Incremental singular value decomposition of uncertain data with missing values

Jajimogga Raghavendar
Assoc Professor
NallaMalla Reddy Engineering College
Telangana India

Abstract

We introduce an incremental singular value decomposition (SVD) of incomplete data. The SVD is developed as data arrives, and can handle arbitrary missing/untrusted values, correlated uncertainty across rows or columns of the measurement matrix, and user priors. Since incomplete data does not uniquely specify an SVD, the procedure selects one having minimal rank. For a dense p × q matrix of low rank r, the incremental method has time complexity O(pqr) and space complexity O((p + q)r)—better than highly optimized batch algorithms such as MATLAB's svd(). In cases of missing data, it produces factorings of lower rank and residual than batch SVD algorithms applied to standard missing-data imputations. We show applications in computer vision and audio feature extraction. In computer vision, we use the incremental SVD to develop an efficient and unusually robust subspace-estimating flow-based tracker, and to handle occlusions/missing points in structure-from-motion factorizations

INTRODUCTION

Many natural phenomena can be faithfully modeled with multilinear functions, or closely approximated as such. Examples include the combination of lighting and pose [20] and shape and motion [12,3] in image formation, mixing of sources in acoustic recordings [6], and word associations in collections of documents [1,23]. Multilinearity means that a matrix of such a phenomenon's measured effects can be factored into low-rank matrices of (presumed) causes. The celebrated singular value decomposition (SVD) [8] provides a bilinear factoring of a data matrix M,

$$U_{p\times r}\,\mathrm{diag}(s_{r\times 1})V > r\times q \leftarrow - \mathrm{SVD}_r\, M_{p\times q},\ r \le \min(p,q)$$

where U and V are unitary orthogonal matrices whose columns give a linear basis for M's columns and rows, respectively. For low-rank phenomena, rtrue min(p,q), implying a parsimonious explanation of the data. Since rtrue is often unknown, it is common to wastefully compute a large rapprox rtrueSVD and estimate an appropriate smaller value rempirical from the distribution of singular values in s. All but rempirical of the smallest singular values in s are then zeroed to give a "thin" truncated SVD that closely approximates the data. This forms the basis of a broad range of algorithms for data analysis, dimensionality reduction, compression, noise-suppression, and extrapolation. The SVD is usually computed by a batch O(pq2 + p 2q+q 3) time algorithm [8], meaning that all the data must be processed at once, and SVDs of very large datasets are essentially unfeasible. Lanczos methods yield thin SVDs in O(pqr2) time [8], but rtrue should be known in advance since Lanczos methods are known to be inaccurate for the smaller singular values [1].

A more pressing problem is that the SVD requires complete data, whereas in many

experimental settings some parts of the measurement matrix may be missing, contaminated, or otherwise untrusted. Consequently, a single missing value forces the modeler to discard an entire row or column of the data matrix prior to the SVD. The missing value may be imputed from neighboring values, but such imputations typically mislead the SVD away from the most parsimonious (low-rank) decompositions.

We consider how an SVD may be updated by adding rows and/or columns of data, which may be missing values and/or contaminated with correlated (colored) noise. The size of the data matrix need not be known: The SVD is developed as the data comes in and handles missing values in a manner that minimizes rank. The resulting algorithms have better time and space complexity than full-data batch SVD methods and can produce more informative results (more parsimonious factorings of incomplete data). In the case of dense low-rank matrices, the time complexity is linear in the size and the rank of the data—O(pqr)—while the space complexity is sublinear—O((p+q)r).

Related work

SVD updating has a literature spread over three decades [5,4,1,10,7,23] and is generally based on Lanczos methods, symmetric eigenvalue perturbations, or identities similar to equation 2 below. Zha and Simon [23] use such an identity but their update is approximate and requires a dense SVD. Chandrasekaran et alia [7] begin similarly but their update is limited to single vectors and is vulnerable to loss of orthogonality. Levy and Lindenman [14] exploit the relationship between the QR-decomposition and the SVD to incrementally compute the left singular vectors in O(pqr2) time; if p,q, and r are known in advance and p q r, then the expected complexity falls to O(pqr).

However, this is also vulnerable to loss of orthogonality and results have only been reported for matrices having a few hundred columns. None of this literature contemplates missing or uncertain values, except insofar as they can be treated as zeros (e.g., [1]), which is arguably incorrect. In batch-SVD contexts, missing values are usually handled via subspace imputation, using an expectation-maximization-like procedure: Perform an SVD of all complete columns, regress incomplete columns against the SVD to estimate missing values, then re-factor and re-impute the completed data until a fixpoint is reached (e.g., [21]).

This is extremely slow (quartic time) and only works if very few values are missing. It has the further demerit that the imputation does not minimize effective rank. Other heuristics simply fill missing values with row- or column-means [19]. In the special case where a matrix M is nearly dense, its normalized scatter matrix Σm,n .= hMi,mMi,nii may be fully dense due to fill-in. In that case Σ's eigenvectors are M's right singular vectors [13]. However, this method does not lead to the left singular vectors, and it often doesn't work at all because Σ is frequently incomplete as well, with undefined eigenvectors.

We investigate the use of dimensionality reduction to improve the performance for a new class of data analysis software called "recommender systems". Recommender systems apply knowledge discovery techniques

to the problem of making personalized product recommendations during a live customer interaction. The tremendous growth of customers and products in recent years poses some key challenges for recommender systems. These are:producing high quality recommendations and performing many recommendations per second for millions of customers and products. Singular Value Decomposition(SVD)-based recommendation algorithms can quickly produce high quality recommendations, but has to undergo very expensive matrix factorization steps. In this paper, we propose and experimentally validate a technique that has the potential to incrementally build SVD-based models and promis

In some ways these two challenges are in conflict, since the less time an algorithm spends searching for neighbors, the more scalable it will be, and the worse its quality. For this reason, it is important to treat the two challenges simultaneously so the solutions discovered are both useful and practical. New technologies are needed that can dramatically improve the scalability of recommender systems. Researchers [1, 4, 9, 10] suggest that Singular Value Decomposition (SVD) may be such a technology in some cases. SVD-based approach produced results that were better than a traditional collaborative filtering algorithm most of the time when applied to a Movie data set [9]. However, SVD-based recommender systems suffer one serious limitation that makes them less suitable for large-scale deployment in E-commerce. The matrix factorization step associated with these systems is computationally very expensive and is a major stumbling block towards achieving high scalability

The whole update procedure takes O((p + q)r 2 + pc2) time1 , spent mostly in the subspace rotations of equation 4. To add rows one simply swaps U for V and U 00 for V 00 . In practice, some care must be taken to counter numerical error that may make J and U not quite orthogonal. We found that applying modified Gram-Schmidt orthogonalization to U when the inner product of its first and last columns is more than some small ε away from zero makes the algorithm numerically robust. A much more efficient scheme will be developed below.

Automatic truncation

n SVD of an r × r matrix would ordinarily take O(r 3) time but since Q is a c-bordered diagonal matrix, it can be rotated into bidiagonal form in O(cr2) time [22], and thence diagonalized in an O(r 2) time bidiagonal SVD [9]. If c = 1, the eigenvalues s 02 and eigenvectors U 0 of arrowhead matrix Q>Q can be computed in O(r 2) time [17]; the remaining singular vectors can also be recovered in O(r 2).s contaminated by numerical or measurement noise). In this case the noise should be suppressed by setting K ← 0 prior to the SVD in equation 3. Since the resulting SVD will have r rather than r +1 singular values, equation 4 can be replaced with the truncated forms

U 00 ← UU0 1:r,1:r ; s 00 ← s 0 1:r ; V 00 ← V 0 :,1:r .

This automatically sizes the SVD is to the effective rank of the data matrix. To explicitly suppress measurement noise, one truncates the completed update to suppress singular values below a noise threshold, derived from the

user's knowledge of noise levels in the measurements.

The update procedure enables online SVDs and SVDs of datasets whose size defeats non-incremental SVD algorithms. The update can be used to add individual vectors, batches of vectors, or to merge SVDs from partitions of the data. We will now concentrate on the vector update and leverage it into linear-time and missing-value SVD algorithms.

Fast incremental SVD of low-rank matrices

Because U and V are tall thin matrices, repeatedly rotating their column spaces makes loss of orthogonality through numerical error an issue. Instead of updating large matrices, we may keep U,V,U 0 ,V 0 separate and only update the small matrices U 0 ,V 0 , with U and V growing strictly by appends. In this fastest incarnation of SVD updating, we build an extended SVD,

Missing data

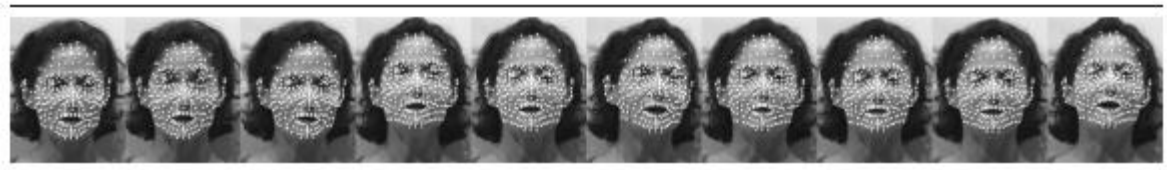
Consider adding a vector c with missing values. In our implementation, these are indicated by setting entries in c to the IEEE754 floating point value NaN (not-a-number). Partition c into c• and c◦, vectors of the known and unknown values in c, respectively, and let U•,U◦ be the corresponding rows of U. Imputation of the missing values via the normal equation

cˆ◦ ← U◦ diag(s)(diag(s)U > • U• diag(s))+(diag(s)U > • c•) = U◦ diag(s)(U• diag(s))+c

The second equality follows from the special sparsity structure of Q. This shows that minimizing k is equivalent to minimizing the log-volume (∑i logσi) of the post-update singular value matrix, which is half the log-volume of the completed data's scatter matrix. Since the amount of total variance in the singular value matrix is lower-bounded by the variance in the known data values, by the log-sum inequality, the only way to minimize the log-volume is to concentrate the variance in a few dominant singular values3 . Consequently equation 11 minimizes growth of the effective rank in the updated SVD. QED. In a related forthcoming paper, we show how these methods can be extended to rapidly factor very large matrices (e.g. 5000×5000) in which more that 95% of the elements are missing. In such cases the minimal rank growth property plays a very important role in guaranteeing a parsimonious model of the data. We show that this translates into considerable improvements over the state-of-the-art in genetic classification and econometric prediction tasks.

Uncertainty, priors, and posteriors

In experimental settings the columns of M are uncertain in the sense that they are samples from a distribution. When the distribution is gaussian and its covariance Σ is known (often the case in vision), the eigenbasis ΩΛΩ> eig ←− Σ enables a directionally weighted least-squares solution for the SVD that maximizes the likelihood p(M|U,S,V) ∝ e −trace(R >Σ −1R) with respect to reconstruction residual R .= Udiag(s)V >−M. Let R 0 .= Λ −1/2Ω>R. Then trace(R 0>R 0) = trace(R >Σ −1R), which is to say that the left-handed certainty warp Λ −1/2Ω> rotates and scales M to make its uncertainty or noise model gaussian i.i.d. Therefore the problem can be solved as



Every 300th frame from a 3000-frame sequence tracked via rank-constrained optical flow with incremental SVD. Dots are superimposed on the video by the tracker.

Subspace optical flow

The use of rank-constraints to regularize rigid-motion optical flow at many points in many frames was first introduced by Irani [11], and the general method has been extended to a variety of projective and motion models. Let P2F×N be the image projections of N points on a 3D surface viewed in F frames, arranged with horizontal and vertical projections on alternating rows. The main insight is that there is an upper rankbound r ≥ rank(P), where r can be determined from inspection of the combined motion/projection model. Algebraically, it follows that dr ≥ rank(P (d)) [11,3], where the vector-transpose operator [](d) partitions a matrix into d-element vertical groups and transposes the groups [15]. This connects to optical flow through the premise that intensity variations through time time are locally linear in surface motion, consequently rank constraints apply directly to measured intensity gradients Y = X d dtP (2) , which should have rank 2r. In this context, it is useful to compute temporal intensity variations Y2F×N and spatial intensity variations X2F×2F in the Kanade-Lucas-Tomasi normal-flow framework, because X may be understood as both the precision matrix of the flow estimate [2] and the covariance matrix of the uncertainty in Y [3]. It follows immediately that the certainty-warp methods in section 6 give the optimal rank-reduction of Y with regard to the information in X

Irani's subspace optical flow algorithm sweeps a W-frame temporal window over an image sequence: In each window of frames, Y,X are measured at estimated correspondences from a reference frame. Y is rank-reduced to rank 2r, then divided by X to estimate the flow, which is in turn rank-reduced to rank r and used to refine the correspondences. This iterates to convergence, and the window advances one frame. Many large SVDs must be computed per frame. We found that most of this computation can be eliminated in favor of incremental SVDs: a rank-2r SVD of gradients Y and a rank-r SVD of correspondences P. In fact, all that is needed are the right subspaces (singular vectors) and singular values of these two SVDs. When new measurements Y are made, they are incorporated into the rank-2r gradient SVD, rank-reduced w.r.t. the updated subspace, divided by X to obtain flow, and cumulatively summed to obtain correspondences P (2) . These are then vector-transposed to P and similarly incorporated into the smaller rank-r SVD and rank-reduced w.r.t. its subspace.5 When the flow has converged within a temporal window, the SVDs are permanently updated with the trusted correspondences, and the window advances

Fig 2:Video frames with profile views synthesized from a structure-from-motion analysis of the 70×80 pixel facial region. The profiles are mirror images except for differences in recovered 3D structure. The profiles on the left have poor structure between the mouth and nose because occlusion artifacts in the tracking were correlated with head nods. The profiles on the right have better shape from the tip of the nose to the top of the mouth because incomplete SVD was used to handle occlusions in the 3D reconstruction

CONCLUSION

We have examined the problem of finding good low-rank subspace models of datasets that may be extremely large, partly or mostly incomplete, contaminated with colored noise, and possibly even nonstationary. By combining an update rule with careful management of numerical noise, rank-minimizing imputations, and uncertainty transforms for MAP inference (with respect to measurement noise and user priors), we developed fast, accurate, and parsimonious online SVD methods, with better

time/space complexity than widely used batch algorithms. This leads to fast online algorithms for vision tasks such as recovery of eigen-spaces, semidense optical flow on nonrigid surfaces with occusions, and automatic handling of occlusions in structurefrom-motion.

References

1. Michael W. Berry. Large scale singular value computations. International Journal of Supercomputer Applications, 6:13–49, 1992.

2. Stan Birchfeld. Derivation of Kanade-Lucas-Tomasi tracking equation. Web-published manuscript at http://robotics.stanford.edu/˜birch/klt/, 1996.

3. Matthew Brand. 3D morphable models from video. In Proc. CVPR01, 2001.

4. J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. Numer. Math., 31:111–129, 1978.

5. P. Businger. Updating a singular value decomposition. BIT, 10:376–385, 1970.

6. Michael A. Casey. MPEG-7 sound-recognition tools. IEEE Transactions on Circuits and Systems for Video Technology, 11(6), June 2001.

7. S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. Graphical models and image processing: GMIP, 59(5):321–332, 1997.

8. Gene Golub and Arthur van Loan. Matrix Computations. Johns Hopkins U. Press, 1996.

9. Benedikt Großer and Bruno Lang. An O(n 2) algorithm for the bidiagonal SVD. Technical Report BUGHW-SC 2000/4, University of Wuppertal Department of Mathematics, 2000.

10. M. Gu and S. C. Eisenstat. A stable and fast algorithm for updating the singular value decomposition. Tech. Report YALEU/DCS/RR-966, Department of Computer Science, Yale University, New Haven, CT, 1993.

11. Michal Irani. Multi-frame optical flow estimation using subspace constraints. In Proc. ICCV, 1999.

12. Michal Irani and P. Anandan. Factorization with uncertainty. In Proc. European Conf. Computer Vision, 2000.

13. J.E. Jackson. A user's guide to principal components. Wiley, 1991.

14. A. Levy and M. Lindenbaum. Sequential karhunenloeve basis extraction and its application to images. Technical Report CIS9809, Technion, 1998.

15. Jan R. Magnus and Heinz Neudecker. Matrix differential calculus with applications in statistics and econometrics. Wiley, 1999.

16. D.D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In Proc. Sixth ICCV, pages 696–702, Bombay, 1998.

17. D.P. O'Leary and G.W. Stewart. Computing the eigenvalues and eigenvectors of arrowhead matrices. Technical Report CS-TR-2203/UMIACS-TR-89-22, University of Maryland Department of Computer Science, February 1989.

18. Conrad Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. Technical Report CMU-CS-93-219, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, December 1993.