Matrix Factorisation (Singular Value Decomposition, SVD)

Computerlinguistische Anwendungen SoSe 2020

Cian Higgins 2 August 2020

Singular value decomposition (SVD) is a fundamental concept of machine learning and one of the most widely used unsupervised learning algorithms. SVD is used extensively by many of the world's leading technology companies for recommendation and dimensionality reduction systems. It is also particularly of interest for natural language processing applications. Fundamentally, SVD makes it possible to ignore unimportant semantic dimensions below a particular threshold. This reduces the amount of required data by removing noise but ensures the strong relationships of interest are preserved. In other words, it makes the best possible reconstruction of the matrix with the least possible information.

Matrix Factorisation

At the core of singular value decomposition is the linear algebra technique of matrix factorisation. This is the decomposition of a matrix into a product of matrices. The SVD theorem states that it is always possible to decompose a real matrix C where U, Σ, and V are unique; with U and V as column orthonormal matrices, and Σ a diagonal matrix. In the following formula, the input data matrix C is factorised into a product of three matrices. U is referred to as the matrix of left singular vectors, Σ as the diagonal matrix of singular values, and V the matrix of right singular vectors.

$$C = U\Sigma V^T$$

$$C'_{m\times n} = U_{m\times k}\,\Sigma_{k\times k}\,V^T{}_{k\times n}$$

The reduced SVD matrix C' with size $m \times n$ is represented by the three lower rank matrices: U of size $m \times k$, Σ of size $k \times k$, and V of size $n \times k$ (note that the matrix V is then transposed). k refers to the number of concepts or dimensions which are extracted from the input data. m and n can refer to different things, depending on the implementation. For latent semantic indexing, m would refer to the number of documents and n to the vocabulary. When learning word embeddings, m and n would refer to target and context words. U and V are column orthonormal matrices which means the column vectors of these matrices have unit length, and any two distinct column vectors are orthogonal to one another (see the formulae below). An orthogonal matrix may also be expressed as $M^TM = MM^T = I$, where I is the identity matrix of M.

Euclidean Length of a vector $\vec{d}$:	Two vectors $\vec{c}$ and $\vec{d}$ are orthogonal iff:
$\left\|\vec{d}\right\| = \sqrt{\sum_{i=1}^{n} d_i^2}$	$\sum_{i=1}^{n} c_i \cdot d_i = 0$

Furthermore, the matrix Σ has the singular values of C on its diagonal. A diagonal matrix is one in which all values apart from those on the main diagonal are zero. These singular values are also sorted in a decreasing order. The matrix Σ is used to perform the dimensionality reduction. Dimensions which have low singular values are set to zero, thus eliminating corresponding columns and rows from U and V, respectively. The purpose of this is to remove statistical noise but retain the important semantic dimensions. This means that the reduced SVD matrix C' is an improvement on and has better similarity values than the original input matrix C.

Learning Word Embeddings

The matrix chosen as the input depends on what SVD is being used for. If the aim of the SVD is to learn word embeddings, the input should be a word co-occurrence count matrix. A word embedding is a dense vector which represents semantic and other properties of a given word. Typical values for the number of properties of the embedding lie in the range of fifty to one thousand $(50 \leq k \leq 1000)$. The classical approach for a co-occurrence matrix is a target word (w_t) and context word (w_c) matrix. Entries in the matrix correspond to how many times a given target word occurs with each context word. The definition of a co-occurrence can be tweaked as required for the specific task; for instance, it could be defined as occurring in the same sentence or as inside a window around the target word. These raw co-occurrence counts can then be weighted using positive pointwise mutual information (PPMI). It is calculated for each entry as follows, with an optional offset, k:

$$PPMI(w_t, w_c) = \max\left(0,\ \log\left(\frac{P(w_t, w_c)}{P(w_t)P(w_c)}\right) - k\right)$$

The resulting matrix with each entry as a weighted PPMI calculation of a co-occurrence count can now be used as the input for the singular value decomposition. The word embeddings with improved word-word similarity values are contained within the reduced U matrix, otherwise known as the left singular vectors.

Latent Semantic Indexing

The input could also be a word-document matrix. The resulting reduced SVD of this weighted word-document matrix would yield better query-document similarities. This is useful in the field of information retrieval and is referred to as latent semantic indexing/analysis. LSI takes documents that communicate the same topics i.e. they have semantic similarity, but use different words, making them seem dissimilar in the vector space. The new representations in the reduced SVD vector space have higher similarities for these documents. LSI has proven to be incredibly useful in the area of search engines. For example, a user could type a query and retrieve documents with the same sentiment as the query, but with little or no overlapping word usage.

Optimality

According to the optimality property, a singular value decomposition gives the closest rank k approximation of the input matrix. This is based on the Eckart-Young Theorem. Optimal in this sense means that there exists no other matrix of the same dimensionality that approximates C better. The Frobenius norm of the matrix $C - C'$ is used as the measure of approximation:

$$\|C - C'\|_F = \sqrt{\sum_i \sum_j \left(c_{ij} - {c'}_{ij}\right)^2}$$

There will only ever be one matrix which is the best possible approximation. The matrix used by the SVD is therefore a unique solution.

Implementation using Python and scikit-learn

SVD is relatively easy to implement using Python and the machine learning toolkit scikit-learn. The class that can be used is `sklearn.decomposition.TruncatedSVD`. A fit and transform method can be performed on the input matrix and a matrix with reduced dimensionality and improved similarity values will be returned.

Conclusion

SVD is clearly a powerful unsupervised learning algorithm. The effect of latent semantic indexing has massive advantages in the field of information retrieval and in particular, search engines. Furthermore, it is hard to imagine streaming and video platforms without the recommendation systems which have been made possible with SVD.

The dimensionality reduction of SVD provides a unique and optimal solution which is the best possible reconstruction whilst using the least possible information. This lower dimensional subspace retains the important relationships of the data and can then be used in other machine learning models with improved performance.

Word embeddings learnt using SVD on a PPMI co-occurrence matrix closely approximate results from Word2Vec making it a useful alternative. The dense word embeddings learnt using SVD provide us with semantically meaningful representations of words. Word embeddings are now ubiquitous in NLP applications, e.g. sentiment analysis and parsing.

References

Roth, B., & Schütze, H. (2020). *Embeddings learned by matrix factorization* [Slides from Lecture 04]. CIS, LMU Munich.

Roth B. (2020). *Korpus-basierte semantische Ähnlichkeit*; WordSpace [Slides from Lecture 04]. CIS, LMU Munich.

Baker, K. (2005). *Singular value decomposition tutorial.* pp. 14-23.

Hofmann, T. (1999). *Probabilistic latent semantic indexing*. Paper presented at the 22nd Annual International ACM SIGIR Conference On Research And Development In Information Retrieval (SIGIR '99) pp. 50-57. Berkeley, CA.

Luboobi, P. (2018). *Foundations of machine learning: singular value decomposition (SVD)*. Medium. https://medium.com/the-andela-way/foundations-of-machine-learning-singular-value-decomposition-svd-162ac796c27d

TruncatedSVD (n.d.). Scikit-learn. Retrieved 1 August 2020 from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html#sklearn-decomposition-truncatedsvd

Eckart, C. & Young, G. (1936). The approximation of a matrix by another of lower rank. *Psychometrika*, 1(3), 211-218.