Scalable training with approximate incremental laplacian eigenmaps and PCA

Conference Paper · October 2013

DOI: 10.1145/2502081.2508124

3 authors:

Eleni Mantziou
The Centre for Research and Technology, Hellas
5 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Symeon Papadopoulos
The Centre for Research and Technology, Hellas
260 PUBLICATIONS **4,909** CITATIONS
SEE PROFILE

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas
1,038 PUBLICATIONS **14,469** CITATIONS
SEE PROFILE

Scalable Training with Approximate Incremental Laplacian Eigenmaps and PCA

Eleni Mantziou
CERTH-ITI
Thessaloniki, Greece
lmantziou@iti.gr

Symeon Papadopoulos
CERTH-ITI
Thessaloniki, Greece
papadop@iti.gr

Yiannis Kompatsiaris
CERTH-ITI
Thessaloniki, Greece
ikom@iti.gr

ABSTRACT

The paper describes the approach, the experimental settings, and the results obtained by the proposed methodology at the ACM Yahoo! Multimedia Grand Challenge. Its main contribution is the use of fast and efficient features with a highly scalable semi-supervised learning approach, the Approximate Laplacian Eigenmaps (ALEs), and its extension, by computing the test set incrementally for learning concepts in time linear to the number of images (both labelled and unlabelled). A combination of two local visual features combined with the VLAD feature aggregation method and PCA is used to improve the efficiency and time complexity. Our methodology achieves somewhat better accuracy compared to the baseline (linear SVM) in small training sets, but improves the performance as the training data increase. Performing ALE fusion on a training set of 50K/concept resulted in a MiAP score of 0.4223, which was among the highest scores of the proposed approach.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Design Methodology

General Terms

Experimentation, Measurement, Performance, Algorithms

Keywords

Concept Detection, Laplacian Eigenmaps, Large-scale Annotation, Machine Learning

1. INTRODUCTION

This document describes our participation in the ACM Yahoo! Multimedia Grand Challenge 2013. The competition introduces a new challenging dataset that features 10 noisy concepts with $150K$ images per concept.

In recent years, the available online content constantly increases. In this context, semi-supervised learning is a promising approach for concept detection since it can benefit from the inclusion of unlabelled images in the training process. More specifically, *manifold learning* approaches rely on the assumption that there is an underlying image manifold, wherein semantically similar images are placed close to each other and semantically dissimilar images are positioned far from each other. Typically, *manifold learning* is implemented by means of constructing a similarity graph between labelled and unlabelled images and leveraging the graph to estimate the labels of the unlabelled images by considering the labels of neighbouring labelled images. Considering the beneficial results of using a manifold learning method in a concept detection problem, we choose to employ an approximate semi-supervised method for this challenge. However the manipulation of such large graphs is computationally costly, which is impractical for large datasets. Also, once new images are observed the graph must be updated.

Our approach, described in Section 2, tackles the scaling problem by constructing approximate eigenvectors based on the density structure of the data and by updating the eigenvectors of the test set incrementally. This is implemented with the use of the marginal distribution by building a density histogram to compute the eigenfunctions considering the limit as the number of points go to infinity. After computing the eigenfunctions, we interpolate them at each of the data points to extract the first k-approximate eigenvectors with the smallest eigenvalues. We will refer to this approach as Approximate Laplacian Eigenmaps (ALE) [1]. Extending the ALE, we devise an incremental method, in which once a new image is inserted, we interpolate its feature vector based on the eigenfunctions and eigenvalues of the training images. In this paper, we construct features that are easy to manipulate especially in large scale problems, by combining SIFT and RGB-SIFT with the VLAD feature aggregation.

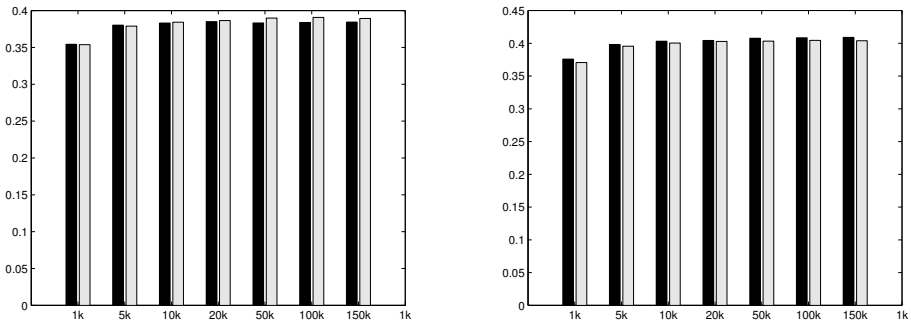
2. OVERVIEW OF METHOD

Our approach is based on Semi-Supervised Learning (SSL) by constructing Laplacian Eigenmaps (LEs) approximately and incrementally. We use the LEs as new features to train concept classifiers. SMaL is a **S**calable **Ma**nifold **L**earning framework [3] on top of ALE, which is linear to the number of images, making possible to use the graph Laplacian in large-scale problems. SMaL makes use of the Vectors of Locally Aggregated Descriptors (VLAD), and reduces their dimensionality by PCA.

VLAD [2] is a simplified non-probabilistic version of the Fisher Vector for feature aggregation, which uses a codebook of size μ_K computed using k-means. By applying nearest

MM'13, October 21–25, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2502081.2508124 .



(a) SIFT

(b) RGB-SIFT

Figure 1: Comparison of MiAP between SVM (**black**) and SMaL (**light gray**).

neighbour (NN) search, each local descriptor x_t is associated with its nearest centroid. Then, the differences between the descriptors x_t and the centroid μ_i are accumulated to u_i. The d-dimensional vector of each feature is concatenated with u_i constructing the Kd dimensional final VLAD vector.

As mentioned above, in order to construct the LEs, we need to build a $n \times n$ similarity matrix between labelled and unlabelled images (n is the number of labelled and unlabelled images together). A matrix like this is very costly to compute in large collections. In SMaL, we tackle this problem based on the approximate computation of LEs by estimating a smaller covariance matrix, as suggested in [1], where it is hypothesized that the data $x_i \in \Re^d$ are samples from a distribution $p(x)$. Rotating the data to be as independent as possible, $s = Rx$, can result in a $B \times B$ histogram of bins, using only marginal distributions that approximate the density $p(s)$ of the rotated data.

Then, instead of computing the eigenvectors of the similarity matrix between the original data ($n \times n$), one can define eigenfunctions g corresponding to the eigenvalues of the rotated data s ($B \times B$), which can be seen as approximations of the LEs of the original data when $n \to \infty$. This is considerably faster, since typically $B \ll n$. These are recovered by solving the following equation:

$$\left(\tilde{D} - P\tilde{W}P\right) g = \sigma P\hat{D}g \tag{1}$$

where $\tilde{W}$ is the affinity between the B discrete points, P is a diagonal matrix whose diagonal elements give the density at the discrete points, $\tilde{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}P$, and $\hat{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}$. An interpolation step follows to the target dimension C_D (described in [1]) and in the end, the $n \times C_D$ approximate LE vectors are derived.

To tackle the Yahoo! challenge which does not allow the use of test data during training, we devised an online implementation by computing the eigenvectors of unlabelled data incrementally. We build the $B \times B$ matrix based on the histogram bins of labelled data. From this matrix the eigenfunctions and the eigenvalues of the eigenfunctions are derived. We use these values to interpolate both the labelled and unlabelled data separately.

In the final step, the LEs are used to define a smoothness operator that takes into account the unlabelled data. The key idea is to find functions f, which agree with the labelled

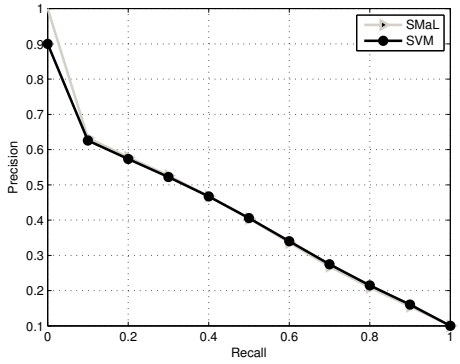


Figure 2: Precision-Recall curve at 50K/Concept between SMaL and Linear SVM, when SIFT & RGB-SIFT are fused.

data and are also smooth with respect to the graph. To avoid the trivial solution $f = 1$, a combination of smoothness vector and the training error loss is minimized [1]. The smoothness of an eigenvector ϕ has the form

$$f = \Sigma_i \alpha \phi_i \tag{2}$$

Thus, smooth vectors are linear combinations of the eigenvectors with small eigenvalues. In our implementation, we define f as $f = U\alpha$. Thus, the minimization problem reduces to the minimization of α, defined as

$$(\Sigma + U^T \Lambda U)a = U^T \Lambda y \tag{3}$$

where Σ are the smallest eigenvalues, U is the $n \times C_D$ approximate LE vectors, Λ is a diagonal matrix, whose diagonal elements are $\Lambda_{ii} = \lambda$ if i is labelled, otherwise $\lambda = 0$, and y are the labels.

3. EXPERIMENTAL SETUP

Training Set Creation: As suggested by the organizers, we evaluate our results on 1K, 5K, 10K, 20K, 50K, 100K and 150K training images per concept. For better consistency of the results, we take random sets and use 10 repetitions at each split (except 150K/concept since it uses all the training data and cannot be split).

Feature Extraction: We have used the SIFT ($d = 128$) and RGB-SIFT ($d = 384$) local descriptors [6] as features.

SMaL			SVM		
	Train	*Test*		*Train*	*test*
	$B \times B$, g, U	f		*Model*	*Prediction*
1K	5.3 sec	10 mins	1K	23 sec	2.5 sec
50K	59 sec	10 mins	50K	19 mins	2.5 sec
150K	3 mins	10 mins	150K	71 mins	2.5 sec

Table 1: Time Trade-Off between SMaL and linear SVM in three different splits. Times refer to one repetition

We use a dense regular grid with a spacing of 6 pixels and perform K-means clustering with a vocabulary size of $K = 64$ centroids for better performance as proposed in [2]. The clustering was performed on an independent set of 10,000 images, randomly sampled from the MIRFLICKR-1M dataset [4]. The final VLAD vectors are power-and L2-normalized and then reduced from $D = Kd$ to $D' = 512$ and L2-normalized again [7]. The choice of D only marginally affects accuracy as observed experimentally. We tested the BoW features, provided by the organizers, with our implementation, but with inferior results. Probably, BoW would also need to be reduced by PCA to perform well with SMaL due to the hypothesis that the data must be as separate as possible [1]. However, we could not apply PCA on BoW, since we would need to extract BoW features for an independent set of images (we did not have access to the BoW extraction software).

SMaL Optimization: In ALE, no variable needs optimization, since it was observed that different values of B and C_D did not affect accuracy. Thus, we choose to set $B = 50$ for computational efficiency reasons. For the number of eigenvectors and the Smooth Functions the best results were obtained with $C_D = 500$ and $\Lambda = 100$ respectively. All values were experimentally defined using the ImageCLEF 2012 dataset [5].

Incremental Learning: Our approach performs incremental learning in batches. We assume that the unlabelled data come in batches (e.g. 1000 images). We chose to include 1000 images in a batch (instead of one) for time efficiency reasons, making sure that the accuracy was unaffected. At first, the algorithm builds the histogram of the selected training set. From this histogram, we derive the $B \times B$ matrix, the eigenfunctions (g) and the eigenvalues of the eigenfunctions (σ). After computing the g and σ, we interpolate g at each of the training data to extract U_{train} and Σ. The $B \times B$ and g, are then used to compute the eigenvectors of the unlabelled data (U_{test_i}) for each batch, by interpolating g of the training set to the batch data. U_{train} and U_{test} are then concatenated to produce U.

To measure accuracy, we use the Mean interpolated Average Precision (MiAP) measure as proposed by the organizers. The MiAP scores reported in Section 4 are the average scores from 10 repetitions at each split. Finally, as baseline a linear SVM was trained using the same local descriptors as input. A linear SVM was chosen as a smore competitive method than a kNN algorithm, since it typically achieves better results in concept detection problems. Also, we could not use a non-linear SVM (i.e. RBF) due to the excessive computational cost.

All experiments were coded in Matlab and executed on an 8-core (Intel Quad Core i7-950 @3.07Ghz, 12G RAM) and a 24-core (an Intel Xeon Q6600 @2.0Ghz, 128G RAM) machine. The reported execution times (Table 1) were obtained from measurements on the 24-core machine.

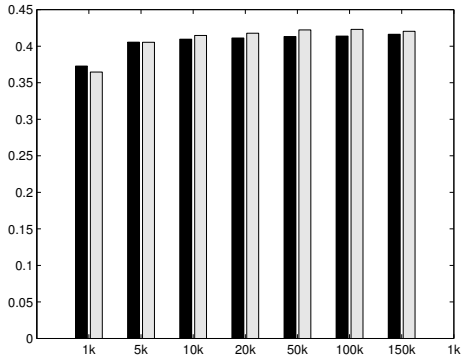


Figure 3: Comparison of MiAP between SVM (**black**) and SMaL (**light gray**) in case of fusion.

4. DESCRIPTION OF RESULTS

In this section, we provide the experimental results of our submission. As mentioned above, our method is benchmarked against a linear SVM classifier. Figure 1 presents the MiAP of these two methods, as measured using the SIFT and RGB-SIFT respectively. When using SIFT, it is observed that as the training set increases, SMaL performs better than the linear SVM. For instance, if 1K/concept are used, SMaL yields a MiAP score of 0.3537, while the linear SVM yields a score of 0.3543, but if 10K/concept are used, MiAP in SMaL is 0.3842, instead of linear SVM which is 0.3831.

The main observation is that the execution time of SMaL remains almost unaffected as the training set increases as illustrated in Table 1. For example, if a 50K/concept training set is used, SMaL needs 59 secs for training and 10 mins to predict the 500K test set in batches, while linear SVM needs 19 mins for training and 2.5 secs for prediction. Moreover, if 150K/concept is used SMaL needs 3 mins to compute the training variables ($B \times B$, g and U) and 10 mins for prediction, while linear SVM needs about 71 mins to learn the model and 2.5 secs for prediction. This capability of SMaL, gives us the advantage of using as many training images as possible, since the computation of eigenvectors is linear to the number of images and the computation of each batch is consistent in each split. On the other hand, we report that SMaL does not perform as well when the RGB-SIFT descriptor is used.

Table 2 depicts the Mean interpolated Precision (MiP) for each concept in the smallest (1K) and the biggest split (150K). Generally, we hypothesize that some concepts like *Sky* are more visually coherent, while some others, like *Nature* or *2012*, are more noisy. According to this hypothesis, we observe that SMaL can predict better than linear SVM the more noisy concepts. More specifically, in the $1K$ set in the *Nature* concept SMaL performs 0.3535, while SVM 0.3387. Another example is the concept *2012*, where the MiP of SMaL is 0.1967 and of the linear SVM is 0.1849. These differences increase as we use more training examples. In $150K$, in the concept *Nature* SMaL performs 0.3987, while the linear SVM yields a MiP of 0.3674. For the concept *2012* the MiP in SMaL is 0.2572 and in linear SVM it is 0.2246. The best performance for both methods is attained for the concept *Sky*, which is much more coherent than the other

SIFT				
Concept	#	$1K$	$150K$	**Diff**
Nature	SMaL	0.3535	0.3987	0.0452
	SVM	0.3387	0.3674	0.0287
Food	SMaL	0.5122	0.5460	0.0338
	SVM	0.5146	0.5499	0.0352
People	SMaL	0.2146	0.2623	0.0477
	SVM	0.2189	0.2605	0.0416
Wedding	SMaL	0.3925	0.4319	0.0394
	SVM	0.3993	0.4436	0.0442
Music	SMaL	0.4064	0.4406	0.0342
	SVM	0.4078	0.4342	0.0264
Sky	SMaL	0.5856	0.6101	0.0245
	SVM	0.5943	0.6235	0.0292
London	SMaL	0.2686	0.2961	0.0275
	SVM	0.2710	0.2903	0.0193
Beach	SMaL	0.4118	0.4392	0.0274
	SVM	0.4154	0.4430	0.0276
2012	SMaL	0.1967	0.2572	0.0605
	SVM	0.1849	0.2246	0.0397
Travel	SMaL	0.1947	0.2104	0.0157
	SVM	0.1981	0.2070	0.0090

RGB-SIFT				
Concept	#	$1K$	$150K$	**Diff**
Nature	SMaL	0.3897	0.4234	0.0337
	SVM	0.3812	0.4313	0.0501
Food	SMaL	0.5492	0.5786	0.0293
	SVM	0.5527	0.5785	0.0258
People	SMaL	0.2034	0.2301	0.0268
	SVM	0.2123	0.2644	0.0520
Wedding	SMaL	0.4115	0.4507	0.0392
	SVM	0.4181	0.4563	0.0383
Music	SMaL	0.4239	0.4557	0.0319
	SVM	0.4271	0.4617	0.0346
Sky	SMaL	0.6143	0.6376	0.0232
	SVM	0.6236	0.6501	0.0266
London	SMaL	0.2665	0.3071	0.0406
	SVM	0.2775	0.3059	0.0284
Beach	SMaL	0.4404	0.4656	0.0252
	SVM	0.4436	0.4735	0.0300
2012	SMaL	0.2073	0.2616	0.0543
	SVM	0.2136	0.2564	0.0428
Travel	SMaL	0.2001	0.2299	0.0298
	SVM	0.2094	0.2121	0.0027

Table 2: Comparison of MiP scores between $1K$ and $150K$.

concepts. On the other hand, one of the most difficult concepts proves to be the *Travel* concept, which in most cases has a MiP under 0.20. However, RGB-SIFT seems to yield good performance in this concept at 150K, as depicted in Table 2.

Fusion: Within SMaL, fusion takes place at the level of the Laplacian Eigenmap vectors. As a result, we obtained improved efficiency and better results than SVM. As illustrated in Figure 3, SVM performs better than SMaL in 1K, but as the training set increases, SMaL improves its accuracy. In 50K, MiAP in SMaL is 0.4223, while in SVM it is 0.4133. Figure 2 illustrates the performance comparison between SMaL and SVM in terms of Precision-Recall curve. Figure 4 illustrates the top 18 results of the test set (ranked by prediction score) for four concepts of the GC dataset.

5. CONCLUSIONS

In this paper, we proposed an approximate incremental semi-supervised learning approach, leveraging VLAD vectors, with the goal of learning very general concepts with high noise. Our framework significantly decreases the computational requirements of training in view of large amounts of training data, while improving the performance. Our experiments demonstrate that the proposed framework gives slightly better results compared to the baseline method, while achieving large computational gains. In the future, we plan to further investigate the behaviour of our method to better manage the noisy labels.

Figure 4: Top 18 images for the concepts: Nature, Food, Beach and Travel (top-to-bottom) on a 150K training set: SVM (left), SMaL (right).

6. ACKNOWLEDGMENTS

This work was supported by the SocialSensor project, partially funded by the European Commission, under the contract number FP7-287975.

7. REFERENCES

[1] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *NIPS*, pages 522–530. 2009.

[2] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conf. on, CVPR*, pages 3304–3311, 2010.

[3] E. Mantziou, S. Papadopoulos, and I. Kompatsiaris. Large-scale semi-supervised learning by approximate laplacian eigenmaps, VLAD and pyramids. In *WIAMIS*, 2013.

[4] B. T. Mark J. Huiskes and M. S. Lew. New trends and ideas in visual concept detection: The MIR FLICKR retrieval evaluation initiative. In *MIR '10*, pages 527–536, New York, NY, USA, 2010. ACM.

[5] B. Thomee and A. Popescu. Overview of the clef 2012 flickr photo annotation and retrieval task. in the working notes for the CLEF 2012 labs/workshop. Rome, Italy, 2012.

[6] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Empowering visual categorization with the GPU. *IEEE Transactions on Multimedia*, 13(1):60–70, 2011.

[7] E. S. Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, and I. P. Vlahavas. An empirical study on the combination of SURF features with vlad vectors for image search. In *WIAMIS*, 2012.