Non-negative Matrix Factorisation for Object Class Discovery and Image Auto-annotation

Jiayu Tang
Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ
United Kingdom
jt04r@ecs.soton.ac.uk

Paul H. Lewis
Intelligence, Agents, Multimedia Group
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ
United Kingdom
phl@ecs.soton.ac.uk

ABSTRACT

In information retrieval, sub-space techniques are usually used to reveal the latent semantic structure of a data-set by projecting it to a low dimensional space. Non-negative matrix factorisation (NMF), which generates a non-negative representation of data through matrix decomposition, is one such technique. It is different from other similar techniques, such as singular vector decomposition (SVD), in its non-negativity constraints which lead to its parts-based representation characteristic. In this paper, we present the novel use of NMF in two tasks; object class detection and automatic annotation of images. Experimental results imply that NMF is a promising sub-space technique for discovering the latent structure of image data-sets, with the ability of encoding the latent topics that correspond to object classes in the basis vectors generated.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous ; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

experimentation, algorithms, measurement

Keywords

Automatic Image Annotation, Non-negative Matrix Factorisation, Object Detection

1. INTRODUCTION

The vector space model (VSM), which represents a collection of documents by a term-by-document matrix, has been a major and popular model in information retrieval, including traditional content-based image retrieval and more recently automatic image annotation. Each column of the matrix represents a document and each item of that column represents the occurrence of a particular term. As for images, each column of the matrix corresponds to an image and each item of the column indicates the number of times a certain visual term appears in the image. Visual terms have been chosen in many forms, for example 'blobs' [1], quantised salient regions [2], and even single pixels [3].

Usually the term-by-document matrices are of high dimension and noisy, which makes it difficult to capture the underlaying semantic structure. Dimensionality reduction or sub-space techniques, e.g. SVD, have been developed to reduce the dimensionality, filter noise, and discover the latent semantic structure. Recently, [3] proposed to use a matrix decomposition technique called non-negative matrix factorisation (NMF). It is distinguished from SVD by its non-negativity constraints, which leads to its unique feature - parts-based representations of documents. They have shown that NMF is able to learn basis images that correspond to face parts, such as mouth, nose and eyes. NMF has been applied for text document retrieval [4, 5], image patch classification [6], and object recognition [7].

In this paper, we explore the use of NMF for images in two different scenarios, object class detection and automatic image annotation. Firstly, we utilise the parts-based representation characteristic of NMF to find object classes from a collection of un-annotated images. Secondly, we investigate the use of NMF as an alternative sub-space technique for image auto-annotation via semantic propagation.

2. NON-NEGATIVE MATRIX FACTORISATION

2.1 Classic NMF

NMF is a technique to find a representation of non-negative data using non-negativity constraints. Under such constraints only additive, not subtractive, combinations are allowed, which lead to a parts-based representation of the original data. Given an $n \times m$ term-by-document matrix V (where each column is a document and each row represents a keyword) with $V_{ij} \geq 0$ and a pre-defined positive integer r, NMF finds two non-negative matrices $W \in R^{n \times r}$ and $H \in R^{r \times m}$ such that $V \approx WH$. The rank r is generally chosen as smaller than n and m, for example $(n+m)r < nm$. NMF approximates each column of V by a linear combination of r column vectors in W. In other words, each column of

CIVR'08, July 7–9, 2008, Niagara Falls, Ontario, Canada.
Copyright 2008 ACM 978-1-60558-070-8/08/07 ...$5.00.

W is regarded as a “basis” vector, while each column of H contains the corresponding weights needed for the approximation. Details of NMF can be found in [3, 8].

2.2 NMF with Sparseness Constraints

Hoyer [9] noticed one of the most useful properties of NMF is that it generates a sparse representation of data. Much of the data is encoded in such a representation using only a few ‘active’ components. This notion is in line with the initial interpretation of NMF that parts are combined to build a whole. It is argued that in some applications, NMF does not lead to parts-based representations because the decomposed matrices are not ‘sparse’ enough. Therefore, sparseness constraints are applied to the objective function, in order to achieve a pre-defined level of sparseness of the decomposition. They proved that in their experiment the parts-based representation of data can be enhanced through this approach.

“The concept of ‘sparse coding’ refers to a representational scheme where only a few units (out of a large population) are effectively used to represent typical data vectors” [9]. In other words, most units take values that are close to zero and only a few take large non-zero values. Hoyer defines the sparseness of a vector x as follows, which is based on the relationship between L_1 norm and L_2 norm:

$$sparseness(x) = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1} \tag{1}$$

where n is the dimensionality of x. This function evaluates the sparseness of a vector to a value within the range of [0,1]. The sparseness equals 1 if and only if x contains only one single non-zero component; it equals 0 if and only if all components of x are equal.

NMF with sparseness constraints is then defined as follows. Given a non-negative matrix V, find the non-negative matrices W and H such that $E(W,H) = ||V - WH||^2$ is minimized, under optional constraints

$$\begin{aligned} sparseness(w_i) = S_w, \forall i, \\ sparseness(h_i) = S_h, \forall i, \end{aligned} \tag{2}$$

where w_i is the ith column of W and h_i is the ith row of H. S_w and S_h are the desired sparsenesses of W and H respectively, and are set by the user.

2.3 NMF vs. PLSA

Researchers have pointed out the similarities between NMF and the technique PLSA (probabilistic latent semantic analysis). For example, [10] showed that PLSA solves the problem of NMF with KL divergence, and that the local fixed point solutions found by NMF and PLSA are the same. However, [11] claimed that the proof of [10] is incorrect. They argued that NMF and PLSA are different algorithms, and showed that NMF and PLSA converge to different solutions using even the same initial conditions. They also reported that in their experiments, the results achieved by a hybrid NMF-PLSI algorithm are better than either of the techniques. Whilst there obviously is a connection between NMF and PLSA, they tackle problems from different perspectives. PLSA models data from a statistical viewpoint, and using Maximum Likelihood estimation to find the approximation of data. NMF deals with data from a sub-space viewpoint, and can theoretically use various objective functions to approximate the decomposition of a matrix.

3. DISCOVER OBJECT CLASS AND EXTENT

3.1 Saliency Based Visual Term Representation

Salient interest points and regions have been shown to outperform global image descriptors in terms of content-based image retrieval [12] performance. In our algorithm, we select salient regions by using the method proposed by Lowe [13], in which scale-space peaks are detected in a multi-scale difference-of-Gaussian pyramid. In addition, Lowe’s SIFT (Scale Invariant Feature Transform) descriptor [13] is used as the feature descriptor. The visual term representation is generated as follows.

1. Salient regions are discovered by using Lowe’s saliency detection technique on each image.

2. For each salient region, the SIFT feature descriptor is calculated.

3. Salient descriptors of all the images are quantised by k-means clustering algorithm into clusters, each of which is regarded as corresponding to a visual term or visual word. A salient region is then represented by a visual term indicating its membership of a cluster.

4. Finally, each image can be described as a histogram/ vector of visual terms, indicating the number of occurrences of each term in the image.

This form of description is analogous to the way in which a set of text words constitute a text document. Here, each quantised salient descriptor is considered as a word, and each image is a document.

3.2 NMF for Object Class Detection

Inspired by the work of [3] in which NMF was used to find the parts that form the whole faces, we explore its application to general-purpose images, say natural outdoor photos. The idea is straightforward - we consider outdoor images as analogous to face images at the whole image level, then the objects (e.g. sky, water, tree, etc.) that constitute the outdoor images are analogous to the face parts (i.e. mouth, eye, nose, etc.) at the objects level. As the basis generated by NMF on the face images correspond to face parts, it is possible that the basis generated on outdoor images will correspond to natural objects.

The problem to be explored here can be formalised as follows. Given a collection of un-annotated images, is it possible to learn the object classes simply from their appearances? An object class is a group of objects which may differ slightly from each other visually but correspond to the same semantic concept, e.g. the object class of ‘buildings’. We propose to answer the question in two main steps. Firstly, use NMF to find the bases which are expected to correspond to objects. Secondly, rank all the image segments that are generated by an automatic segmentation method, according to the distances to each bases object to see if the basis actually represent different object classes.

[3] used the grey level pixel values of the face images to construct the term-by-document matrix. Each column is a face image and each element in the column corresponds to a pixel. Since the resolutions of the images used were 19×19 in their case, it does not cause a problem when all the pixels

are used. However, it can result in a very large matrix with general-purpose images which are often of high resolution. Resizing the images will lose a lot of useful information, and make them hardly recognisable at a resolution level as low as 19×19. Therefore, another manner of image representation has to be developed. We choose the visual term representation described in Section 3.1. All the vectors from step 4 are arranged as columns to form a matrix. Suppose the image collection is I_i ($i = 1, 2, ..., m$, where m is the total number of images), mathematically we now have a $n \times m$ term-document matrix V, where V_{ij} is the occurrence of the ith visual term in image I_j, and n is the size of the visual vocabulary.

Since all the elements in the term-by-document matrix are non-negative, we can now apply NMF to it. We adopted the projected gradient method[1] for NMF that is developed by [14], because it converges faster than the popular multiplicative update approach [3]. NMF decomposes the term-by-document matrix V into W and H where $V \approx WH$. If W is represented by its column vectors as $W = [W_1, W_2, ..., W_r]$ (r is the number of basis vectors, or the dimensionality of the subspace), W_i is considered as a basis vector, or a conceptual part/object of the image collection. Each element of a basis vector indicates how many times a particular visual term appears in this conceptual object. In order to demonstrate if the basis vectors (i.e. W_i) actually correspond to object classes, the following approach is chosen. We use Normalized Cuts [15], which is an automatic image segmentation algorithm, to divide each image into regions. The visual terms within a specific region are used to form a vector representing the region. As a result, all the segments from the data-set can be represented by vectors I^t ($t = 1, 2, ..., M$, where M is the total number of segments in the data-set). For each basis vector, we rank all the image segments according to the distance, which is calculated as the cosine value of the angle between the two vectors, $cos(W_i, I^t)$[2]. The top ranked segments are examined to see if they represent an object class, which corresponds to the basis vector.

3.3 Experimental Results and Discussion

For comparison purposes, we choose the same data-set as used by [16] for our experiments. The data-set is a subset of a large image database named LabelMe [16]. The subset has 1554 images which are returned by querying the LabelMe data-set with words "cars", "trees" and "buildings". The images also contain many other additional objects. Most of the images have a resolution of 640×480. The size of the visual vocabulary is set to 2000, which results in a term-by-document matrix of 2000×1554. Finding the optimal value of r, the dimensionality of sub-space or the number of basis vectors, is a difficult problem in itself. In this work, we set the value empirically to 35.. In terms of segmentation, we followed the setting of the work by [16]. Specifically, to produce multiple segmentations, we varied two parameters of Normalised Cuts: the number of segments K and the size of the input image. K is set to $3, 5, 7, 9, 11, 13$ and the segmentation algorithm is applied at 2 image scales: 50- and 100-pixels across. This results in 12 different levels of segmentation per image. For each basis vector, segments from all segmentation levels are ranked according to their distances to the basis vector. Figure 1 shows montages of segments for the object classes found by NMF, each group corresponding to a basis vector of W. Each of the depicted segments comes from a different image to avoid showing multiple segments of the same object from the same image. As can be seen, NMF manages to find some object classes (e.g. "trees", "sky", "buildings", etc.) fairly well. However, it was not successful on "cars" in our experiments.

Methods	buildings	cars	roads	sky
Our Methods				
Multi Seg NMF	0.69	0.11	0.39	0.83
Sing Seg NMF	0.50	0.09	0.47	0.67
Russell et al.'s Methods				
Multi Seg LDA	0.53	0.21	0.41	0.77
Multi Seg pLSA	0.59	0.09	0.16	0.77
Sing Seg LDA	0.55	0.29	0.32	0.65

Table 1: Segmentation accuracy of the top 20 segments returned by NMF on four object classes from the LabelMe dataset. It is compared with the results from Russell et al. (2006) on the same data-set.

We also conducted an evaluation on the effect of using multiple segmentations to see if more accurate segments can be found. Segmentation accuracy, which is a metric used by [16], is calculated on the top ranked LabelMe segments for each object class. The top twenty returned images for four object classes are tested: "buildings", "cars", "roads" and "sky". They are compared with the ground truth object segmentation that was generated manually. Suppose R and GT denote the set of pixels in the retrieved object segment and the ground truth segmentations of the object. The accuracy score ρ measures how correct the area segmented by the retrieved object segment is. It is calculated as the ratio of the intersection of GT and R to the union of GT and R, as follows

$$\rho = \frac{GT \cap R}{GT \cup R}$$

If more than one ground truth segment intersects with R, we choose the one achieving the highest score. The accuracy score is averaged over the top 20 returned object segments for the four classes. The results are shown in Table 1. The table also included the results of using single segmentation, for comparison with multiple segmentations. As can be seen, NMF performs as well as, if not better than, the methods proposed by [16]. In particular, NMF with multiple segmentations ourperforms LDA (Latent Dirichlet Allocation) with multiple segmentations by 0.16 on "buildings" and 0.06 on "sky", altough 0.02 and 0.10 worse on "roads" and "cars" respectively.

It is interesting to note the difference between our approach and that of [16]. Although both methods use quantised descriptors of salient regions as visual terms, we treat each entire image as a document, while Russell et al. treat each image segment as a document. We build a term-document matrix and then rely on NMF to find the basis vectors, or underlying "topics" as called by Russell et al. They apply statistical models to the whole set of image segments to find the "topics". Therefore, the data that needs to be processed is less in our approach.

[1] Code available at: http://www.csie.ntu.edu.tw/∼cjlin/nmf/index.html

[2] For vectors V_1 and V_2, $cos(V_1, V_2) = \dfrac{V_1 \cdot V_2}{|V_1||V_2|}$

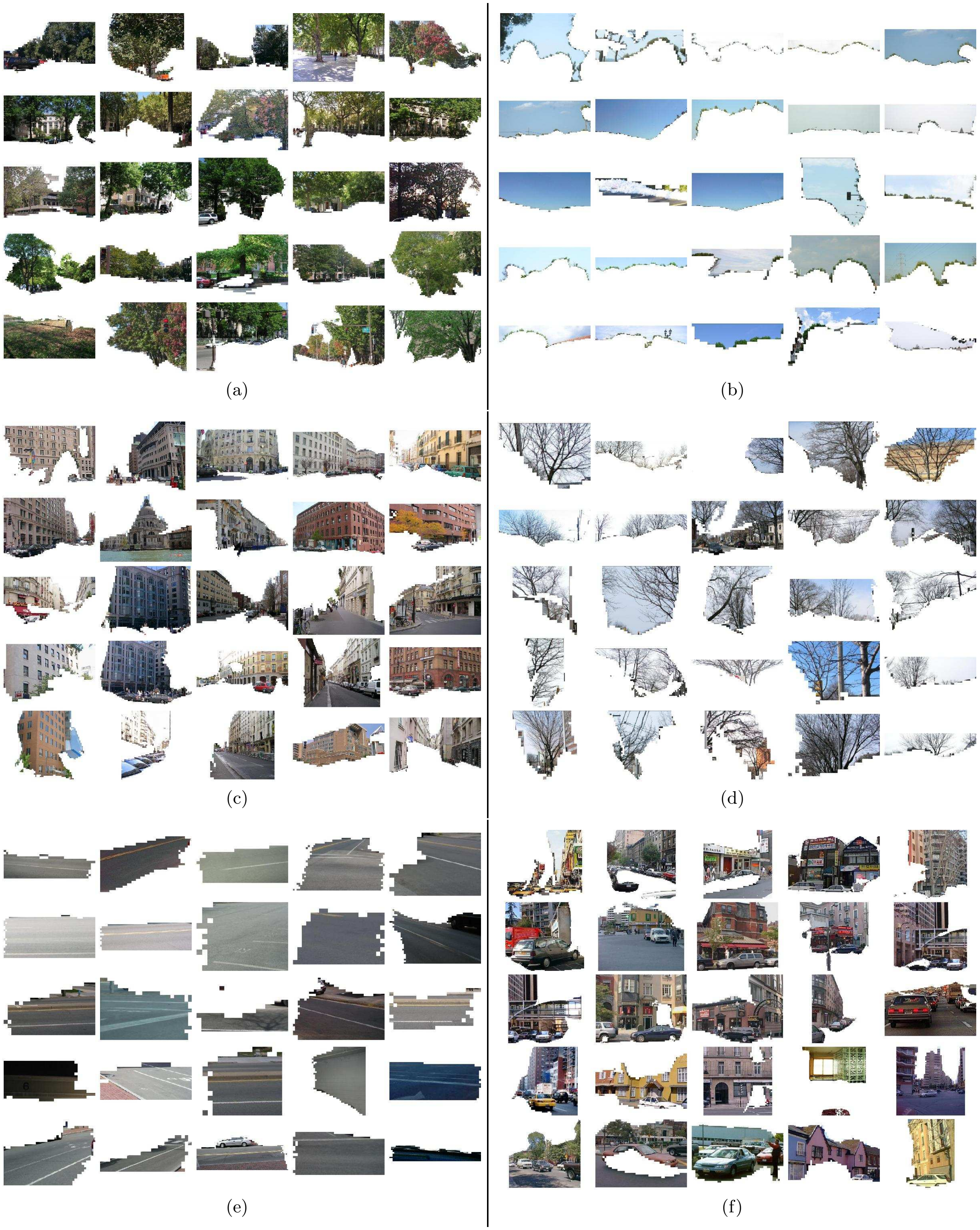


Figure 1: Top segments for 6 (out of 35) object classes discovered in the LabelMe data-set. Note how the segments, learned from a collection of unlabeled images, correspond to trees (a), sky (b), buildings (c), leafless trees (d), roads (e). However, for the last group of segments (f), it is not obvious which class of objects it corresponds to. We consider it as the class of cars in our evaluations.

4. AUTO-ANNOTATION VIA SEMANTIC PROPAGATION IN SUB-SPACE

The technique latent semantic indexing (LSI) was proposed by [17] to perform document clustering. They demonstrated that it is possible to reveal the implicit higher-order structure in the association of terms with documents, by projecting the term by document matrix into a sub-space through singular value decomposition (SVD). SVD is a popular matrix decomposition technique that decomposes a $m \times n$ matrix A into the product of a $m \times r$ matrix T, a $r \times r$ matrix S, and a $r \times n$ matrix D: $A = TSD^T$ such that $TT^T = D^T D = I$, where I is the identity matrix. Therefore, T and D^T are orthogonal matrices. S is a diagonal matrix in which diagonal elements are called singular values of matrix A, in monotonically decreasing order. It is claimed that the k largest singular values together with the corresponding left and right eigenvectors encode the most important information of A [17]. Hare and Lewis [2] used this technique (SVD for LSI) for automatic image annotation via sementic propagation. The premise of their approach is based on the intuition that visually similar images often have similar meaning or semantics. NMF as another matrix factorisation technique can be used as an alternative to SVD in order to project high dimensional data to a low dimensional sub-space, in which the semantics of data is expected to be more explicit. In this section, we will examine the use of NMF for image auto-annotation via sementic propagation.

4.1 NMF for Sub-space Projection

Given a document corpus, or a collection of images, we assume that it consists of k topics. Note that we use the same terminology "k" as used in SVD where it refers to the k largest singular values. Each document in the corpus either completely belongs to a particular topic, or is partially related to several topics. Ideally, if the documents can be projected into a k dimensional semantic space in which each axis corresponds to one of the k topics, the semantic structure of the data-set will be more explicit. In other words, each document can be represented by a linear combination of the k topics. Because it is more natural to consider each document as an additive instead of subtractive mixture of different topics, the coefficients of the linear combination should be all non-negative. Moreover, it is usually the case that topics of a corpus are not completely independent of each other. Overlaps may exist among the topics. Therefore, the axes of the semantic space that capture the topics are not necessarily orthogonal, which is the case for the sub-space generated by SVD.

NMF is theoretically superior to SVD for the following reasons. First, overlaps exist among topics. The orthogonal restriction by SVD makes the derived latent semantic directions less likely to correspond to each topic. Second, NMF decomposes the matrix in such a way that each document can be considered as an additive combination of topics, which makes more sense in the image domain. Third, for a particular document, the coefficients of the linear combination in NMF give direct indications of to what extent this document belongs to each of the topics. In contrast, SVD can not give this advantage because those negative values do not have intuitive interpretations.

Based on the above theory, we propose to utilize NMF to find the latent semantic structure for a collection of images, and then use a semantic propagation method to annotate images automatically.

4.2 Semantic Propagation based Auto-annotation

Semantic propagation is perhaps the simplest automatic image annotation method. The basic idea is intuitive; images that have similar visual appearance should have similar semantics. For a given new un-annotated image, a CBIR-like process is carried out first in order to rank the training images which are already annotated. Then, labels are chosen from the top (most similar) training images to annotate the new image. Therefore, most of the traditional CBIR techniques can be directly transfered to image auto-annotation applications in the manner described above. For example, [2] search for visually similar images in the semantic space that is generated by applying SVD to the term-by-document matrix of an image collection, and then propagate the labels from the top ranked images (1, 2 and 3 respectively) to a new query image.

We choose the same approach as that of [2], except that NMF is used to find the latent semantic topics. The whole process is conducted as follows.

1. The visual term representation (Section 3.1) of training images are calculated and used to build the term-by-document matrix V. NMF is applied on V to generate W and H such that $V \approx WH$.

2. Each query image q is projected into the semantic space spanned by W. Because we assume that the query image shares the same latent semantic structure as the training set, equation $q = Wh_q$ stands, where h_q is the new coordinates of q. h_q can be calculated as $h_q = W^{-1}q$.

3. Training images are ranked according to their distances to the query image in the space of W. In other words, we compare each column of H with h_q. Cosine distance of vectors is used in this work.

4. Labels of the top M training images are propagated to the new image as its predicted labels.

4.3 Experiment and Results

4.3.1 *The Washington Image Data-set*

For comparison, the same data-set as used by [2] is chosen for experiments, namely the University of Washington Ground Truth Data-set[1]. The Washington data-set contains 697 public-domain images, each of which has been semantically annotated with between 1 and 13 keywords. For example, an image may have several labels describing its content, such as "trees", "buildings", "sky", etc. On average there are 4.8 keywords per image. After the original keyword labels were processed by correcting mistakes and merging plurals into singular forms [2], e.g. "trees" became "tree", the vocabulary consisted of 170 keywords.

4.3.2 *Performance Evaluation*

For each test image, precision and recall, as well as the *normalised score* proposed by Barnard *et al* [18], are calculated for performance evaluation. Each kind of metric is

[1] http://www.cs.washington.edu/research/imagedatabase/groundtruth/

averaged over the entire test set to get a mean value. The definitions of these metrics are as follows.

$$\begin{aligned} Recall &= r/n \\ Precision &= r/(r+w) \\ E_{ns} &= \tfrac{r}{n} - \tfrac{w}{N-n} \end{aligned} \quad (3)$$

where, r is the number of correctly predicted words, n is the actual number of words in the test image, w is the number of wrongly predicted words, and N is the number of words in the vocabulary.

4.3.3 Experiment Settings

We compare the results of three different experiments on sub-space techniques for semantic propagation based image auto-annotation, i.e. classic NMF (denoted as CNMF), NMF with sparseness constraints (denoted as NMFsc) and SVD. The results of SVD based approach were taken directly from the work of [2] for comparison purposes. The projected gradients based method developed by [14] was used for the classic NMF. As for NMF with sparseness constraints, the algorithm[1] developed by [9] was adopted.

Additional parameters need to be set in using the sparseness constrained version of NMF, namely the degree of sparseness of W and H. The constraints can be placed on W, or H, or both, depending on the particular problem to be solved. [9] described an example in which a doctor tries to analyze disease patterns. It was assumed that most diseases are rare (hence sparse), and present in a small number of patients. However, each disease can cause many symptoms. Therefore, given a matrix in which each column denotes an individual patient and each row denotes a symptom, it might be better to place sparseness constraints on the "coefficients" (rows in H) but not the "basis vectors" (columns in W). Based on empirical analysis of the Washington images set, we chose to constrain W but not H for two reasons. Firstly, as the number of visual terms was set to 3000 but on average each image generated only several thousand salient regions, it is unlikely that an object or object part from an image contains a variety of different visual terms. In other words, the "basis vectors" in W tend to be sparse. Secondly, many objects/keywords exist in a large number of images in the data-set. For example, 484 of the images contain "tree", and 218 and 199 of them have "building" and "people". These keywords affect a big portion of the data-set. It is more appropriate to unconstrain H. Our experiments also confirmed this hypotheses; the results of experiments using constrained W and unconstrained H were much better than using unconstrained W and constrained H, or when both were constrained. When both W and H are unconstrained, it becomes the classic NMF, the results of which are presented in the following.

4.3.4 Results

We repeated the experiments of CNMF and NMFsc for image auto-annotation 100 times on different training and test sets. For each run, a randomly selected 50:50 mix of images from the Washington data-set were used to build a set of training images and a set of test images. Precision, recall and normalised score (E_{ns}) were calculated at different values of $M(1,2,3)$, which represents the number of top images chosen for propagation. The average results from the 100 runs are used in the following. The number of visual terms was set to 3000 and the term-by-document matrix was not weighted.

The dimensionality of the sub-space generated by NMF, or the value of r in $V \approx W_{n\times r}H_{r\times m}$, is a number pre-defined by users. Theoretically, it should relate to the class number of object or object parts in the data-set. However, at this time, finding the optimal value of r is still a difficult and unsolved problem. In our experiments, we varied its value from 2 to 200 with a fixed step of 2. Besides, for NMFsc, the results were calculated at different sparseness degrees of W, i.e. 0.5, 0.6, 0.7, 0.8 and 0.9.

In order to choose the optimal sub-space dimensionality and degree of sparseness for NMFsc, we use normalised score as a single value indicator. Figure 2 depicts the values of E_{ns} at different settings of dimensionality (r) for different degrees of sparseness of W. For each test image, the closest training image was chosen for propagation, i.e. $M = 1$. The horizontal axis represents the value of r, and the vertical axis represents the value of normalised score E_{ns}. Each degree of sparseness generated one curve in the chart, denoted by different colours. Figure 3 and 4 show the results of using 2 and 3 top images for propagation respectively, i.e. $M = 2, 3$. As can be seen from the figures, E_{ns} achieves the highest when the sparseness is 0.8 (the green curve) and r is around 100. We have also calculated the results for CNMF and depicted this in Figure 5. The best performance is found at $r \approx 40$, as shown in the chart. The above mentioned values of parameters are chosen for comparisons with SVD.

The results in terms of precision, recall and normalised score are summarised in Table 2, along with the results of the methods proposed by [2], namely the vector space and LSI (based on SVD) model. The results of each method are also plotted into a precision-by-recall chart, Figure 6, for a better view of the comparison. As can been seen, the annotation results of NMF with sparseness constraints are better than that of the classic NMF. Besides, NMFsc achieved similar results as SVD when $M = 1$, and slightly better when $M = 2$ and 3. Some samples of annotation results are shown in Figure 7.

5. CONCLUSIONS AND FUTURE WORK

We have investigated the application of NMF, a relatively new matrix factorisation technique, in two different tasks, object class detection and automatic image annotation. In both cases, the quantised salient region based visual term representation of image is used to build the term-by-document matrix. Thanks to the parts-based representation feature of NMF, the basis vectors generated by NMF correspond to object classes that occur frequently in the image set. As a sub-space technique, NMF also outperformed SVD slightly in terms of propagation based image auto-annotation. Therefore, we argue that NMF is not only a potentially effective sub-space technique for information retrieval, but also one that comes with the advantage of parts-based representation of documents.

As we have mentioned, finding the optimal dimensionality of the sub-space in using NMF is still an unsolved problem. Ways to circumvent it are interesting as this problem exists in almost all the sub-space techniques. As for image auto-annotation, the simple semantic propagation based approach was used in this work. We plan to explore more

[1] Code available at: http://www.cs.helsinki.fi/u/phoyer/software.html

advanced approaches that build upon the sub-space generated by NMF, for example, the linear-algebraic technique proposed by [19].

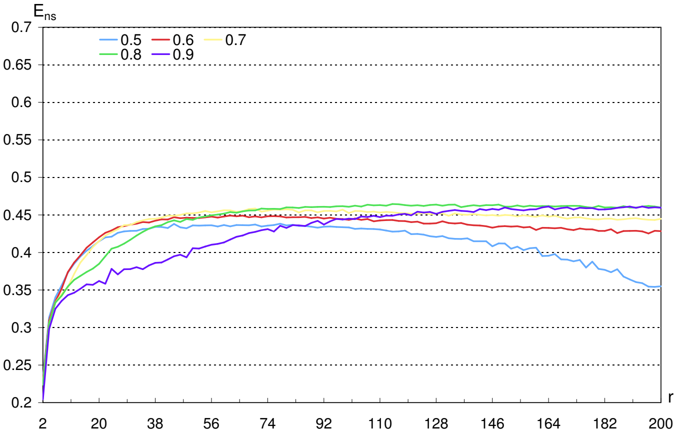


Figure 2: The normalised score (E_{ns}) of applying NMFsc for image auto-annotation. $M = 1$ and the sparseness of W is varied from 0.5 to 0.9.

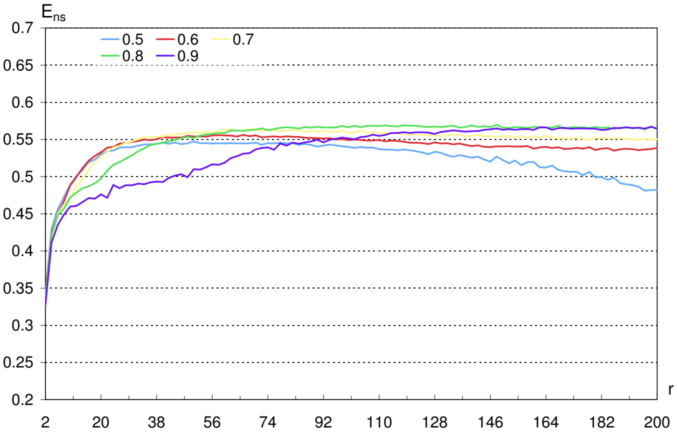


Figure 3: The normalised score (E_{ns}) of applying NMFsc for image auto-annotation. $M = 2$ and the sparseness of W is varied from 0.5 to 0.9.

6. ACKNOWLEDGMENTS

The authors are grateful for the support of EU funding under the Tripod project, IST-FP6-045335, and ImageCLEF, the cross language image retrieval track of the Cross Language Evaluation Forum (CLEF) (first author) and the OpenKnowledge project, IST-FP6-027253 (second author).

7. REFERENCES

[1] Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos, "Automatic image captioning," in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004)*, 2004, pp. 1987–1990.

[2] Jonathon S. Hare and Paul H. Lewis, "Saliency-based models of image content and their application to

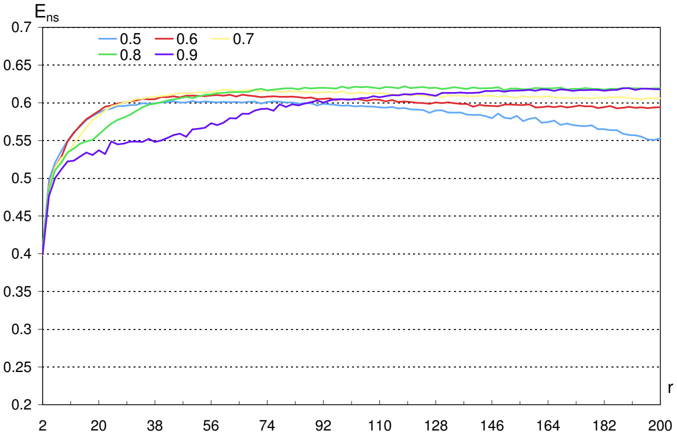


Figure 4: The normalised score (E_{ns}) of applying NMFsc for image auto-annotation.$M = 3$ and the sparseness of W is varied from 0.5 to 0.9.

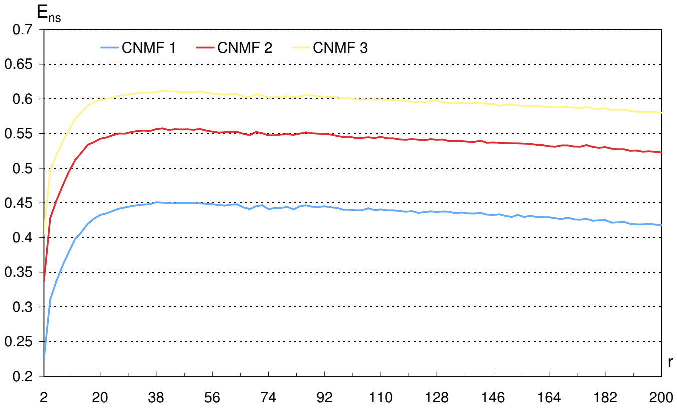


Figure 5: The normalised score (E_{ns}) of applying CNMF for image auto-annotation. The curve "CNMF 1" represents the results when $M = 1$. "CNMF 2" and "CNMF 3" represent the case when $M = 2$ and $M = 3$.

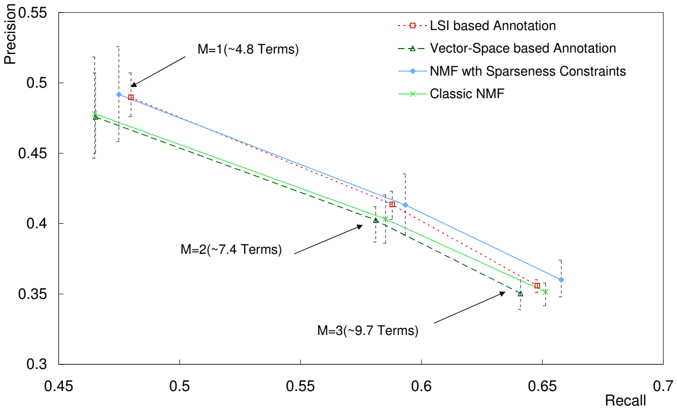


Figure 6: Precision-Recall curves for several different semantic propagation-based image auto-annotation methods. Error bars show range of precision over 100 repeated runs, each of which used a random separation of the Washington set into training and test sets.

Method	Number of Words	Precision	Recall	E_{NS}
Vector-Space	~ 4.8	0.476	0.465	0.450
	~ 7.42	0.402	0.581	0.554
	~ 9.70	0.350	0.641	0.602
LSI(K=40)	~ 4.8	0.490	0.480	0.466
	~ 7.42	0.414	0.588	0.561
	~ 9.70	0.356	0.648	0.609
CNMF	~ 4.8	0.478	0.465	0.450
	~ 7.42	0.403	0.585	0.557
	~ 9.70	0.352	0.651	0.612
NMFsc	~ 4.8	0.492	0.475	0.461
	~ 7.42	0.413	0.593	0.566
	~ 9.70	0.360	0.658	0.619

Table 2: Summary of results of image auto-annotation using several different semantic propagation-based methods.

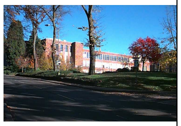
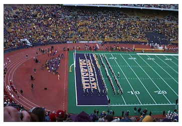
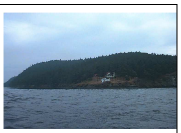
Images Methods			
GT Labels	Clear Sky, Building, Tree, Leafless Tree, Tree Trunk, Grass, Street, Tree	Stadium, Stand, People, Football Field, Band, Post, Track, Banner	Cloudy Sky, Hill, Tree, Building, Water
CNMF	Tree, Grass, Pole, Building, People, Clear Sky, Water, Partially Cloudy Sky, Sidewalk, Elk	Tree, Stadium, Stand, Football Field, People, Band, Post, Track, Banner, Sky, Lake	Tree, Cloudy Sky, Boat, Water, Mountain, Dock, Powerboat, Mast, Greenery
NMFsc	Tree, Grass, Pole, Building, People, Clear Sky, Sidewalk, Leafless Tree	Cloudy Sky, Stadium, Stand, Football Field, People, Band, Post, Track, Banner, Stands	Tree, Cloudy Sky, Building, Water, Dock, Powerboat, Hill, Ferryboat

Figure 7: Some sample results of image auto-annotation using the classic NMF (CNMF) and NMF with sparseness constraints (NMFsc).

auto-annotation by semantic propagation," in *Proceedings of Multimedia and the Semantic Web / European Semantic Web Conference 2005*, 2005.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788, october 1999.

[4] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, "Dimensionality reduction using non-negative matrix factorization for information retrieval," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2001, pp. 960–965.

[5] Wei Xu, Xin Liu, and Yihong Gong, "Document clustering based on non-negative matrix factorization," in *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 2003, pp. 267–273, ACM Press.

[6] David Guillamet, Jordi Vitrià, and Bernt Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, 2003.

[7] Weixiang Liu and Nanning Zheng, "Non-negative matrix factorization based methods for object recognition," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 893–897, 2004.

[8] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[9] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[10] Eric Gaussier and Cyril Goutte, "Relation between plsa and nmf and implications," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 601 – 602.

[11] Chris Ding, Tao Li, and Wei Peng, "Nmf and plsi: equivalence and a hybrid algorithm," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 641 – 642.

[12] Jonathon S. Hare and Paul H. Lewis, "Salient regions for query by image content.," in *CIVR '04: Proceedings of the 6th ACM international conference on Image and video retrieval*, 2004, pp. 317–325.

[13] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] Chih-Jen Lin, "Projected gradient methods for non-negative matrix factorization," Tech. Rep. Information and Support Service ISSTECH-95-013, Department of Computer Science, National Taiwan University, 2005.

[15] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pp. 888–905, 2000.

[16] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2006, pp. 1605–1614.

[17] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391 – 407, 1990.

[18] Kobus Barnard, Pinar Duygulu, Nando de Freitas, David Forsyth, David Blei, and Michael I. Jordan, "Matching words and pictures," *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[19] Jonathon S. Hare, Paul H. Lewis, Peter G. B. Enser, and Christine J. Sandom, "A linear-algebraic technique with an application in semantic image retrieval," in *CIVR '06: Proceedings of the 6th ACM international conference on Image and video retrieval*, 2006, pp. 31–40.