A Discrete Optimization Approach for SVD Best Truncation Choice based on ROC Curves

Davide Chicco and Marco Masseroli

***Abstract*— Truncated Singular Value Decomposition (SVD) has always been a key algorithm in modern machine learning. Scientists and researchers use this applied mathematics method in many fields. Despite a long history and prevalence, the issue of how to choose the best truncation level still remains an open challenge. In this paper, we describe a new algorithm, akin a the discrete optimization method, that relies on the Receiver Operating Characteristics (ROC) Areas Under the Curve (AUCs) computation. We explore a concrete application of the algorithm to a bioinformatics problem, i.e. the prediction of biomolecular annotations. We applied the algorithm to nine different datasets and the obtained results demostrate the effectiveness of our technique.**

I. Introduction

Singular Value Decomposion (SVD) is one of the most famous and widely used machine learning algorithms. Since its first appearance in 1873 and its final common definition [1], SVD has shown to be a very powerful applied mathematics method, and used in many fields, such as signal processing, natural language processing, robotics, bioinformatics, recommender systems, and many more.

In SVD, an input matrix $\mathbf{A}$ is decomposed into the $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V^T}$matrices. It is then possible to select a truncation level $k \in \mathbb{N}$ and use the truncated matrices ($\mathbf{U_k}$, $\mathbf{\Sigma_k}$, $\mathbf{V_k^T}$) to reconstruct the input matrix $\mathbf{A}$. Since the resulting reconstruction matrix is slightly different from the input one and shows some semantic relationships between elements, it can be used to infer new statistical knowledge about input matrix elements. But how to best choose the SVD truncation k level remains an open challenge, because this choice can strongly influence the resulting matrix $\widetilde{\mathbf{A}}$, and greatly affects the quality of the obtained solution.

In the past, other scientists have dealt with this issue in various ways. Frank et al. [2] used the Approximation Set Coding (ASC) technique to compute the capacity of a hypothetical channel for a noisy dataset. By using the matrix A as an input noisy dataset, they took advantage of the resulting maximum approximation set coding model that has the highest capacity to choose the best SVD truncation.

Earlier, Vogel [3] chose the best truncation as the one that leads to the best approximate solution of a Fredhom first kind integral equation.

In the electronical engineering field, Isam et al. [4] used the Truncated SVD to overcome the ill conditioning of a Spectrally Efficient Frequency Division Multiplexing (SEFDM) system. They chose the best SVD truncation as the one that minimizes the competitive Bit Error Rate (BER) in system performances. More recently, in the numerical analysis field, Jbilou et al. [5] decided the best SVD truncation through vector extrapolation methods while Hansen [6] selected the optimal truncation index that satisfies the Picard condition. Lastly, in the bioinformatics field, Khatri and colleagues [7] used a heuristically fixed value for the SVD truncation, and so we did in a recent paper [8].

Although these methods are all very good and clever, none of them provides a general solution to the SVD best truncation choice problem that is suitable for all the domains. In this paper, we present a new algorithm based on a discrete optimization method of the Receiver Operating Characteristic (ROC) Areas Under the Curves (AUCs). We applied the algorithm to nine different annotation datasets, and we finally compared the results produced by our algorithm truncation with the results obtained by choosing a fixed truncation value as in [7].

II. Prediction of biomolecular annotations

An *annotation* is the association of a gene (or gene product) with a feature, that describes it. Despite their biological significance, available gene annotations present some issues. As we amply described in [8], they are incomplete by definition, and the annotation profiles of individual genes (or gene products) may be lacking and/or erroneous. In this scenario, computational algorithms able to analyze available annotation data in order to predict new annotations and to appraise the relevance of inferred annotations, or generate a ranked list of new predicted annotations (e.g. to quicken the curation process) are an excellent contribution to the field [10]. One of these algorithm is the Truncated SVD [1].

III. Singular Value Decomposition

Let $\mathbf{A} \in \{0,1\}^{m \times n}$ define the matrix representing all direct annotations of a specific GO sub-ontology (i.e. CC, MF, BP) [11] for a given organism. The m rows of A correspond to genes, while the n columns correspond to GO feature terms (directly or indirectly) annotated to those genes. Let A denote a gene-to-feature matrix, where the assignment of its entries is given by:

$$A(i,j) = \begin{cases} 1, & \text{If gene } i \text{ is annotated to feature } j \\ & \text{or to any descendant of } j. \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

Davide Chicco and Marco Masseroli are with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy davide.chicco@elet.polimi.it masseroli@elet.polimi.it

978-1-4799-3163-7/13/$31.00 ©2013 IEEE

The annotation prediction can be performed by computing the SVD of the matrix $\mathbf{A}$, which is given by $\mathbf{A} = \mathbf{U}\ \mathbf{\Sigma}\ \mathbf{V^T}$ By considering only the most significant rows of $\mathbf{U}$ and columns of $\mathbf{V^T}$, the Truncated SVD method is able to produce a good reduced rank approximation matrix of the input matrix $\mathbf{A}$. This output matrix $\widetilde{\mathbf{A}}$ contains meaningful information about the semantic relationships between the annotations [7]. Therefore, we then use this output matrix for a comparison to the input matrix. The number $r < p$ of $\mathbf{\Sigma}$ non-zero singular values is equal to the rank of the matrix $\mathbf{A}$, where $p = min(m; n)$. For any positive integer $k < r$, it is possible to create a matrix $\widetilde{\mathbf{A}}$, with:

$$\widetilde{\mathbf{A}} = \mathbf{U_k}\ \mathbf{\Sigma_k}\ \mathbf{V_k^T} \tag{2}$$

where $\mathbf{U_k}$ ($\mathbf{V_k^T}$) is a $m \times k$ ($n \times k$) matrix achieved by retaining the first k columns of $\mathbf{U}$ ($\mathbf{V^T}$) and $\mathbf{\Sigma}$ is a $k \times k$ diagonal matrix with the k largest singular values along the diagonal. The matrix $\widetilde{\mathbf{A}}$ is a rank-k approximation of $\mathbf{A}$.
As we extensively explained in [8], this prediction output matrix $\widetilde{\mathbf{A}}$ can be used for a comparison with the $\mathbf{A}$ input matrix.

A. Input matrix and output matrix comparison

The entries of the matrix $\widetilde{\mathbf{A}}$ are real valued. We introduce a threshold τ such that, if $\widetilde{A}(i,j) > \tau$, then gene i is predicted to be annotated to term j. Subject to the original values assumed by the matrix A, the following cases may arise:

- $A(i,j) = 0$ & $\widetilde{A}(i,j) > \tau$: *annotation predicted* (AP), similar to a False Positive (FP)
- $A(i,j) = 1$ & $\widetilde{A}(i,j) > \tau$: *annotation confirmed* (AC), similar to a True Positive (TP).
- $A(i,j) = 0$ & $\widetilde{A}(i,j) \leq \tau$ *non existing annotation confirmed* (NAC), similar to a True Negative (TN).
- $A(i,j) = 1$ & $\widetilde{A}(i,j) \leq \tau$ *annotation to be reviewed* (AR), similar to a False Negative (FN).

As made in [7], we chose the rate of τ as the value that minimizes the number of presumed errors (APs + ARs). Since scientists and biologists discover new biomolecolar annotations every day, and also review and correct the old ones, the annotations in (current or future) databanks cannot be used as the true *gold standard* to appraise the correctness of our predictions. The only item of information that we can count on is that newer annotation datasets base better likelihood of correctness.

IV. Receiver Operating Characteristics (ROC)

A *Receiver Operating Characteristic (ROC)* curve is a graphical plot which depicts the performance of a binary classifier system while its discrimination threshold is varied [12]. Our ROC curves depicts the trade-off between the *ACrate* and the *APrate*, where:

$$ACrate = \frac{AC}{AC + AR} \qquad APrate = \frac{AP}{AP + NAC} \tag{3}$$

for all the possible values of τ. Notice that, in statistical terms, $ACrate = Sensitivity$ and $APrate = 1 - Specificity$. Our ROC curves are built with the $ACrate$ on the y axis and with the $APrate$ on the x axis. In our tests, we considered only the $APrate$ in the normalized interval $[0, 1]\%$, in order to evaluate the best predicted annotations (APs) having the highest likelihood score, because the more NACs we have, the closer the APrate is to zero.

V. Minimizing the truncation level

For the reconstruction of the matrix $\mathbf{A}$, we want to avoid using the $\mathbf{\Sigma}$ matrix singular values that augment the time costs but not the quality of the $\widetilde{\mathbf{A}}$ matrix. For example, in Fig. 1 one may notice that the upper thick red AUC function is asymptotic after the truncation level 80. This means that all the SVD reconstructions using a truncation level larger than 80 would have similar $\widetilde{\mathbf{A}}$ matrix reconstruction and results. On the other hand, the most important $\mathbf{\Sigma}$ matrix singular values are those present in the initial positions. For example, in Fig. 1, where the cyan dotted line represents the singular values ordered by position from the largest to the smallest, the highest singular values are those in the first ten positions. This means that, after these first ten values, the further we go on the x axis, the less important the singular values are with respect to our matrix reconstruction.
For these reasons, minimizing the truncation, while optimizing the quality of the predictions, remains an important goal.

VI. Maximizing the Area Under the Curve (AUC)

Since an AC is an annotation present in input and predicted confirmed in output, and a NAC is an annotation absent in input and confirmed absent in output, having a high number of them means having many confirmations. With many ACs and NACs, the $ARrate$ (in equations (3)) tends to zero and the $ACrate$ tends to one. This corresponds to having a large AUC. For these reasons, the best prediction performances corresponds to larger AUCs, which can be found through a classical optimization. A computationally inefficient way to

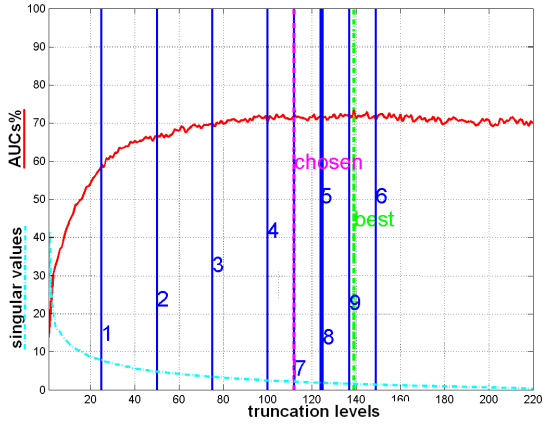

Fig. 1. Illustration of the algorithm behavior for the dataset Gallus gallus BP. The upper continuous red line represents every AUC value (in percentage, on the y axis) for any truncation (on the x axis). The vertical lines represent all the truncations chosen by the algorithm for this dataset; the numbers indicate the order in which they were computed. The truncation chosen by the algorithm as best is the dotted line and labeled *"chosen"*, while the maximum AUC is the dotted green line and labeledl *"best"*. The lower dotted cyan plot represents the $\mathbf{A}$ matrix singular values.

choose the best truncation for the SVD would be to perform the SVD with all possible truncations, and then choose the one that provides the largest AUC. The possible truncation levels range from 1 to $p = min(m, n)$.
For example, if we considered the datasets shown in the #g (number of genes) and #f (number of features) columns of Table I, the number of total SVD's to execute would range from $p = 131$ (Danio rerio CC) to $p = 1,176$ (Danio rerio BP). Executed on a personal computer with Intel Xeon E5320 1.86Ghz two-processors, 32GB RAM, a Dell PERC 5/i SCSI disk with 256Mb cache, and a Microsoft Windows Server 2008 R2 64-bit operating system, tests have shown that a single SVD execution for the most computationally expensive case of Table I (Danio rerio BP) would need about 23 minutes and 4 seconds. We report time performances for the analyzed datasets in the Time column of Table I.
Therefore, we developed a novel algorithm to automatically select the best SVD truncation based on the optimization of the ACrate vs. APrate ROC curve.

VII. The algorithm

The SVD best truncation selection algorithm we developed works as follows. Initially, we make a sampling of all the N non-null singular values, with constant sample intervals of size *step*. The singular values are ordered from the largest to the smallest. We heuristically chose *step* as the 10% of all the non-null singular values of the $\mathbf{A}$ matrix ($step = 10\% \cdot N$). For example, in Fig. 1, the $step = 25$.
For every sampled singular value, we compute the SVD and its corresponding ROC AUC for ACrate in [0,100]% and APrate in [0,1]%. If the AUCs of all the three subsequent samples decrease, or the AUC differences of the last three singular values are lower than $\gamma = 10\%$, we stop the sampling. Otherwise, we keep computing the AUCs for all the subsequent sampled singular values. For example, in Fig. 1, we compute the AUCs for truncations from the first (the line with label 1) to the sixth (label 6) sample. Since the difference between AUC_6 and AUC_5 is less than 10%, and the difference between AUC_5 and AUC_4 is less than 10%, then we stop. After that, we evaluate which is the singular value that corresponds to the largest AUC, and denote it as *lbi*. We then consider a new sub-interval where we refine the search of the maximum AUC; this is the range between the sampled singular value prior to *lbi* (that is $lbi - 1$) and the sampled singular value following *lbi* (that is $lbi + 1$). In the Fig. 1 example, the local maximum corresponds to the truncation at the fifth (label 5) singular value. Therefore, the new sub-interval spans from the fourth (label 4) to the sixth (label 6) truncation computed.
We iterate the search in the new sub-interval, by using a new sampling interval equal to half of the previous ($step = 13$ in Fig. 1). We continue this iteration until in the considered sub-interval there are no more singular values to compute a new AUC, or until the maximum number of sub-interval computation ($numZoom = 4$, heuristically set) is reached, or until the two new truncations computed generate AUCs smaller than the maximum AUC previously computed. This is the case of the Fig. 1 example, where the AUC_8 and the AUC_9 are smaller than previously computed AUC_7.
Once the search is finished, we take as optimal truncation the one that corresponds to the maximum AUC among those computed during the search. In the Fig. 1 example, the seventh (label 7) truncation computed is the optimal.

VIII. Experiments and results

We tested the effectiveness and efficiency of our algorithms on nine datasets, dowloaded from the GPDW data warehouse [13]. The nine different annotation datasets are made by genes of three different organisms (Bos taurus (Bt); Danio rerio (Dr); Gallus gallus (Gg)) and their Gene Ontolgy (GO) annotated terms regarding the biological function features (Cellular Component: CC; Molecular Function: MF; Biological Process: BP). In Table I, we report the results obtained in applying the algorithm to our nine considered datasets. In the Area Diff% column, we show the percentage difference between the AUC computed with the truncation chosen by the algorithm, and the maximum AUC among all the possible truncation. In the #AUCs column, we report the number of AUCs computed before stopping the algorithm.
As a paradigmatic example, we present the results for an indicative case, for the Gallus gallus BP dataset in Fig. 1. One may notice how the best truncation *chosen* by the algorithm appears near to the global *best* truncation, that corresponds to the global maximum AUC.
We reported the dimensions of the nine datasets and the results of our algorithm performance in Table I. One can notice that in two cases (Danio rerio BP and Gallus gallus MF) the algorithm selects the truncation that corresponds to the overall maximum AUC as the global *best*. In many other cases, the difference between the *chosen* AUC and the *best* one is very small: less than 1% (Bos taurus MF, Bos taurus BP, Danio rerio MF, Gallus gallus CC). In two cases where the *chosen* area is not very close from the *best* (Bos taurus CC and Gallus gallus BP), this is balanced by a smaller

TABLE I

Number of genes *(#gs)*, features *(#fs)* and annotations *(#as)* in the July 2009 GPDW version. Percentage difference between the best area chosen by the algorithm and the maximum area of the dataset (*Area Diff%*), number of SVDs and AUCs computed to select the global best area (*#AUCs*), average time for an SVD and AUC computation (*Time*).

dataset	#gs	#fs	#as	Area Diff%	#AUCs	Time (ms)
Bt CC	497	234	7,658	2.59	6	71,445
Bt MF	234	422	3,574	0.59	10	18,232
Bt BP	512	1,023	18,167	0.93	11	57,416
Dr CC	430	131	4,844	1.18	11	16,653
Dr MF	699	261	4,861	0.24	10	23,693
Dr BP	1,528	1,176	38,624	0.00	10	1,380,400
Gg CC	260	309	3,450	0.33	12	18,531
Gg MF	148	225	1,944	0.00	12	86,679
Gg BP	478	509	8,731	1.06	9	162,730

number of iterations (#AUCs column in Table I). Except for one dataset (Danio rerio CC), where we have many iterations and the penultimate largest area, all the dataset tested show that our algorithm hit the target of choosing an AUC near to the maximum one, and maximized the truncation value.

Once we computed the best truncations with our algorithm, we ran our prediction software and tested its performance, in terms of correctness of the annotations predicted. We also performed the prediction by using the fixed SVD truncation used by Khatri et al. in [7], and compared these results. Additionally, we ran our analysis prediction software on the GPDW November 2009 datasets, and then we compared our predictions with the March 2013 version of the same datasets. We show the results of this comparison in Table II.

Comparing the results of the prediction made by using our best truncation in the uppert part of Table II with the results come from the use of a fixed truncation equal to $10\% \cdot min(\#genes, \#features)$ in the lower part of Table II, one can notice that our method outperforms the other one. Although the prediction with the fixed truncation produces more annotations, the percentages of the annotations predicted and found confirmed in the new version of the database of the upper part of Table II (upDb% column) are higher than the corresponding percentages the table lower part, in five cases on nine, while in the other cases they are the same (no annotation found). Anyway, since the databanks are always incomplete, many annotations predicted by our program might be correct but not yet added to the available databases.

TABLE II

COMPARISON BETWEEN THE RESULTS OF THE SVD PREDICTION BY USING OUR ALGORITHM FOR BEST TRUNCATION SELECTION (UPPER TABLE), AND THE RESULTS BY USING A FIXED TRUNCATION USED IN THE [7] PAPER (LOWER TABLE). THE τ THRESHOLD MINIMIZES THE SUM $APs + ARs$. *APs*: THE NUMBER OF ANNOTATIONS PREDICTED; *anDb*: THE NUMBER AND PERCENTAGE OF PREDICTED ANNOTATIONS FOUND IN THE NOVEMBER 2009 GPDW VERSION; *IEA anDb%*: PERCENTAGE OF IEAS AMONG THESE ANNOTATIONS; *upDb* (*upDb%*): NUMBER OF PREDICTED ANNOTATIONS FOUND IN THE MARCH 2013 UPDATED DB (PERCENTAGE OVER THE PREDICTED ONES).

dataset	k	τ	APs	anDb	IEA anDb%	upDb	upDb%
k chosen by our algorithm							
Bt CC	25	0.45	57	2	100	4	7.02
Bt MF	74	0.46	12	1	100	1	8.33
Bt BP	126	0.44	27	1	100	2	7.41
Dr CC	55	0.48	3	0		0	
Dr MF	88	0.50	21	1	100	3	14.29
Dr BP	189	0.44	143	0		1	0.70
Gg CC	47	0.49	0	0		0	
Gg MF	25	0.49	11	1	100	0	
Gg BP	112	0.44	7	0		0	
$k = 10\% \cdot min(\#gene, \#features)$							
Bt CC	23	0.42	84	2	100	4	4.76
Bt MF	23	0.37	161	6	100	4	2.48
Bt BP	51	0.40	230	1	100	4	1.74
Dr CC	13	0.52	21	0		0	
Dr MF	26	0.50	40	1	100	2	5.00
Dr BP	117	0.45	82	0		0	
Gg CC	26	0.45	30	1	100	0	
Gg MF	15	0.43	33	1	100	0	
Gg BP	48	0.43	50	0		0	

IX. CONCLUSIONS

In this paper we presented a general discrete optimization algorithm for the choice of the best truncation for the Truncated SVD method, and applied it to the prediction of biomolecular annotations. Our optimization algorithm reaches the goal to find the truncation that optimizes the AUC of the ROC curve. Our tests shown that our algorithm chooses truncation levels that, besides being very near to the general maximum AUC and with few iterations, outperforms the predictions made with the heuristically fixed level chosen in [7]. Future works will address benefits and disadvantages related to the annotation prediction by considering all the GO ontologies collectively instead of independently, in order to exploit the latent correlations existing between them, and additional work will also be related to a new validation procedure based on the check of the presence the predicted annotations in the literature.

REFERENCES

[1] G. H. Golub, "Singular value decomposition and least squares solutions", *Numerische Mathematik*, vol. 14, 1970, pp. 403–420.

[2] M. Frank and J. M. Buhmann, Selecting the rank of truncated SVD by Maximum Approximation Capacity. *ArXiv.org*, vol. 3176, 2011, pp. 1–8.

[3] C. R. Vogel, Optimal Choice of a Truncation Level for the Truncated SVD Solution of Linear First Kind Integral Equations When Data are Noisy, *SIAM Journal on Numerical Analysis*, vol. 23, 1986, pp. 109–117.

[4] S. Isam, I. Kanaras and I. Darwazeh, A Truncated SVD Approach for Fixed Complexity Spectrally Efficient FDM Receivers. *Proceedings of IEEE WCNC 2011*, 2011, pp. 1584–1589. IEEE Press, New York, NY, USA.

[5] K. Jbilou, L. Reichel and H. Sadok, Vector extrapolation enhanced TSVD for linear discrete ill-posed problems. *Numerical Algorithms*, vol. 51, 2009, pp. 195 – 208.

[6] P. C. Hansen, The discrete picard condition for discrete ill-posed problems. *Bit Numerical Mathematics*, vol. 30, 1990, pp. 658–672.

[7] P. Khatri, B. Done, A. Rao, A. Done and S. Draghici, A semantic analysis of the annotations of the human genome. *Bioinformatics*, vol. 21, 2005, pp. 3416–3421.

[8] D. Chicco, M. Masseroli and M. Tagliasacchi, Genomic Annotation Prediction Based on Integrated Information. *Proceedings of CIBB 2011 - LNCS*, vol. 7548, 2012, pp. 238–252, Springer, Heidelberg, Germany.

[9] M. Masseroli, D. Chicco and P. Pinoli. "Probabilistic Latent Semantic Analysis for prediction of Gene Ontology annotations", in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2012.

[10] G. Pandey, V. Kumar and M. Steinbach, Computational Approaches for Protein Function Prediction: A Survey. Technical report, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA; 2006.

[11] The Gene Ontology Consortium, Creating the Gene Ontology Resource: Design and Implementation, *Genome Res.*, vol. 11, 2001, pp. 1425–1433.

[12] T. Fawcett, ROC Graphs: Notes and Practical Considerations for Researchers. Technical report, HP Laboratories, 2004.

[13] A. Canakoglu, G. Ghisalberti and M. Masseroli, Integration of Biomolecular Interaction Data in a Genomic and Proteomic Data Warehouse to Support Biomedical Knowledge Discovery, *Proceedings of CIBB 2011 - LNCS*, vol. 7548, 2012, pp. 112–126, Springer, Heidelberg, Germany.