

Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost



Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka

Represented by Ahmad Mustofa HADI

Presentation Outline





































- ▶ Introduction
- ▶ Metric Learning Concept
- ▶ Methodology
- ▶ Experimental Evaluation
- ▶ Conclusion



Introduction

- ▶ The image and video available on net
- ▶ Image Annotation
- ▶ New Image in Dataset?



 <p>Cliff dwelling L2 11.0% - Mah. 99.9%</p>	 <p>horseshoe crab 0.99%</p>	 <p>African elephant 0.99%</p>	 <p>mongoose 0.94%</p>	 <p>Indian elephant 0.88%</p>	 <p>dingo 0.87%</p>	L2
 <p>Cliff dwelling L2 11.0% - Mah. 99.9%</p>	 <p>cliff 0.07%</p>	 <p>dam 0.00%</p>	 <p>stone wall 0.00%</p>	 <p>brick 0.00%</p>	 <p>castle 0.00%</p>	Mah.
 <p>Gondola L2 4.4% - Mah. 99.7%</p>	 <p>shopping cart 1.07%</p>	 <p>unicycle 0.84%</p>	 <p>covered wagon 0.83%</p>	 <p>garbage truck 0.79%</p>	 <p>forklift 0.78%</p>	L2
 <p>Gondola L2 4.4% - Mah. 99.7%</p>	 <p>dock 0.11%</p>	 <p>canoe 0.03%</p>	 <p>fishing rod 0.01%</p>	 <p>bridge 0.01%</p>	 <p>boathouse 0.01%</p>	Mah.
 <p>Palm L2 6.4% - Mah. 98.1%</p>	 <p>crane 0.87%</p>	 <p>stupa 0.83%</p>	 <p>roller coaster 0.79%</p>	 <p>bell cote 0.78%</p>	 <p>flagpole 0.75%</p>	L2
 <p>Palm L2 6.4% - Mah. 98.1%</p>	 <p>cabbage tree 0.81%</p>	 <p>pine 0.30%</p>	 <p>pandanus 0.14%</p>	 <p>iron tree 0.07%</p>	 <p>logwood 0.06%</p>	Mah.



Metric Learning Concept

- ▶ Metric Learning
 - ▶ Learning a distance function for particular task (Image Classification)
 - ▶ LMNN -> Large Margin Nearest Neighbor
 - ▶ LESS -> Lowest Error in a Sparse Subspace
- ▶ Transfer Learning
 - ▶ Method that share information across classes during learning
 - ▶ Zero Shot learning
 - ▶ a new class no training instance with a description is provided such as attributes or relation to seen classes.



Methodology

- ▶ Train Dataset with classifier method
- ▶ Obtain a classification model
- ▶ Test other dataset
- ▶ *Does it work for a new image who belongs to new class?*
 - ▶ *SVM ? Add new category, re-run your training step*
 - ▶ *Proposed Method? No need to re-run training step*



Methodology

- ▶ Metric Learning for **k-NN Classification**
- ▶ Metric Learning for **Nearest Class Mean Classifier**



Methodology

- ▶ Metric Learning for k-NN Classification
 - ▶ K-NN
 - ▶ a ranking problem which is reflected in LMNN
 - ▶ LMNN
 - ▶ the goal that the k-NN always belong to the same class while instances of different classes are separated by a large margin
 - ▶ SGD (Stochastic Gradient Descend)
 - ▶ Minimizing the LMNN function by computing gradient



Methodology

- ▶ Metric Learning for Nearest Class Mean Classifier (multi-class logistic regression)
 - ▶ Compute the probability of a class by given image using the mean of each class.
 - ▶ Compute the log-likelihood of ground truth class.
 - ▶ Minimize the likelihood function using Gradient



Experimental Evaluation

- ▶ Experimental Setup
- ▶ K-NN Metric Learning
- ▶ NCM Classifier Metric Learning
- ▶ Generalization to New Class



Experimental Evaluation

▶ Experimental Setup

▶ Dataset

- ▶ ILSVRC'10 (1,2M training image of 1,000 class)

▶ Features

- ▶ Fisher Vector of SIFT & Local Color Features
- ▶ PCA to 64 dimension
- ▶ Use 4K & 64K dimensional Feature Vector

▶ Evaluation Measure

- ▶ Flat Error : one if the ground truth does not correspond to top label with highest score, zero otherwise
- ▶ Top-1 and Top-5 Flat Error



Experimental Evaluation

- ▶ Experimental Setup
 - ▶ Baseline Approach
 - ▶ SVM (one-vs-rest)
 - ▶ SGD Training
 - ▶ To optimize the learning metric, projection matrix W is computed
 - ▶ SGD runs for 750K-4M iteration
 - ▶ Select lowest top-5 error



Experimental Evaluation

► K-NN Metric Learning

Table 1. k-NN classification performance with 4K dimensional features. For all methods, except those indicated by 'Full', the data is projected to a 128 dimensional space.

	k-NN classifiers							
	SVM	ℓ_2	ℓ_2	LMNN		All	Dynamic	
	Full	Full	+ PCA	10	20		10	20
Flat top-1 error	60.2	75.0	76.3	72.9	72.8	67.9	65.1	66.0
Flat top-5 error	38.2	55.9	57.3	50.6	50.4	44.2	39.8	40.7



Experimental Evaluation

► NCM Classifier Metric Learning

Table 2. Performance of k-NN and NCM classifiers, as well as baselines, using the 4K and 64K dimensional features, for various projection dimensions, see text for details.

Projection dim.	4K dimensional features							64K dimensional features			
	32	64	128	256	512	1024	Full	128	256	512	Full
SVM baseline							38.2				28.0
k-NN, dynamic 10	47.2	42.2	39.8	39.0	39.1	40.4					
NCM, learned metric	49.1	42.7	39.0	37.4	37.0	37.0		31.7	31.0	30.7	
NCM, PCA+ ℓ_2	78.7	74.6	71.7	69.9	68.8	68.2	68.0				63.2
NCM, PCA+inv.cov.	75.5	67.7	60.6	54.5	49.3	46.1	43.8				
PCA+Ridge-regress.	86.3	80.3	73.9	68.1	62.8	58.9	54.6				
WSABIE [7]	51.9	45.1	41.2	39.4	38.7	38.5		32.2	30.1	29.2	



Experimental Evaluation

▶ NCM Classifier Metric Learning

Table 4. Average per-class performance of the NCM classifier on the ImageNet-10K dataset, using metrics learned on the ILSVRC'10 dataset, and comparison to previously published results.

	4K dimensional features					64K dimensional features				21K	128K	128K
Proj. dim.	128	256	512	1024	SVM	128	256	512	SVM	[3]	[8]	[11]
Flat top-1	91.8	90.7	90.5	90.4	86.0	87.1	86.3	86.1	78.1	93.6	83.3	81.9
Flat top-5	80.7	78.9	78.6	78.6	72.4	71.7	70.5	70.1	60.9			



Experimental Evaluation

► Generalization to New Class

Table 3. Performance of 1,000-way classification among test images of 200 classes not used for metric learning, and control setting with metric learning using all classes. Left column denotes the number of training classes used, and “plain” denotes k-NN or NCM using the ℓ_2 distance.

	4K dimensional features									64K dimensional features				
	SVM	k-NN			NCM					SVM	NCM			
	Full	128	256	Full	128	256	512	1024	Full	Full	128	256	512	Full
Plain		54.2			66.6						61.9			
1000	37.6	39.1	38.4		38.6	36.8	36.4	36.5		27.7	31.7	30.8	30.6	
800		42.2	42.4		42.5	40.4	39.9	39.6			39.3	37.7	37.1	



Experimental Evaluation

► Generalization to New Class

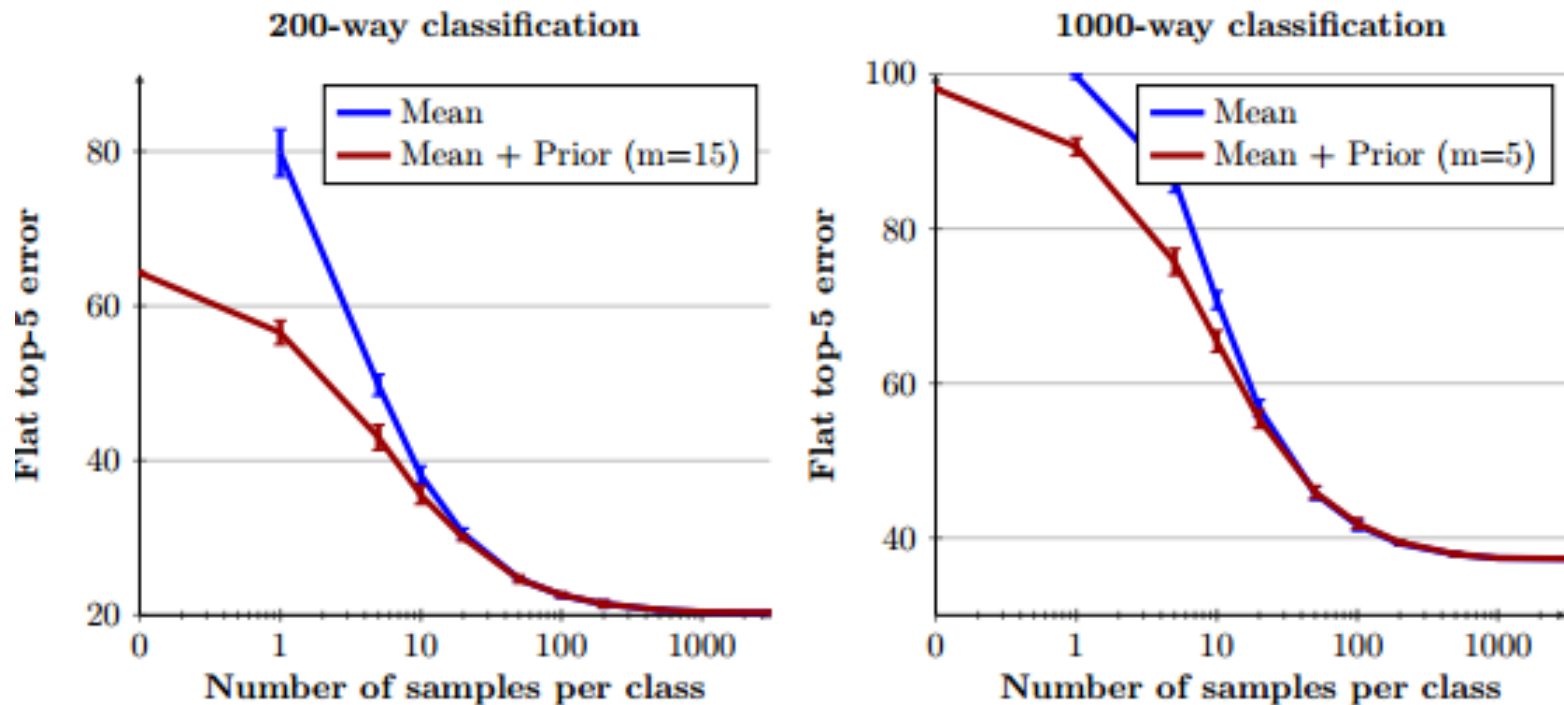


fig. 2. Performance of NCM as a function of the number of images used to compute the means or classes not used during training, with and without the zero-shot prior. See text for details.



Conclusion

- ▶ Metric Learning can be applied on large scale dynamic image dataset
- ▶ Zero cost to new classes can be achieved
- ▶ NCM outperforms k-NN
 - ▶ NCM is linear classifier
 - ▶ K-NN is highly non-linear and non parametric classifier
- ▶ NCM is comparable to SVM





learning approach of LMNN [16]. Their learning objective is based on triplets of images, where the distance between a query image q and a target image p of the same class should be smaller than the distance to a negative image n of a different class. The 0/1-loss for such a triplet is upper-bounded by the hinge-loss on the distance difference:

$$L_{qpn} = [1 + \|x_q - x_p\|_W^2 - \|x_q - x_n\|_W^2]_+, \quad (2)$$

which is zero if the negative image n is at least one distance unit farther from the query q than the positive image p , and positive otherwise. The sum of the per-triplet loss is the final learning criterion:

$$L = \sum_{q=1}^N \sum_{p \in P_q} \sum_{n \in N_q} L_{qpn}, \quad (3)$$

where P_q and N_q denote the set of positive and negative images for a query image x_q . The sub-gradient of the loss is obtained as:

$$\nabla_W L = \sum_{q=1}^N \sum_{p \in P_q} \sum_{n \in N_q} \nabla_W L_{qpn}, \quad (4)$$

$$\nabla_W L_{qpn} = [L_{qpn} > 0] 2W \left((x_q - x_p)(x_q - x_p)^\top - (x_q - x_n)(x_q - x_n)^\top \right), \quad (5)$$

where we use Iversons bracket notation $[\cdot]$ that equals one if its argument is true, and zero otherwise.



We formulate the NCM classifier using multi-class logistic regression and define the probability for a class c given an image feature vector x as:

$$p(c|x) = \frac{\exp - \|\mu_c - x\|_W^2}{\sum_{c'=1}^C \exp - \|\mu_{c'} - x\|_W^2}, \quad (6)$$

where μ_c is the mean for class $c \in \{1, \dots, C\}$. Our objective is to minimize the negative log-likelihood of the ground-truth class labels $y_i \in \{1, \dots, C\}$ of the training images:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \ln p(y_i|x_i). \quad (7)$$

The gradient of this objective function is easily verified to be:

$$\nabla_W \mathcal{L} = \frac{2}{N} \sum_{i=1}^N \sum_{c=1}^C \left(\mathbb{1}[y_i = c] - p(c|x) \right) W(\mu_c - x_i)(\mu_c - x_i)^\top. \quad (8)$$

To learn the projection matrix W , we use SGD training and sample at each iteration a fixed number of m training images to estimate the gradient.

Note that the NCM classifier is linear in x since we assign an image x to the class c^* with minimum distance:

$$c^* = \arg \min_c \{ \|x - \mu_c\|_W^2 \} = \arg \min_c \{ \|W\mu_c\|^2 - 2\mu_c^\top (W^\top W)x \}. \quad (9)$$

