A Review on the Applications of PSVD, Datasets and Matrices

Zheng Wang

April 21, 2014

1 Applications of PSVD

This section introduces some applications that use PSVD to find a low-rank approximation to some specially constructed matrix. The performance of these applications highly relies on the efficiency, accuracy and scalability of the PSVD algorithm.

1.1 Latent Semantic Indexing

LSI [8][2][3] is an information retrieval technique. Unlike traditional techniques that lexically match words in the query with words in each document, LSI searches for a semantic structure in word usage across all documents. However, the semantic structure is obscured by the variability in word choice due to synonymy and polysemy. Therefore the PSVD of a term-document matrix is performed to remove the noise caused by variability in word usage and to keep the important semantic structure. With LSI, one can retrieve information based on the conceptual topics of the documents.

Consider a collection of n documents that includes m terms (or words) in total. LSI starts by writing this collection into an $m \times n$ term-document matrix $\mathbf{A}$, where each column represents a document and each row represents a term. The element a_{ij} in $\mathbf{A}$ is a weight measuring how important term i is to document j. This weight is calculated based on the frequency of term i in document j and the frequency of term i in the whole collection. Since usually only a small portion of the terms occur in one document, $\mathbf{A}$ is sparse. Considering that the number of conceptual topics covered by the documents is much smaller than the number of documents, PSVD is used to reduce the rank of $\mathbf{A}$. Computing the dominant k singular triplets corresponds to extracting the k most important conceptual topics. The dominant singular vectors span an LSI space where the information retrieval is performed.

Real-world applications involving millions of documents is common. For example, the PubMed Abstracts dataset in Table 2.1 contains over 8 million abstracts of life sciences and biomedical articles. Massive datasets like this impose a great challenge on the PSVD calculation.

1.2 PCA for Feature Extraction

Using a good set of features to describe a dataset is key to success in pattern recognition tasks such as classification, clustering and regression. However, real-world observations usually involve a large number of variables that are redundant and noisy. Therefore extracting a small set of representative features to capture the main structure of a dataset prove to be an essential pre-processing step in data analysis.

Feature extraction techniques seek a feature space of reduced dimensionality and map the original high-dimensional data points into it. PCA [6] is a widely used dimensionality reduction techniques for feature extraction.

Consider a dataset of n data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ that is described by m variables. PCA seeks a k-dim ($k < m$) subspace such that the orthogonal projection of $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ onto this subspace maximizes the variance of the projected data. The orthonormal basis of the wanted subspace consists of the k dominant left singular vectors of the centered data matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_n]$, where $\mathbf{a}_j = \mathbf{x}_j - \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$. The j-th dominant left singular vector of $\mathbf{A}$ is called the j-th principal component.

1.3 Eigenfaces for Facial Recognition

Facial recognition [4][10] is the task of identifying a given face in a human face image database. One important technique for facial recognition is computing a set of vectors–the eigenfaces–that best capture the variations among the known face images. These eigenfaces span a low-dimensional "face space" in which recognition is done. Given a new face image, one can project it into the face space and then compare its position in the face space with the positions of the known images from the database [10].

Each digital image with $r \times c$ pixels is treated as an rc-dimensional column vector by concatenating the columns of pixels in the original image. Therefore a dataset of n such images constitutes an $rc \times n$ matrix. Subtracting the mean of all face vectors from each face vector results in a zero-centered matrix $\mathbf{A}$. The left singular vectors of $\mathbf{A}$ are the principal components [6] of the image dataset, and they are called the eigenfaces in the context of facial recognition. Not all the eigenfaces are needed to form a face space for recognition. Since the principal components are directions that capture most variance among the known face images, only the dominant left singular vectors of $\mathbf{A}$ are needed to reconstruct most faces fairly but efficiently.

2 Datasets and Matrices

We tested the PSVD algorithms on a large set of matrices that naturally arise from various real-world applications (section 1). This section briefly describes types and sources of the datasets, as well as how matrices for testing are constructed from these datasets. Basic statistics of the matrices are summarized in Table 2.1.

2.1 Sparse Matrices

The *Enron Emails* [1], *20 Newsgroups* [9], *NYtimes News* [1] and *PubMed Abstracts* [1] datasets are collections of texts in the form of bag-of-words.

- *Enron Emails* contains 39,861 emails with 28,102 unique words.
- *20 Newsgroups* contains 11,269 news articles with 53,975 unique words.
- *NYtimes News* contains 300,000 news articles with 102,660 unique words.
- *PubMed Abstracts* contains 8,200,000 journal abstracts with 141,043 unique words.

Dataset		Matrix			
Name	Type	# rows (m)	# col's (n)	# nonzeros	size (in GB)
20 Newsgroups	text	53,975	11,269	1,467,345	0.022
Enron Emails		28,102	39,861	3,710,420	0.056
NYTimes News		102,660	300,000	69,679,427	1.041
PubMed Abstracts		141,043	8,200,000	483,450,157	7.265
Gisette	handwritten digits	5,000	13,500	67,500,000	0.503
MNIST		784	70,000	54,880,000	0.409
siam-compete2007	text	30,438	21,519	654,995,322	4.88
epsilon	artificial	2,000	400,000	800,000,000	5.961

Table 2.1: Basic statistics of the matrices on which the PSVD solvers are tested.

These datasets record the number of occurrences of each word in each text. As needed in the LSI application (section 1.1), we constructed term-document matrices from the datasets using tf-idf weighting:

$$tf\_idf(i,j) = tf(i,j) \times idf(i). \tag{2.1}$$

In eq(2.1), $tf(i,j)$ is the (raw) term frequency of term i in document j, i.e. the number of occurrence of term i in document j. $idf(i)$ is the inverse document frequency of term i:

$$idf(i) = log\left(\frac{total\ number\ of\ documents}{df(i)}\right), \tag{2.2}$$

where $df(i)$ is the document frequency of term i, i.e. the number of documents containing term i.

2.2 Dense Matrices

- *MNIST* [7] contains $70,000$ handwritten digits (from 0 to 9), each having 28×28 pixels.
- *Gisette* [1] was constructed from a subset of the MNIST dataset for a feature selection challenge. It contains $13,500$ handwritten digits (4 and 9), each having $5,000$ features. $2,500$ of the features were created from the pixels and are useful for disambiguating digit 4 from digit 9. The other $2,500$ are distractor features having no predictive power.
- *siam-compete2007* [5] was constructed from a collection of texts having $21,519$ documents and $30,438$ terms for a competition on document classification. It uses the binary term frequency and normalizes each data point to unit length.
- *epsilon* [5] was constructed from an artificial dataset used in the Pascal Large Scale Learning Challenge on classification. It contains $400,000$ data points each having $2,000$ features.

For each dataset, the average of all data points are subtracted from every data point to form the centered data matrix required by PCA. Therefore these matrices are dense even if the original dataset is sparse.

References

[1] K. Bache and M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[2] M. W. Berry, S. T. Dumais, and G. W. O'Brian. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37:573–595, 1995.

[3] S. T. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Inform. Sci.*, 41:391–407, 1990.

[4] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006.

[5] R-E Fan. Libsvm data: Classification, regression, and multi-label. `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

[6] I. T. Jolliffe. *Principal Component Analysis.* Springer, 2002.

[7] Y. LeCun, C. Cortes, and C. J.C. Burges. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

[8] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[9] J. Rennie. 20 newsgroups. `http://qwone.com/~jason/20Newsgroups/`, 2008.

[10] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.