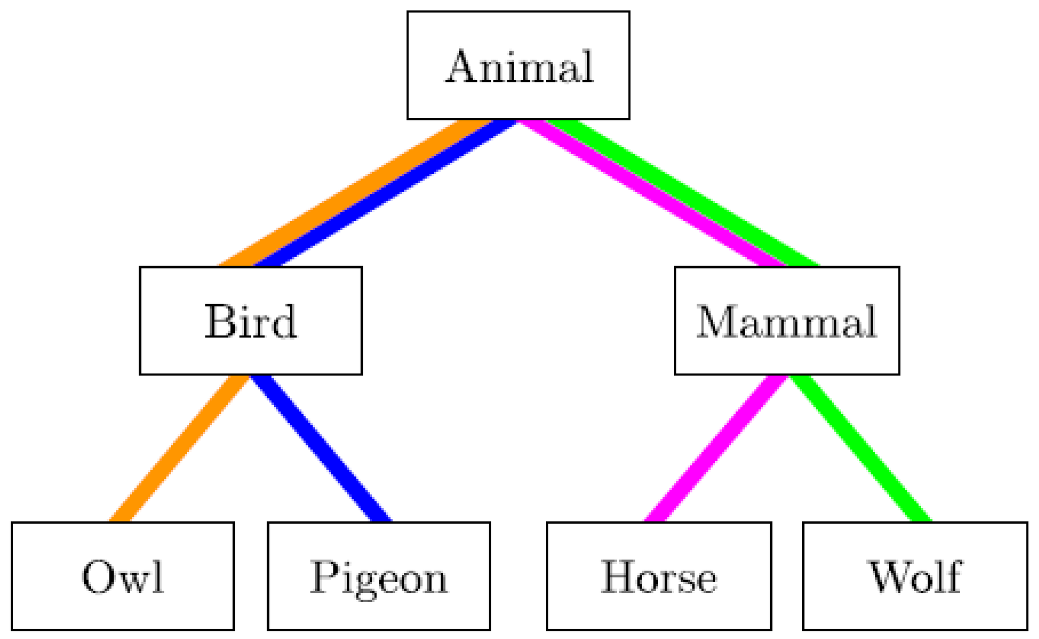
Learning Hierarchical Similarity Metrics

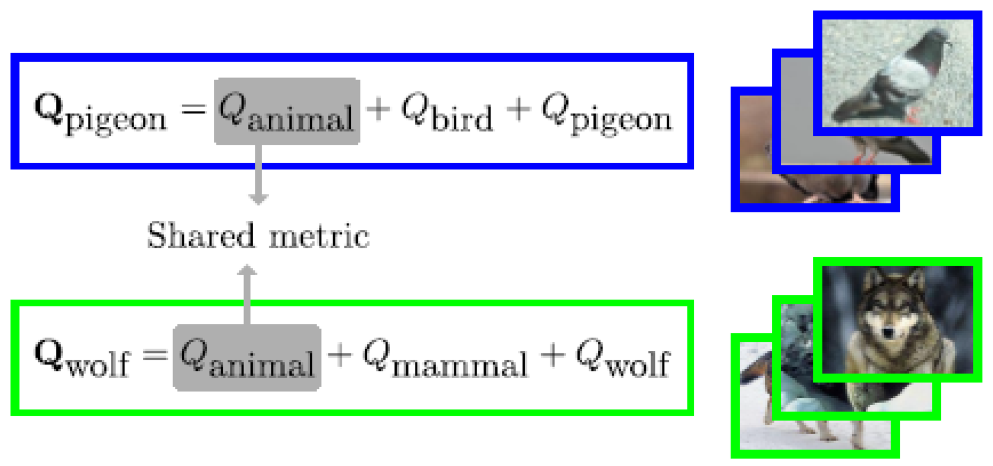
Nakul Verma — UC San Diego — naverma@cs.ucsd.edu

Dhruv Mahajan, Sundararajan Sellamanickam, Vinod Nair — Yahoo! Labs, Bangalore — {dkm, ssrajan, vnair}@yahoo-inc.com

Introduction

- Performance of many classification algorithms relies heavily on having a good notion of similarity or a metric on the input space.
- Learning good similarity metrics is especially hard for image categorization, with hundreds of categories.
- **Observation:** categories in multiclass data are often part of a underlying semantic taxonomy.
- **Goal:** to learn ***similarity metrics*** that leverage the class taxonomy to yield good classification performance.





Key Idea

- Associate a ***separate*** metric with each node of the taxonomy, and ***distribute*** the burden of discriminating amongst categories.
- Information is ***shared*** between the metrics using the parent-child relationships.

Advantage:

- Sharing helps to **distribute the burden** of category recognition: each metric is mainly responsible for discriminating amongst the categories associated with its siblings and children.
- Since each metric is responsible to **discriminate amongst only a few categories**, the overall classification becomes easier!
- Using the hierarchy enables us do well on **hierarchy specific** tasks.

Formulation

- Given a class taxonomy with T nodes, associate metrics $Q_1, \dots, Q_T$ one with each node. We call them ***local*** metrics.
- Define the aggregate metrics $\mathbf{Q}_1, \dots, \mathbf{Q}_T$ as the ***combination*** of the local metrics (from root to the node):

$$\mathbf{Q}_t := Q_t + \mathbf{Q}_{\text{parent}} = Q_t + \sum_{i \in \text{ancestor}(t)} Q_i$$

- We can thus define **distance** between any two examples x_1 and x_2 with respect to a metric $\mathbf{Q}_t$ as

$$\rho(x_1, x_2; \mathbf{Q}_t) := (x_1 - x_2)^{\mathsf{T}} \mathbf{Q}_t (x_1 - x_2)$$

- Now, for an arbitrary example x_q, we can measure its **affinity** to a class y as its distance to the nearest neighbors $\mathcal{N}_y(x_q)$ in class y (using metric $\mathbf{Q}_y$)

$$f(x_q; y) := \textstyle\sum_{x \in \mathcal{N}_y(x_q)} \rho(x_q, x; \mathbf{Q}_y)$$

- In a probabilistic framework, we can define the probability of an example x belonging to class y as:

$$p(y|x, Q_1, \dots, Q_T) := \frac{\exp(-f(x; y, \mathbf{Q}_y))}{\sum_{\bar{y}} \exp(-f(x; \bar{y}, \mathbf{Q}_{\bar{y}}))}$$

- Now, given training samples: $(x_1, y_1), \dots, (x_n, y_n)$ we obtain a good set of metrics $Q_1, \dots, Q_T$ by maximizing:

$$\mathcal{L}(Q_1, \dots, Q_T) := \frac{1}{n} \sum_{i=1}^{n} \log p(y_i | x_i; Q_1, \dots, Q_T) - \frac{\lambda}{2} \sum_t \text{trace}(Q_t^{\mathsf{T}} Q_t)$$

subject to PSD constraint $Q_t \succeq 0$

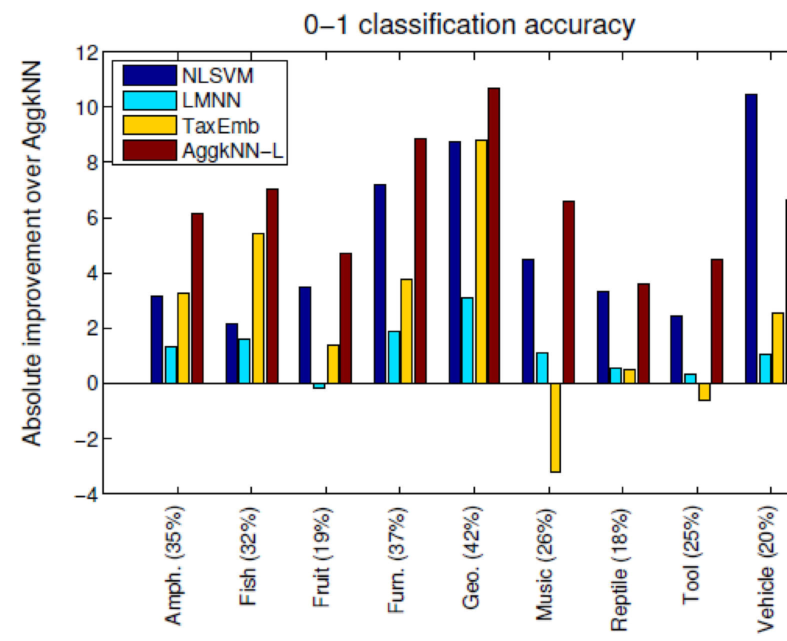
Observations

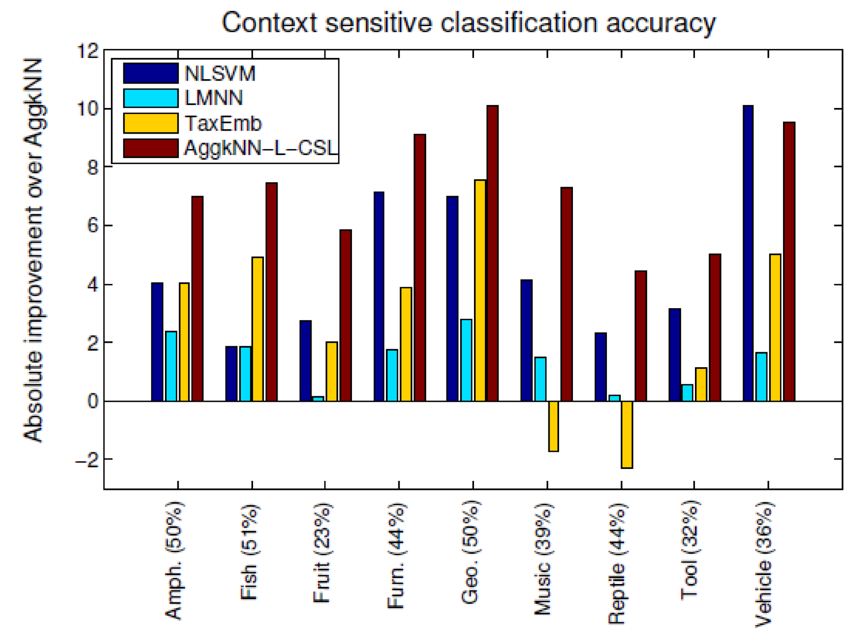
- Optimization is jointly **convex**.
- Geometrically, the likelihood is maximized by: ***pulling together*** the neighbors belonging to the same class, while ***pushing away*** the neighbors from different class.
- The regularization reduces the complexity of the learned metrics.
- The optimization can be easily modified to incorporate context sensitive loss.

Experimental results

Improved classification performance

- Good accuracy on various subtrees of ImageNet datasets from LSVRC challenge.
- Features: SIFT-based bag-of-words representation (provided), vocabulary size 1000-dimensional, reduced to 250 with PCA.

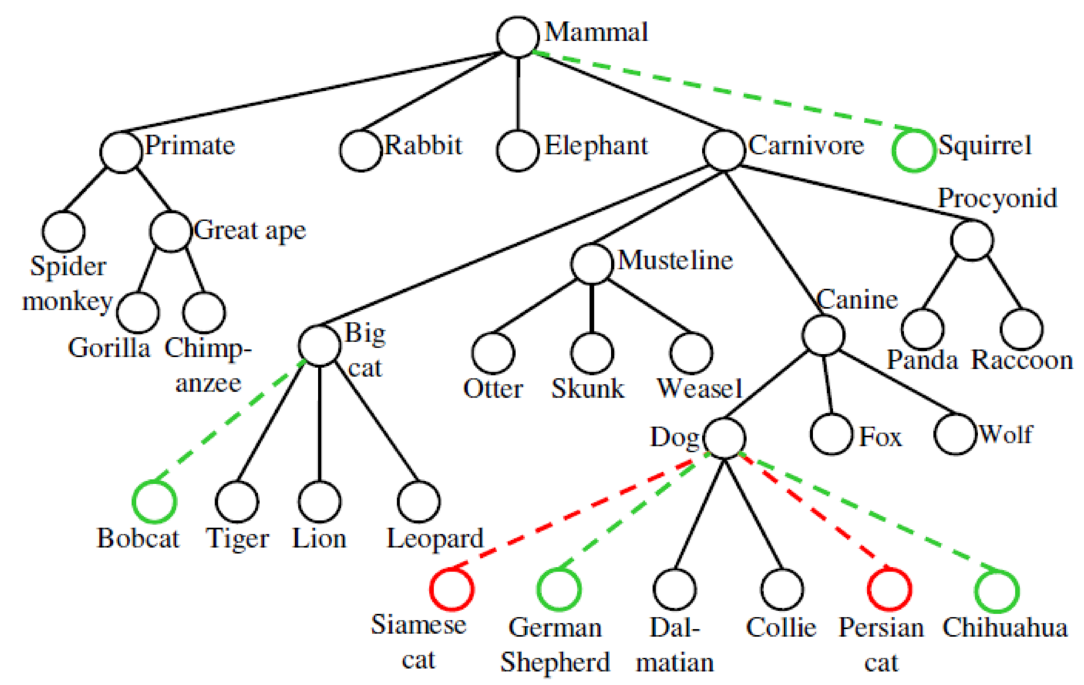


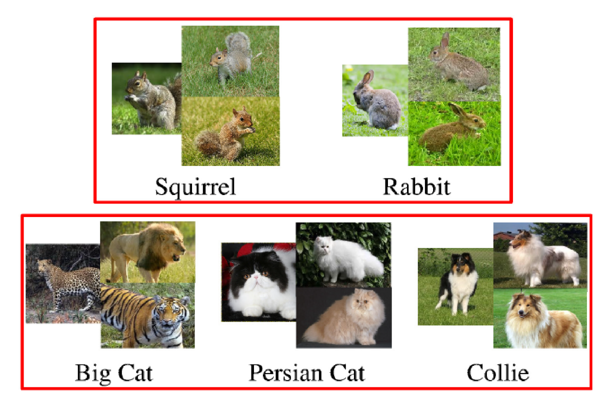


- Our method (AggkNN-L), compared with regular kNN (baseline), Non-linear SVM (NLSVM) (poly. kernel of deg. 9), Large Margin Nearest Neighbor (LMNN), and Taxonomy Embedding (TaxEmb).

Placing unseen categories in the taxonomy

- Given a taxonomy of 17 categories from Animals with Attribute dataset (solid lines), we can place new categories (dashed lines) by predicting the most likely parent.
- **Green lines** show correct placement, while **red lines** show incorrect placement.





Visual similarity between example classes