Automatic Feature Detection and Clustering Using Random Indexing

Haïfa Nakouri[1] and Mohamed Limam[1,2]

[1] Institut Supérieur de Gestion, LARODEC Laboratory
University of Tunis, Tunisia
[2] Dhofar University, Oman
nakouri.hayfa@gmail.com, mohamed.limam@isg.rnu.tn

Abstract. Random Indexing is an incremental indexing approach that simultaneously performs an implicit Dimensionality Reduction and discovers higher order relations among features lying in the vector space. The present work explores the possible application of Random Indexing in discovering feature contexts from image data, based on their semantics. We propose an automatic approach of image parsing, feature extraction, indexing and clustering, showing that the Feature Space model based on Random Indexing captures the semantic relation between similar features through a mathematical model. Experiments show that the proposed method achieves good clustering results on the large Corel database of 599 different semantic concepts.

Keywords: Random indexing, Dimensionality reduction, Semantic indexing, Context discovery, Feature clustering.

1 Introduction

Most of the image analysis approaches consider each image as a whole, represented by a D-dimensional vector. However, the user's query is often just one part of the query image (i.e., a region in the image that has an obvious semantic meaning). Therefore, rather than viewing each image as a whole, it is more reasonable to view it as a set of semantic regions of features. In this context, we consider an image feature as a relevant semantic region of an image that can summarize the whole or a part of the context of the image.

In this work, we propose the Feature Space model similarly to the Word Space model [14] that has long been used for semantic indexing of text. The key idea of a Feature Space model is to assign a vector (generally a sparse vector) to each feature in the high dimensional vector space, whose relative directions are assumed to indicate semantic similarities or similar representations of the features. However, high dimensionality of the semantic space of features, sparseness of the data and large sized data sets are the major drawbacks of the Feature Space model. We also use a representation formalism called Random Indexing (RI) where the whole vector in its integrity has a meaning, not any single element of the vector alone. RI is based on Kanerva's work [7] on sparse distributed memory. It was proposed by Karlgren and Sahlgren [14,8] and was originally used

A. Elmoataz et al. (Eds.): ICISP 2014, LNCS 8509, pp. 586–593, 2014.

as a text mining technique. It is a word-occurrence based approach to statistical semantics. RI uses statistical approximations of the full word-occurrences data to achieve Dimensionality Reduction. Besides, it is an incremental vector space model that is computationally less demanding. The RI model reduces dimensionality by, instead of giving each word a whole dimension, it gives them a random vector with less dimensionality than the total number of words in the text. Thus, RI results in a much quicker time and fewer required dimensions.

Therefore, RI was developed to cope with the problem of high dimensionality in the Word Space model and also as an alternative to Latent Semantic Analysis [9]. Many works used RI for text indexing and words' semantic creation [3,6,13,16,17,19]. Wan et al. [18] used RI to identify and capture Web users' behaviour based on their interest-oriented actions. To the best of our knowledge, no Random Indexing approaches have been used to deal with image features in the Feature Space model, especially for similar semantics discovery between features in image data sets. In this paper we aim to show that a Feature Space modelled using RI can be used efficiently to cluster features, which in turn can be used to identify the context/style represented by a feature. In a Feature Space model, the geometric distance between the features is an indicative of their semantic similarity.

2 The Feature Space Model and Random Indexing

In the Feature Space model, the complete features of any image (containing n features) can be represented in a n-dimensional space in which each feature occupies a specific point in the space, and has a vector associated with it defining its meaning. The features are placed on the Feature Space model according to their distributional properties in the image, such that:

1. The features that are used within similar group of features should be placed nearer to each other.
2. The features that lie closer to each other in the Feature Space represent the same context. Meanwhile, the features that lie farther from each other in the Feature Space model are dissimilar in their representation.

In RI, each dimension in the original space is given a randomly generated *index vector*. These *index vectors* are high dimensional, sparse and ternary. Sparsity is controlled via a length that specifies the number of randomly selected non-zero dimensions. RI has several advantages. it can be performed incrementally and on-line as long as data arrives. Any image can be indexed (i.e., encoded as an RI vector) independently from all other images in the data set. This avoids to build and store the entire feature-image matrix. Besides, newly encountered dimensions (features) in the image data set are easily accommodated without having to recalculate the projection of previously encoded documents. On the other hand, the conventional Singular Value Decomposition (SVD), for instance, requires global analysis where the number of images and features are fixed. Complexity of RI is also very satisfying; it is linear in the number of features in an

image and independent of the data set size. The RI algorithm begins by assigning an *index vector*, of dimension d, to each feature in the image data set. These assignments are chosen to be random, sparse and ternary. The ternary criteria for the *index vectors* were basically introduced by Achlioptas [1] as being a suitable alternative for database environments. As for the sparse *index vectors*, its major concern is to reduce computational time and space complexity. Besides, it has been shown that sparse *index vectors* do not affect the quality of results [11]. The *context vectors* are then constructed by iterating through all the extracted features, and for each feature we identify the context that feature appears in. In cases where the feature appears many times in the same context, that feature is given a higher weight because of its frequency.

3 Feature Clustering Using Random Indexing

In practice, to construct a *context vector*, this latter is initially set to zero. Then, each *index vector* corresponding to a feature is added to the *context vector* that feature appears in. When all features have been added, the *context vectors* are normalized to unit length. Figure 1 illustrates the overall procedure of the feature clustering process based on RI. The clustering procedure is based on four steps: data parsing, data preprocessing, modelling the Feature Space using RI and the feature clustering. More details are outlined in this Section.

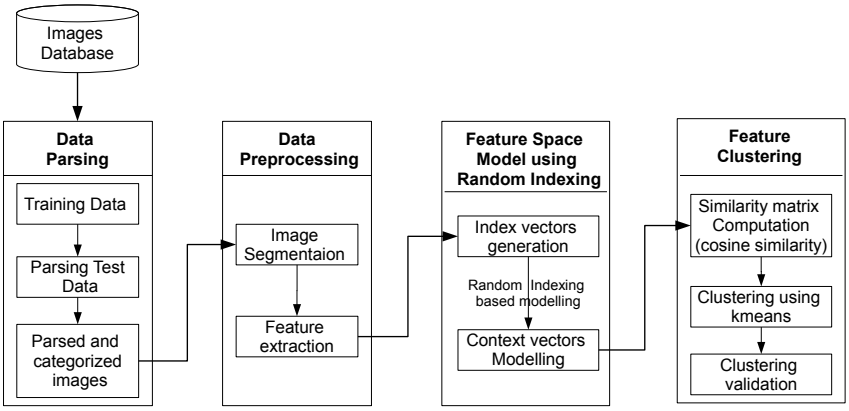

Fig. 1. Feature clustering approach based on Random Indexing

3.1 Image Parsing

The major problem encountered in the context vectors generation is how to detect the relevant index vectors to add to a context vector while parsing the data. To this end, we propose to use an image parsing method before the features extraction and indexing. The parsing phase consists of a training step and a

generalization step. The purpose of this phase is to ensure the complete automation of the process by depicting the semantic category of an image (which is different from the semantic of a single feature) so that we guarantee that the correct index vectors are automatically added to each context vector. For image parsing, we use the method proposed by Schmid [15].

3.2 Data Preprocessing

The preprocessing phase consists in the feature extraction from images. To this end, we first need to perform an image segmentation and then extract the relevant features. In our experiment, we choose to use the conventional Blob-world [2] as an image segmentation method. Figure 2 shows an example of segmented images using the Blob-world method and the extracted features.

Fig. 2. Examples of segmented images

3.3 Feature Space Model Using Random Indexing

Once all relevant features are extracted from images, further analysis should be done to find common contexts between features and create a proper context model for the features clustering. Feature semantics are computed by scanning the features set and keeping a running sum of all *index vectors* for the features that co-occur. As previously noted, a link exists between the occurrence of a feature and its semantics. Finally, the set of generated *context vectors* represent the Feature Space model corresponding to the image data set. Algorithm 1 summarizes the *context vectors* generation procedure.

Algorithm 1. Context vector generation

INPUT: Features $f_i, \quad i = 1, \ldots, n.$
OUTPUT: $n \times d$ context window A.

1. For each feature f_i, obtain a d-dimensional *index vector* $ind_i, i = 1, \ldots, n$ where n is the total number of features.
2. Scanning the feature set, for each feature f_i appearing in the same context than another feature, update its context's vector c_i by adding the feature's corresponding ind_i.
3. Create the feature-to image $(n \times d)$ matrix, also called the context window, where each row is the *context vector* c_i of each single feature.

3.4 Similarity Measure in the Feature Space Model

Basically, *context vectors* give the location of the word in the Word Space model. Similarly, we can assume that *context vectors* give the location of the feature in the Feature Space model. In order to determine how similar the features are in the context, a similarity measure has to be defined. Various schemes e.g., scalar product or vector, Euclidean distance, Minkowski metrics [14], are used to compute similarity between vectors corresponding to the features. However, the cosine similarity [14] might make sense for these data because it would ignore absolute sizes of the measurements, and only consider their relative sizes. Thus, two flowers that were different sizes, but which had similarly shaped petals and sepals, might not be close with respect to squared Euclidean distance, but would be close with respect to cosine distance. We used cosine of the angles between pairs of vectors x and y defined as:

$$sim_{\propto}(x, y) = \frac{xy}{abs(x)abs(y)} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{(\sum_{i=1}^{n} x_i^2)}\sqrt{(\sum_{i=1}^{n} y_i^2)}} \quad (1)$$

3.5 Feature Clustering

The third phase of the clustering process takes in as input the similarity matrix between features. These objects have a cosine similarity between them, which can be converted to a distance measure, and then be used in any distance based classifier, such as nearest neighbor classification. A simple application of the K-means algorithm is performed to cluster the features. K-means [10] is a popular and conventional clustering algorithm that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This partition-based clustering approach has been widely applied for decades and should be suitable for our clustering problem.

4 Experiments

4.1 The Image Dataset

In this work, we used the Corel database [12] to generate the Feature Space model. The Corel image database contains close to $60,000$ general purpose photographs. A large portion of images in the database are scene photographs. The rest includes man-made objects with smooth background, fractals, texture patches, synthetic graphs, drawings, etc. This database was categorized into 599 semantic concepts. Each concept/category/context, containing roughly 100 images, e.g. 'landscape', 'mountain', 'ice', 'lake', 'space', 'planet', 'star'. For clarification, general-purpose photographs refer to pictures taken in daily life in contrast to special domain such as medical or satellite images.

4.2 Clustering Validity Measures

To evaluate the performance of the proposed clustering algorithm, we use the CS index [5] that computes the ratio of *Compactness* and *Separation*. A common measure of *Compactness* is the intra-cluster variance within a cluster, named $Comp = \frac{1}{k}\sum_{i=1}^{k} \parallel \gamma(C_i) \parallel$ where $\gamma(X)$ represents the variance of data set X. *Separation* is computed by the average of distances between the centers of different clusters: $Sep = \frac{1}{k}\sum \parallel z_i - z_j \parallel^2$ where $i = 1, 2, \ldots, k-1$ and $j = i+1, \ldots, k$. It is clear that if the data set contains compact and well separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. Thus, clustering results can be compared by taking the ratio between $Comp$ and Sep: $CS = \frac{Comp}{Sep}$. Based on the definition of CS, we can conclude that a small value of CS indicates compact and well-separated clusters. CS reaches its best score at 0 and worst value at 1. Therefore, the smaller it is the better the clusters are formed. in order to evaluate the effects of varying dimensionality on the performance of RI in our work, we computed the values of CS with d ranging from 100 to 600. The performance measures are reported using average values over 5 different turns. Therefore, we choose $d = 300$ as the dimension of the *index vectors* for RI, which is way less than the original $D = 59900$ (corresponding to total number of images in the data set). As stated in [14], even though the low-dimensional vector space is still relatively large (a few hundred dimensions), it is nonetheless lower than the original space corresponding to the data size (thousands of dimensions). another parameter is crucial for the quality of our indexing is the number of $+1$ and -1 in the *index vectors* ϵ. We use $\epsilon = 10$ as proposed in [4].

4.3 Clustering Results

For the clustering results, the 599 predicted clusters corresponding to the 599 different contexts have been correctly formed and Table 1 shows some of the formed clusters/contexts and their assigned features.

Table 1. Some of the features and their discovered contexts

Description of the feature cluster/Context	
Beach umbrella	Beach
Person lying in the beach	Beach
The Colosseum	Landmarks
London bus	Vehicle

We report the rest of the results using three other validation criteria: precision, recall and the F-measure. These three measures are widely used in pattern recognition and information retrieval. According to our evaluation context, we slightly changed the definitions: *Precision* of a feature is defined as the ratio of the correct clusters to the total number of clusters it is assigned to. The precision (P) of the algorithm is the average precision of all features. *Recall* of a

feature is defined as the ratio of the correct clusters of the feature and the total number of contexts the feature is used in the data set. The recall (R) of the algorithm is the average recall of the features. P and R range between 0 and 1. The F-measure (F) is the combination result of precision and recall and is given by $F = \frac{2RP}{R+P}$. The F-measure reaches its best value at 1 and worst score at 0. As showed in Table 2, the best results of the proposed measures are given for dimension $d = 300$: the smallest Compactness Separation ($CS = 0.32$) and accordingly the largest F-measure ($F = 0.646$). The best formed clusters (e.g. with the least CS index) cause a decrease in precision and hence in F-measure. It can be observed from the results that Random Indexing can improve the quality of features clustering and allows the construction of a high quality Feature Space model. For all context discoveries, a feature is assigned to a cluster if it is closer to this cluster's center. Thus, a feature is assigned to its most similar context.

Table 2. Results of RI-based clustering

Dimension	d=200				d=300				d=400			
Validation Measure	CS	P	R	F	CS	P	R	F	CS	P	R	F
RI-Clustering	0.43	0.462	0.308	0.370	0.32	0.718	0.588	0.646	0.35	0.546	0.434	0.483

The results also support the argument that by constructing the Feature Space model, the outcome space captures the relevant co-occurrence patterns incarnate in the image data set. Each *index vector* of a feature represents a condensed version of all the contexts the feature appears in, and each *context vector* discovers a summary of the significant features corresponding to the context. The collection of vectors all together represents the semantic nature of related features and image contexts.

5 Conclusion

In this paper, we have used a RI based approach, in conjunction with the K-means clustering technique to automatically discover feature semantics from images. Experiments show that the proposed approach works efficiently on the Corel database which support the hypothesis that Feature Space models based on Random Indexing capture the semantic relation between similar image features.

References

1. Achlioptas, D.: Database-friendly Random Projections, pp. 274–281. ACM Press (2003)
2. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(8), 1026–1038 (2002)

3. Giesbrecht, E.: In Search of Semantic Compositionality in Vector Spaces. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS, vol. 5662, pp. 173–184. Springer, Heidelberg (2009)
4. Gorman, J., Curran, J.R.: Random Indexing using Statistical Weight Functions. In: Proceedings of EMNLP, pp. 457–464 (2006)
5. Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality Scheme Assessment in the Clustering Process. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 265–276. Springer, Heidelberg (2000)
6. Hare, M., Jones, M., Thomson, C., Kelly, S., McRae, K.: Activating event knowledge. Cognition Journal 111(2), 151–167 (2009)
7. Kanerva, P.: Sparse Distributed Memory and Related Models. Associative Neural Memories, pp. 50–76. Oxford University Press (1993)
8. Karlgren, J., Sahlgren, M.: From words to understanding. In: Uesaka, Y., Kanerva, P., Asoh, H. (eds.) Foundations of Real-World Intelligence, pp. 294–308 (2001)
9. Landauer, T.K., Foltz, P.W., Laham, D.: An Introduction to Latent Semantic Analysis. In: 45th Annual Computer Personnel Research Conference. ACM (2004)
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
11. Bingham, E., Mannila, H.: Random Projection in Dimensionality Reduction: Applications to Image and Text Data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250 (2001)
12. Müller, H., Marchand-Maillet, S., Pun, T.: The Truth About Corel – Evaluation in Image Retrieval. In: Lew, M., Sebe, N., Eakins, J.P. (eds.) CIVR 2002. LNCS, vol. 2383, pp. 38–49. Springer, Heidelberg (2002)
13. Chatterjee, N., Mohan, S.: Discovering Word Senses from Text Using Random Indexing. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 299–310. Springer, Heidelberg (2008)
14. Sahlgren, M.: An Introduction to Random Indexing. In: Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE (2005)
15. Schmid, C.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178 (2006)
16. Turian, J., Ratinov, L., Bengio, Y.: Word Representations: A Simple and General Method for Semi-supervised Learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
17. Turney, P.D., Pantel, P.: From Frequency to Meaning: Vector Space Models of Semantics. J. Artif. Int. Res. 37(1), 141–188 (2010)
18. Wan, M., Jönsson, A., Wang, C., Li, L., Yang, Y.: Web user clustering and Web prefetching using Random Indexing with weight functions. Knowledge and Information Systems 33(1), 89–115 (2012)
19. Widdows, D., Ferraro, K.: Semantic vectors: a scalable open source package and online technology management application. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), pp. 1183–1190 (2008)