Randomised algorithms for approximate matrix factorizations

Kaspar Märtens

Abstract

We give an overview of the randomised algorithms for constructing approximate low-rank factorizations of a matrix, and their error bounds. We focus on the approximate SVD and carry out low-rank image reconstruction, comparing different approaches in terms of accuracy and speed.

1 Introduction

A common task in statistical modelling is to find a low-rank approximation of some matrix of interest. For example, often it is of interest to identify a small number of latent variables underlying the data generation process. This rank reduction problem can be specified as follows: given a matrix $A \in \mathbb{R}^{m \times n}$, we aim to find matrices $B \in \mathbb{R}^{m \times k}, C \in \mathbb{R}^{k \times n}$ with rank $k \ll \min\{m, n\}$ such that $A \approx BC$, i.e. the product of low-rank matrices B and C would be a good approximation of A.

This factorization problem arises in various situations, e.g. when performing principal component analysis (PCA), latent semantic analysis, or even image compression. The standard approach is to carry out Singular Value Decomposition (SVD) on the data matrix, and truncate this to obtain the low rank approximation. However, for large data sets this may be computationally infeasible, and thus more efficient randomised algorithms have been suggested for this purpose.

In this report, we give an overview of the randomised algorithms for low-rank matrix approximations, based mainly on the papers by Halko et al. (2011) and Witten and Candès (2013). For demonstrative purposes, we carry out image compression (i.e. rank reduction on an image) with different approaches and evaluate the approximation quality.

2 SVD for low-rank approximation

Any $m \times n$ matrix A (we assume $m \geq n$) can be decomposed as follows $A = U\, S\, V^T$, where $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times n}$ have orthonormal columns, and S is the diagonal matrix with singular values $\sigma_i \geq 0$ in descending order on the diagonal. This is called the SVD (thin representation).

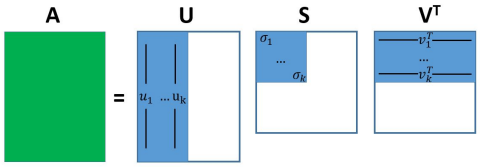

Figure 1: Illustration of SVD ($A = USV^T$) and the truncated SVD. To obtain a rank k approximation of the original matrix A, only the blue shaded regions from U, S, V are used.

When $\text{rank}(A) = k$, then $\sigma_{k+1} = \cdots = \sigma_n = 0$, and $A = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ holds exactly. Otherwise, this expression is used to obtain a rank k approximation of A, and it is called the truncated SVD (Figure 1).

It turns out that this approximation is optimal in a certain sense. When we search for the best approximation of A from the set of all rank k matrixes $\tilde{A}$, we want their difference $A - \tilde{A}$ to be as small as possible. To characterise the size of a matrix (as a linear operator), it is natural to use the norm. For any matrix $B \in \mathbb{R}^{m \times n}$, we define its spectral norm $||B|| := \sup\{||Bx|| : \ ||x|| = 1\}$ which also equals

the largest singular value of B. It turns out that over all matrices $\tilde{A}$ with $\text{rank}(\tilde{A}) \leq k$, the norm of the residual $||A - \tilde{A}||$ is minimised for the truncated SVD, i.e. $\tilde{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, whereas

$$\min\{||A - \tilde{A}|| : \text{ rank}(\tilde{A}) \leq k\} = \sigma_{k+1} .$$

3 Randomised approach for low-rank approximation

As SVD construction is of a computational complexity $O(mn^2 + n^3)$ Golub and Van Loan (1996), this becomes infeasible with large m, n. Thus a faster randomisation based approach has been suggested.

3.1 General framework for low-rank approximation

The procedure for computing a low-rank approximation of some matrix A can be divided into two stages:

1. Construct a low-dimensional subspace which would approximate the range of A. That is, find a matrix $Q \in \mathbb{R}^{n \times k}$ with orthonormal columns ($Q^T Q = I$) such that $A \approx QQ^T A$.
2. Form $C = Q^T A$, and construct SVD of this low rank matrix $C = \tilde{U} S V^T$. As $A \approx QQ^T A$, then we obtain an approximate SVD as follows $A \approx U S V^T$ with $U := Q\tilde{U}$.

The idea is the following: if we manage to carry out step 1 efficiently, then we only need to factorize a small matrix C in step 2. Next, we will describe how random variables can be used in the first step.

3.2 Random variables for approximating the range of a matrix

Viewing matrix A as a linear operator, we define its range to be the set $\{Ax : \ x \in \mathbb{R}^n\}$. We will give some intuition how to approximate the range of matrix with the help of random variables.

Suppose that $\text{rank}(A) = k$ exactly. For a random vector ω, we can think of vector $y := A\omega$ as a random sample from the range of A. Thus, for independent random vectors ω_i, the set $\{y_i = A\omega_i : \ i = 1, ..., k\}$ forms a basis for the range of A, and can be further orthonormalised to obtain Q. However, when A is not exactly of a rank k and contains some additional perturbations, we can use an enriched set $\{y_i = A\omega_i : \ i = 1, ..., k + p\}$ with a small oversampling parameter $p \geq 1$ to have a better chance of spanning the range of A. Halko et al suggest that setting $p = 5$ or $p = 10$ is sufficient for practical purposes.

Denoting G the matrix of stacked column vectors ω_i, i.e. $G := (\omega_1, ..., \omega_{k+p})$, the range of AG approximates the range of A. In principle, various distributions can be used for the random variables, but Witten and Candes show that using i.i.d. Gaussian entries minimises the worst case expected error. As a result, we have the following algorithm for step 1:

- Generate a random $n \times (k + p)$ matrix G with elements i.i.d. from $N(0, 1)$.
- Construct an orthonormal basis $Q \in \mathbb{R}^{n \times (k+p)}$ for the range of matrix $H := AG$.

where Q can be constructed from the QR decomposition of H.

3.3 Bounds for approximation error

When we have fixed k, p and obtained our Q after randomisation step, it is natural to ask how close is $QQ^T A$ to A. By emphasizing the dependence on G, and denoting the residual

$$f(A, G) := A - QQ^T A = (I - QQ^T)A,$$

we are interested in the size of its norm $||f(A, G)||$. Witten and Candes demonstrate that when $p = 0$, the expected error $\mathbb{E}||f(A, G)||$ is not bounded, i.e. it can get arbitrarily large for some choices of A. This motivates $p \geq 1$.

They have derived a sharper upper bound for $\mathbb{E}||f(A,G)||$ than the previous one by Halko et al. By introducing the notion of stochastic ordering, i.e. $X \overset{d}{\leq} Y \iff P(X \leq t) \geq P(Y \leq t)$ for all t, they obtain a stochastic upper bound

$$||f(A,G)|| \overset{d}{\leq} \sigma_{k+1} W \tag{1}$$

where W is the random variable $W := ||f(I_{n-k}, G_2)\,[G_1 \Sigma^{-1}, I_{n-k}]||$, and $G_1 \in \mathbb{R}^{(n-k)\times k}, G_2 \in \mathbb{R}^{(n-k)\times p}, G_3 \in \mathbb{R}^{(k+p)\times k}$ have i.i.d. $N(0,1)$ entries, and $\Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_k)$ contains the singular values of G_3.

As a result, they obtain an upper bound for the worst case expectation of the approximation error

$$\sup_A \mathbb{E}||f(A,G)|| = \sigma_{k+1}\,\mathbb{E}W \leq \sigma_{k+1}\left(1 + (\sqrt{n-k} + \sqrt{k})e\frac{\sqrt{k+p}}{p}\right). \tag{2}$$

3.4 Power trick

When the singular values of A do not decay quickly enough, the performance of the algorithm suffers. The idea of the power trick is based on the fact that if $A = USV^T$, then $(AA^T)^q A = US^{2q+1}V^T$, i.e. the singular values of the matrix $(AA^T)^q A$ decay more rapidly. So the power trick replaces the step $H = AG$ in the algorithm with $H = (AA^T)^q AG$ for some small $q \geq 1$. This introduces $2q$ extra matrix-matrix multiplications, but decreases approximation error.

4 Experimental results

We carried out image compression with randomised SVD approaches in R. I.e. we fixed some target approximation rank k, and evaluated the quality of the result both numerically and visually (Figure 2).

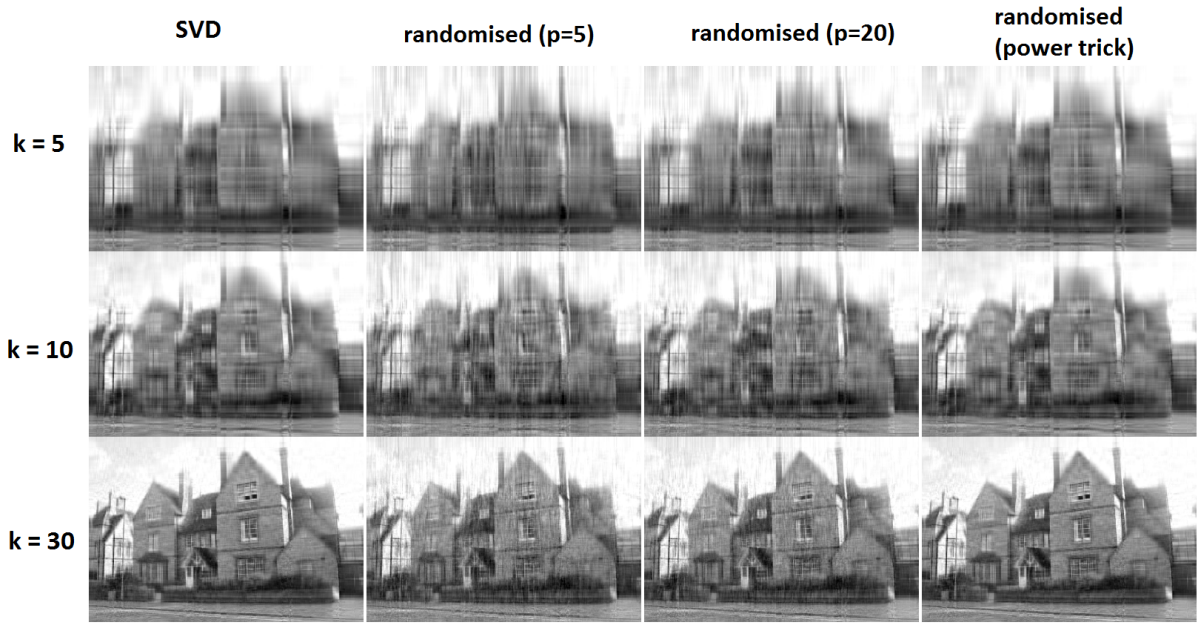

Figure 2: Image reconstruction. A grayscale image (800×539 pixels) was approximated with truncated SVD (first column), and randomised SVDs for p=5 (second column), p=20 (third column), and power trick (q=1, p=5, last column). Results for ranks $k \in \{5, 10, 30\}$ are shown.

Having fixed target rank $k \in \{1, 3, 5, 10, ..., 100\}$, we obtained a reference approximation by truncating the traditional SVD, and compared this with randomised SVD with $k + p$ components. To obtain a fair comparison (i.e. not to exceed rank k), we then truncated our approximate SVD to k first terms.

We experimented with oversampling parameter $p = 5$ and $p = 20$. The latter performs slightly better (distribution of the error in Figure 3), being slightly slower (computation times in Figure 5). The power trick with even $q = 1$ improves approximation accuracy notably (being very close to σ_{k+1}), but it is also slower. We conclude that the randomised rank reduction is substantially faster compared to the traditional one. In case its accuracy remains insufficient, it may be useful to apply the power trick.

In Figure 4, we compare our observed error $||f(A, G)||$ with the upper bounds presented by Witten and Candes, equations (1) and (2). Even though (1) is tight in the sense that for some choices of A the error can get arbitrarily close to the bound, in our scenario the gap between observed errors and the bounds is large (note the y-axis has log-scale). Our experiments with other images showed similar behaviour.

We note that the bounds for the approximation error hold only if we use the whole the rank $k + p$ SVD, and do not truncate this further to the target rank k. Halko et al discuss this issue and provide an upper bound on the increase of approximation error, mentioning that this could be overly pessimistic. Indeed, our observations showed negligible effect of this truncation step (Figure 6).

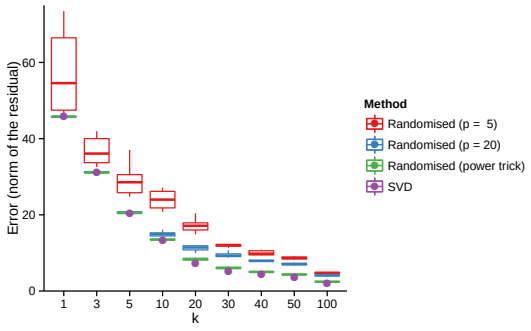

Figure 3: Error (residual norm, y-axis) for image reconstruction from the truncated rank k (x-axis, discrete) SVD representations. The boxplots show error distribution for various approaches (red, blue, green) and comparison to error from SVD (σ_{k+1}, purple dot).

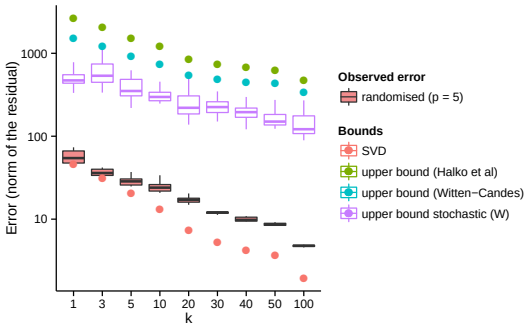

Figure 4: Error (residual norm, y-axis) of the randomised approach (red boxplots) and three upper bounds: distribution of stochastic bound (1) (purple), bound (2) (blue) and the previously known (green). Note the log-scale for y-axis.

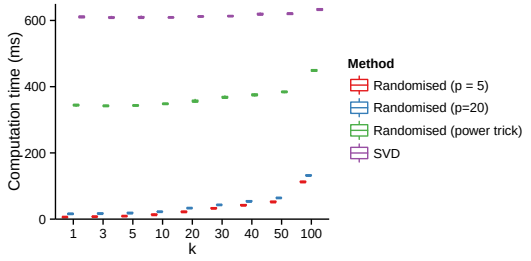

Figure 5: Computation time (y-axis, boxplots) for various approaches.

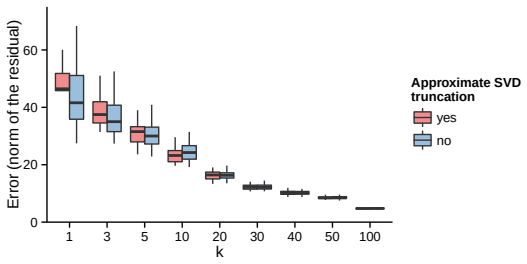

Figure 6: The effect of truncating rank $k + p$ approximate SVD to rank k (blue vs red).

References

Gene H Golub and Charles F Van Loan. Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, 1996.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2), 2011.

Rafi Witten and Emmanuel Candès. Randomized algorithms for low-rank matrix factorizations: sharp performance bounds. *Algorithmica*, 72(1):264–281, 2013.