ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

Neural Dimensionality Reduction for Document Processing

Mathieu Delichère [1] & Daniel Memmi [2]

[1] AMOWEBA
1 ave Berthollet, 74000 Annecy (France)
[2] LEIBNIZ-IMAG
46 ave Félix Viallet, 38000 Grenoble (France)

Abstract: Document processing usually gives rise to high-dimension representation vectors which are redundant and costly to process. Reducing dimensionality would be appropriate, but standard factor analysis methods such as PCA cannot deal with vectors of very high dimension. We have used instead an adaptive neural network technique (the Generalized Hebbian Algorithm) to extract the first principal components of a text corpus in order to represent documents economically. The approach is efficient and gives good results in a real Web page clustering application.

1. Introduction

The increasing flow of information, notably through the Internet, has given a renewed importance to information retrieval techniques and textual document processing. Information selection has now become a crucial issue.

To perform the indexing, retrieval and filtering tasks needed for document management, the first operation is to represent texts in a concise operational format. The most common approach is probably the vector-space model (Salton & McGill 1983)(Manning & Schütze 1999). A document is converted to a numerical vector, generally by using lexical features which presumably indicate semantic content.

Some simple pre-processing is required to filter out function words (such as articles, prepositions, particles...) and possibly to reduce morphological variants to a common stem. Only the most significant words (usually nouns and verbs) are retained, and the count of each such term in the document gives a vector component (i.e. a co-ordinate in vector-space). A document thus becomes a vector in the space of possible words or terms, in which the usual vector operations are now applicable, notably vector comparisons for document classification tasks.

Unfortunately, this vector-space representation of texts gives rise to vectors of very high dimension. Vectors of three to five thousand lexical features are common. Such vectors are costly to store and to process. The vectors are also extremely sparse (often containing 90% of null values), and there are obvious correlations between words, making the representation highly redundant.

It is then important to try to reduce the dimensionality of document vectors before further processing. Several methods are available, which can be seen as variants of factor analysis, to find a smaller set of representative dimensions.

The problem is that most common algorithms for factor analysis cannot deal with vectors of such size. We have then used a connectionist technique, the Generalized

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

Hebbian Algorithm, to reduce such large vectors. The adaptive neural algorithm has given us good results for document processing, as will reported here.

2. Principal Component Analysis

Dimensionality reduction methods in general try to find a reduced number of new dimensions to account for the original data (Jolliffe 1986).

Principal Component Analysis (PCA) is the best known of these techniques: the new dimensions, linear combinations of the original features, are given by the eigenvectors (ordered by decreasing eigenvalue) of the covariance matrix of input data. The new features, called principal components, are uncorrelated and of maximum variance so that the new representation is now minimal. Successive components are of decreasing importance, and the first principal components (of higher eigenvalue) usually account for most of the variance in the input data.

Unfortunately, the size of the covariance matrix is very large for high-dimension data vectors, as input vectors of dimension n give rise to a matrix of size $n \times n$. Standard PCA methods cannot then deal with data vectors of more than several hundred features, because space and time costs become prohibitive. This is precisely the case for textual documents, which are often represented with hundreds or thousands of lexical features.

One possible solution would be given by Latent Semantic Indexing (LSI) which computes the singular eigenvectors of the document-term matrix (Deerwester et al. 1990). Because the number of documents is usually smaller, this matrix is much less bulky and it is not necessary to compute a covariance matrix. Final results seem to be semantically acceptable on the whole, but it remains unclear what the new dimensions really mean theoretically.

Another approach is to map high-dimension vectors directly into a low-dimension space using a random matrix, as demonstrated in the WEBSOM system (Kohonen 1998). This can be done very efficiently, but the conservation of semantic information is by no means guaranteed.

We have then elected to use a neural network version of PCA, first proposed by Oja and later developed by Sanger. This technique processes one data vector at a time without ever using the covariance matrix. It is therefore readily applicable to vectors of several thousand features.

3. The Generalized Hebbian Algorithm (GHA)

This is the rather unclear name for an ingenious neural variant of PCA. The original idea was introduced by (Oja 1982) who devised a learning rule enabling a linear neuron ($y = \mathbf{w}.\mathbf{x}$) to extract the first principal component of its input data:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta(t)\,[y(t)\mathbf{x}(t) - y^2(t)\mathbf{w}(t)]$$

where $\mathbf{x}$ is the input vector, $\mathbf{w}$ the weight vector, y the output, and η is the learning rate (typically between 0.1 and 0.01) which may be decreased with time.

The term $y(t)\mathbf{x}(t)$ is clearly Hebbian, maximizing output variance, and $y^2(t)\mathbf{w}(t)$ is a normalizing factor, designed to keep $\|\mathbf{w}\|$ close to 1.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

Learning thus causes the weight vector to represent the first covariance eigenvector, extracting maximum input variance. The output, projection of the input vector on the eigenvector, gives the new value of the input data along the new dimension (i.e. the first principal component).

(Sanger 1989) generalized this learning rule to a one-layer neural network of linear neurons receiving the same input vector, where each neuron extracts one principal component in turn. The learning rule now becomes:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\,[y_i(t)x'_j(t) - y_i^2(t)w_{ij}(t)]$$

with each successive neuron seeing a modified input $x'_j(t)$ computed by subtracting the operation of the preceding neurons:

$$x'_j(t) = x_j(t) - \sum_{k<i} w_{kj}(t)y_k(t)$$

The learning rule for GHA can also be expressed and implemented more concisely with one equation only:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\,[y_i(t)x_j(t) - y_i(t)\sum_{k\leq i} w_{kj}(t)y_k(t)]$$

This generalized rule can be shown to converge toward the first covariance eigenvectors, ordered by decreasing eigenvalue. The algorithm requires only one input vector at a time, and can be implemented locally in true connectionist fashion. But it is simpler and more efficient to train output neurons one after the other rather than in parallel (although Sanger doesn't make this explicit).

Compared to standard PCA, GHA is an indirect method: correlations between features of input vectors are estimated through the accumulation of repeated weight changes during training, without computing the covariance matrix of input data.

Several variants of GHA have later been devised (Diamantaras & Kung 1996)(Fiori 2000) but they do not appear to improve markedly on Sanger's original algorithm, which we have therefore retained here.

4. Document processing

We used a document corpus of about 90 Web pages (corresponding to a page or two of ordinary text) in French, dealing with about 10 different themes. The Web pages were taken from the bookmarks of a human user, and the goal was to perform an unsupervised classification of the set of pages in the context of an interactive browsing system. For more details, see (Delichère 2001).

4.1 Pre-processing

Each document (i.e. Web page) was represented by a lexical vector. Pre-processing followed a common procedure: removal of function words by using a stop-list, 5-letter stemming to reduce to root-forms (also called lemmas or terms), computing the frequency of words by using a variant of the classical TFIDF measure: term frequency in document divided by total frequency in corpus. The stop-list is the only language-dependent item in the procedure and is easy to change.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

We only kept words (or more accurately, terms) above a given frequency threshold and only if they could be found in more than 10 to 20% of documents (to ensure minimum corpus significance, while the TFIDF measure gives more weight to more discriminative terms). But no high-frequency threshold was used.

Pre-processing resulted in a representation space of about 600 dimensions. This is in fact fairly low in such a domain, because our corpus was small; dimension usually increases quickly with corpus size, to stabilize to about 3000 to 5000 lexical features, depending on pre-processing. Yet 600 is already quite a large number for factor analysis techniques.

4.2 Dimension reduction and clustering

Documents vectors were then fed to a GHA network with 20 output neurons for 600 input lines. The output number was chosen by trial and error: higher numbers resulted in very small or inaccurate weight values for subsequent neurons, as computing errors accumulated with successive components. To speed up learning, we used a sequential version of GHA, by training output neurons independently one after the other, rather than in parallel.

We used different learning rates between 0.1 and 0.01 in various experiments, but we kept the value constant during learning. In fact we systematically stopped training after 3 minutes' computation (on a standard workstation) rather than after a fixed number of epochs, because we did not want to keep the end-user waiting.

The GHA algorithm thus reduced 600 dimensions to 20, projecting documents into a new space. We called *concepts* the new dimensions, as they turned out to be quite significant in themselves. Because they represent correlations between words in the documents, the new features reveal semantic themes in the corpus. They are also corpus-dependent and automatically adapted to different corpora.

Here are for example the concepts (new dimensions) computed by the first seven neurons corresponding to the highest eigenvalues. Representing them by the terms contributing most to each concept, one can easily recognize some recent events and the interest centers of the bookmark-owner:

Concept 1: israéliens palestiniens barak sharon ehud
Concept 2: neurone couche entrée poids matrice
Concept 3: kasskooye manager newsletter incubation business
Concept 4: bové josé alegre brésil porto
Concept 5: kasskooye manager incubation business strategy
Concept 6: hoax virus hoaxbuster hybris pétition
Concept 7: hockey tennis sélectionner football
(...)

We now used the reduced data vectors to cluster the documents. To avoid normalizations, we chose Euclidean distance (rather than a dot product) to measure document similarity, and experimented with several versions of the well-known k-means clustering algorithm (Anderbeg 1973). Document clustering might also be done with competitive neural networks (Memmi & Meunier 2000).

We encounter here a typical problem in document processing: how to evaluate the quality of unsupervised clustering. In this case, there are unfortunately no general objective measures (such as recall and precision in supervised classification) and results have to be evaluated by humans in the context of a given task.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

The clusters obtained were very significant indeed, clearly showing the main themes of the corpus. Clusters corresponded closely the initial categories which had motivated the user's bookmark collection. Clustering was also much quicker and easier in the new reduced space than in the original space. In fact, clustering the original vectors would have been impossible within the strict time limits imposed by our interactive task, as clustering was performed in real time at the end-user's request.

Dimensionality reduction thus improved computational efficiency and was semantically relevant at the same time, because the new reduced dimensions represent the most highly discriminating features accounting for the original textual data.

5. Discussion

As far as we know, this is the first application to textual documents of the GHA algorithm, which has usually been applied to image compression and coding. We found that GHA was able to deal with the high-dimension input vectors typical in document processing, whereas images are often represented by much smaller vectors obtained by scanning the image (typically 64 inputs for each sample).

Although the size of our document vectors was still low (600 features) compared to other common document representations (which may number several thousand features), we are confident that GHA can deal with still higher input dimensions: see for example another experiment with 4096 inputs, as reported in (Sanger 1989, 7.3).

To sum up, the Generalized Hebbian Algorithm is able to extract successfully the principal components of high-dimension input vectors, computing only the first components (representing the major part of input variance), a notable saving in time. One may then extract principal components in sequence, stopping when output variance becomes too low.

The adaptive nature of GHA also makes it possible to compute quickly a rough estimate the first principal components (which can be later refined if necessary). This is impossible to do with standard PCA methods, which compute all eigenvectors and perform all the work at once.

GHA is not without drawbacks, however. As with many neural network techniques, the learning rate must be estimated by trial and error, and for precise results, learning may be slow. Moreover, errors accumulate from one neuron to the next and accuracy decreases for subsequent components. The number of useful components should then be estimated, whereas the proportion of variance extracted by each component is not given directly by the algorithm.

The use of GHA is therefore recommended only when one will be satisfied with the first principal components. As they represent most of the input variance, this is generally a reasonable assumption, but one should be aware of the fact.

Lastly, just like PCA, GHA is a linear method, which can only extract linear correlations between the original features. To go beyond this limitation, other methods should be considered, such as SOM nets (Ritter & Kohonen 1989) or Independent Component Analysis (see Hérault & Jutten 1994).

6. Conclusion

To deal with the high-dimension vectors typical of document processing, we have used an adaptive neural technique, the Generalized Hebbian Algorithm, to extract the

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 211-216

first principal components of a corpus of text. This algorithm makes it possible to reduce the dimensionality of very large vectors without computing the covariance matrix of input data. GHA can also extract a limited number of principal components so as to save time. We found the technique to work quite fast (at least for a limited corpus) and to produce good results for document clustering in a real end-user application. We think the approach should be considered as a good candidate to reduce the dimensions of document vectors before further processing.

Acknowledgment

The work described here was done during the development by Amoweba of the Human Links system, a collaborative and distributed search engine designed to mine expert knowledge on the Web. For more information, see the firm's Web site:

http://www.amoweba.com

References

Anderberg M.R. (1973) *Cluster Analysis for Applications*, Academic Press.

Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K. & Hashman R. (1990) Indexing by latent semantic analysis, *Journal of the American Society for Information Science* 41 (6), p. 391-407.

Delichère M. (2001) Etat de l'art et implémentation d'algorithmes de recherche et de classification automatique de documents sur Internet, rapport final EPITA, Paris.

Diamantaras K.I. & Kung S.Y. (1996) *Principal Component Neural Networks: Theory and Applications*, John Wiley & Sons.

Fiori S. (2000) An experimental comparison of three PCA neural networks, *Neural Processing Letters* 11 (3), p. 209-218.

Hérault J. & Jutten C. (1994) *Réseaux Neuronaux et Traitement du Signal*, Hermès.

Jolliffe I.T. (1986) *Principal Component Analysis*, Springer Verlag.

Kohonen T. (1998) Self-organization of very large document collections: state of the art, *Proc. of ICANN'98*, London.

Manning C.D. & Schütze H. (1999) *Foundations of Statistical Natural Language Processing*, MIT Press.

Memmi D. & Meunier J.G. (2000) Using competitive networks for text mining, *Proc. of Neural Computation 2000*, Berlin.

Oja E. (1982) A simplified neuron model as a principal component analyzer, *Journal of Mathematics and Biology* 15, p. 267-273.

Ritter H. & Kohonen T. (1989) Self-organizing semantic maps, *Biological Cybernetics* 61 (4), p. 241-254.

Salton G. & McGill M. (1983) *Introduction to Modern Information Retrieval*, McGraw-Hill.

Sanger T.D. (1989) Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks* 2 (6), p. 459-473.