as we can better discriminate between classes and obtain a better inference.

2. Proposed Method

From the feature embeddings, we apply different **feature reduction** methods, and obtain an **intra-class and inter-class score** for each one. These scores are used to select the method which helps us to obtain the best dimensions for the current task. The features obtained are then used by a metric learner to produce a classification.



The **ICNN Score** is a measure that combines the distance and variance of the inter-intra k-nearest neighbors of each instance in the data

1) ICNN Formula

$$ICNN(X) = \frac{1}{|X|}\sum_{x_i \in X} \lambda(X_i)^{\frac{1}{p}}\omega(X_i)^{\frac{1}{p}}\gamma(X_i)^{\frac{1}{p}}$$

λ is a function that penalizes the neighbors of Xi with the same class based on how distant they are, and the neighbors of different classes based on how close they are:

$$\lambda(X_i) = \frac{\sum_{p \in K_{\bar{x}_i}} \frac{d(X_i,p)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)} \sum_{q \in K_{x_i}} 1 - \frac{d(X_i,q)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)}}{|K_{x_i}| + |K_{\bar{x}_i}|}$$

ω is a function that penalizes the distance variance of neighbors:

$$\omega(X_i) = 1 - (Var(\sum_{p \in K_{\bar{x}_i}} \frac{d(X_i,p)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)}) + Var(\sum_{q \in K_{x_i}} 1 - \frac{d(X_i,q)-\theta(X_i)}{\alpha(X_i)-\theta(X_i)}))$$

γ function describes the ratio of the neighbor's classes

$$\gamma(x_i) = \frac{|K_{x_i}|}{|K_{x_i}| + |K_{\bar{x}_i}|}$$

3 Results and Discussion

We compared the three main metric learning methods and our proposed model, obtaining an **improvement of around 1.5% for the 5-way 1-shot** setting on the test set using Prototypical Networks.

Method	5-way 1-shot	5-way 5-shot
Matching Net [20]	43.56	55.31
Prototypical Net [17]	49.52	**68.20**
Relation Net [18]	50.44	65.32
Prototypical Net + ICNN	**50.96**	67.72

In order to fit the feature reduction model, we can use only the support data or the data from the support and the query set. We found that using the support and query data, allowed the feature reduction methods to better interpret the structure of the data, thus obtaining a better ICNN score.

Future work:

There are still some design choices we need to test

1. We need to test if reducing to different **number of components** would give a better ICNN
2. Test the feature reduction with ICNN score only on the testing phase.
3. Try other **embedding networks** (variations of ResNet)
4. Add a **pre-training** phase to improve the performance

Our experiments are based on the combination of Prototypical Networks and ICNN but, as this method is proposed to obtain better features, any other metric learning technique (Matching Networks, Prototypical Networks) should expected to improve. The experimentation with these other techniques are left for future work.