Ordozgoiti, Bruno; Pai, Sachith; Kolczynska, Marta

Insightful dimensionality reduction with very low rank variable subsets

Published in:
Proceedings of the Web Conference, WWW 2021

DOI:
10.1145/3442381.3450067

Published: 03/06/2021

Document Version
Publisher's PDF, also known as Version of record

Insightful Dimensionality Reduction with Very Low Rank Variable Subsets

Bruno Ordozgoiti
Dept. of Computer Science
Aalto University
Finland
bruno.ordozgoiti[at]aalto.fi

Sachith Pai
Dept. of Computer Science
University of Helsinki
Finland
sachith.pai[at]helsinki.fi

Marta Kołczyńska
Institute of Political Studies of the
Polish Academy of Sciences
Poland
mkolczynska[at]isppan.waw.pl

ABSTRACT

Dimensionality reduction techniques can be employed to produce robust, cost-effective predictive models, and to enhance interpretability in exploratory data analysis. However, the models produced by many of these methods are formulated in terms of abstract factors or are too high-dimensional to facilitate insight and fit within low computational budgets.

In this paper we explore an alternative approach to interpretable dimensionality reduction. Given a data matrix, we study the following question: are there subsets of variables that can be primarily explained by a single factor?

We formulate this challenge as the problem of finding submatrices close to rank one. Despite its potential, this topic has not been sufficiently addressed in the literature, and there exist virtually no algorithms for this purpose that are simultaneously effective, efficient and scalable.

We formalize the task as two problems which we characterize in terms of computational complexity, and propose efficient, scalable algorithms with approximation guarantees. Our experiments demonstrate how our approach can produce insightful findings in data, and show our algorithms to be superior to strong baselines.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

data mining, dimensionality reduction, variable selection, explainability

ACM Reference Format:
Bruno Ordozgoiti, Sachith Pai, and Marta Kołczyńska. 2021. Insightful Dimensionality Reduction with Very Low Rank Variable Subsets. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3450067

1 INTRODUCTION

In computer science, high dimensionality is both a blessing and a curse. A large number of variables generally implies that our data records are represented in high fidelity. However, as this number grows, the application of data mining methods rapidly becomes more challenging. As is well known, computations in high-dimensional vector spaces defy our intuition [2], and the amount of data required to acquire a representative sample increases exponentially in the number of dimensions [1]. The assortment of associated difficulties is so pervasive that it has earned the moniker of "curse of dimensionality".

For this and other reasons, significant research efforts have been devoted to the development of techniques to reduce the dimensionality of data, like Principal Component Analysis (PCA) [18] and non-linear alternatives like Kernel PCA [23] or autoencoders [16]. A drawback of these techniques is that they produce models in terms of abstract variables. This may make it hard for the practitioner to make sense of the results of their analysis. To alleviate this, different methods have been proposed, such as Sparse PCA, where each latent factor relates to a smaller number of variables [31], or nonnegative matrix factorization, where the factors have nonnegative entries exclusively [21]. The resulting factors remain, however, abstract and their meaning often unclear.

An alternative is the use of algorithms that preserve the original variables intact. In machine learning, this is known as feature selection. Given a data matrix, one way to accomplish this is to choose a subset of particularly representative columns. When the quality of representation is measured as the approximation error using a linear model and the Frobenius —or spectral— norm, this is known as the Column Subset Selection Problem (CSSP). The CSSP is in all likelihood a hard combinatorial optimization problem [7, 28] and approximation algorithms are known [4, 5, 15]. Closely related is the CUR decomposition, which approximately factors a matrix based on a subset of its rows and columns [22].

Despite increased interpretability with respect to the models discussed before, the CSSP has shortcomings of its own. Given a target rank k, an algorithm for this problem will output a model that approximately explains all of our variables in k dimensions. Suppose our data matrix has 100 variables, and the target rank is 20. The result will be a choice of 20 variables and a 20×100 matrix relating them to the rest. Such a model can remain difficult to read for domain experts, and might be large for computationally constrained systems.

Our approach. In this paper we study a related but fundamentally different approach to dimensionality reduction, with an emphasis on the interpretability of the results and the size of predictive models. In particular, we analyze the problem of finding groups of variables that can be approximately explained by a single *underlying factor*. In addition, we allow

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450067

	Math	Physics	Engines	Cars	Motorbikes	...
User 1	4	5	4	2	1	
User 2	5	5	4	1	0	
User 3	1	1	4	5	4	
User 4	0	1	5	4	5	
...						

Cluster 1: {Math, Physics, Engines}
Underlying factor: "Likes physics."
Cluster 2: {Cars, Motorbikes, Engines}
Underlying factor: "Likes racing."

Figure 1: An example illustrating the kind of results we hope to achieve, and why existing methods might be unsatisfactory. A database indicates how much user X is interested in topic Y. Two subsets are identified, both containing the topic "Engines", but for different reasons.

overlapping variable subsets to be part of the output. Conventional methods such as clustering and orthogonal factorization techniques often preclude or discourage overlap. We argue that this might result in a failure to uncover valuable insights. We now propose two examples to illustrate our goals.

Example 1. Suppose we have a database storing tastes of users, as depicted in Figure 1. Each entry represents how much the user in the corresponding row likes the topic in the corresponding column, from 0 to 5. We are interested in uncovering factors driving tastes. Users who like, say, physics, might enjoy the topics in cluster 1, which contains the topic "Engines". Users who like racing enjoy the topics in cluster 2, which also contains "Engines", but for different reasons. A partitioning-based clustering algorithm would allocate "Engines" to either cluster, resulting in an incomplete description of one of the factors. A similar problem might arise in independent factor analysis methods such as PCA. Our approach, however, can easily accommodate both factors with their overlapping topics.

Example 2. We additionally consider the case of data from social surveys. Social scientists are often interested in studying unobservable characteristics, such as values and attitudes, which cannot be measured directly, but can be inferred on the basis of the respondent's answers to survey items known to be tapping into the same construct. For example, questions such as "*Are immigrants good or bad for the country's economy?*", "*Do you think that the country's cultural life is enriched or undermined by immigrants?*", and "*Do you think that immigrants make the country a better or worse place to live?*" are used together as a scale of general support or opposition to migration. It is useful for practitioners to measure how strongly the latent traits manifest themselves in the results, or to be able to discover unexpected latent factors.

The first of these goals can be accomplished by (multi-group) confirmatory factor analysis (CFA), which measures how well the data can be explained by a predetermined latent factor structure [3, 13, 19, 29]. However, this confirmatory technique is not well suited for exploration. Meanwhile, an algorithm returning all subsets of features driven by a single factor would meet both goals. The proposed method could also be applied in survey design to omit some of the questions, as their answers could be inferred based on a few others.

Contribution. We propose a method to find strongly correlated sets of variables in data, with an emphasis on exploratory analysis and interpretability. To measure the quality of the sets we use *closeness-to-rank-one* (CRO) [20], the squared ratio between the spectral and Frobenius norms of a matrix. The literature is scarce in methods for exactly this purpose, and existing ones have disadvantages. The problem is often approached as a clustering task [6, 20]. This can be disadvantageous for the purpose of exploration and interpretation, as it rules out potentially interesting overlapping subsets, and the resulting partition can fail to highlight subsets with sufficiently high CRO, aside from being inefficient. Our contributions can be summarized as follows.

- We formalize the task of finding high-CRO subsets as two distinct problem statements, which we characterize in terms of computational complexity.
- We propose efficient, scalable and easily implemented algorithms with approximation guarantees.
- We propose several optimizations to facilitate the analysis of very high-dimensional data.
- Through various experiments we demonstrate that (1) finding high-CRO subsets is effective for building small predictive models based on feature selection, (2) our methods produce a rich diversity of high-quality results and (3) are as efficient (or more) as strong baselines.

2 BACKGROUND AND RELATED WORK

The problem of interpretable dimensionality reduction is central to machine learning and data mining. Some methods are based on matrix factorization techniques, such as latent semantic analysis [9], sparse PCA [31] and nonnegative matrix factorization [21]. A related method is sparse coding, which puts the focus on the sparsity of the representation rather than the number of underlying factors [26]. When these factors are learned, this is known as dictionary learning [11]. Factor analysis addresses the problem from a probabilistic standpoint, and can be seen as a generalization of methods such as PCA [25]. In general, these methods attempt to approximately express the input data as a function of a set of abstract factors, the interpretation of which is not always straightforward.

To circumvent this drawback, other methods have have been proposed, such as CUR decompositions [22], which select a subset of the matrix rows and columns, and column subset selection, [4, 5, 15]. These techniques are closely linked to rank-revealing QR factorizations, which permute the matrix columns so as to isolate the most representative ones [14, 17]. In our setting, these methods have several drawbacks. First, they factorize the entire matrix. If the number of employed factors is small, certain parts of the input matrix will dominate the results, while some variable sets whose relationships might be insightful could be neglected. Second, the models they produce can be large (e.g. $k \times n$ for k factors and n variables in the case of column subset selection), and thus hard to interpret.

In this paper we propose algorithms to locate variable sets that can be approximated well using a single factor. To this end,

we employ closeness-to-rank-one CRO [20], i.e. the squared ratio between the spectral and Frobenius norms of a matrix. To the best of our knowledge, the only method explicitly targeting CRO in the literature is the hierarchical clustering algorithm of Kim & Choi [20], which is inefficient and fails to find overlapping clusters. A method for a similar purpose was proposed by Boutsidis et al. [6], and consists essentially in clustering the matrix columns via k-means. This method can fail to produce subsets with satisfactory CRO unless the number of clusters is correctly selected. Furthermore, these methods offer no quality guarantees. Recent works approach similar problems, such as clustered matrix approximation [27].

3 PROBLEM STATEMENT

3.1 Notation

We will employ lowercase letters for vectors (v) and uppercase letters for matrices and sets (A, S). Context will be sufficient to avoid ambiguity. $\|A\|_F$ and $\|A\|_2$ denote, respectively, the Frobenius and spectral norm of matrix A. C^+ is the pseudo inverse of matrix C. $A^{(i)}$ is the i-th column of A. $(A\ B)$ is the concatenation of matrices A, B.

3.2 Preliminaries

Our goal is to find insightful relationships between variables. To motivate our work, we first discuss the limitations of some existing methods for this purpose. Dimensionality reduction techniques can be employed to improve the interpretability of our data. To avoid dealing with the abstract factors output by methods such as PCA, we can employ column subset selection algorithms. One way to formulate the Column Subset Selection Problem (CSSP) is as follows. Given a matrix A of n columns, find a matrix C comprised of a subset of $k < n$ of the columns of A such that $\|A - CC^+A\|_F^2$ is minimized. Here, CC^+A minimizes the above expression given C. In essence, the CSSP asks for a column subset that approximates the whole matrix well. Such a subset gives a compact representation of our data, and what is more, expressed in terms of variables we can understand.

The CSSP, however, has important drawbacks. The result will be a subset of k columns and a coefficient matrix C^+A of size $k \times n$. If k is small, the approximation will be poor. If it is large, it might be hard to elicit meaningful insights. We illustrate a second disadvantage with an example. Consider a matrix $A = (B\ C)$, with $\|C\| \gg \|B\|$ and $\text{rank}(C) \gg \text{rank}(B) \approx 1$. An algorithm to approximately factorize A with a matrix of rank $k \leq \text{rank}(C)$ will generally focus on recovering $\|C\|$, and will fail to identify the almost-rank-1 cluster B. To domain experts it might be interesting to learn that the variables contained in B are driven by a single factor. However, a CSSP algorithm might miss this fact.

To alleviate these shortcomings, instead of trying to factor the entire matrix we will look for sets of "well-clustered" variables. To formally define what we mean by "well-clustered", we employ the concept of closeness-to-rank-one [20].

Definition 1. *(CRO) Given a matrix A, we define its closeness-to-rank-one (CRO), denoted by $\rho(A)$, as*

$$\rho(A) = \|A\|_2^2 \|A\|_F^{-2}.$$

The CRO of a matrix determines how close it is to being rank-1, and thus provides a measure of how well its variables can be explained by a single factor.

A shortcoming of CRO is that it is sensitive to scale. If $\|u\| \gg \|C\|$ then $\rho((u\ C)) \approx 1$. In practice, one typically wants an algorithm that is not sensitive to scale. For this reason, we define a modified measure that addresses this problem.

Definition 2. *(No-CRO) Given a matrix A, let $\bar{A}$ denote A after normalizing its columns to unit norm. We define the normalized closeness-to-rank-one of A (No-CRO) as $\bar{\rho}(A) = \rho(\bar{A})$.*

3.3 Formulations

Now that we have established what constitutes a good variable subset, we set out to discover an algorithm that can find one.

In this paper we consider two approaches to finding variable subsets with high CRO: (1) given a target size, we want to find the best subset of exactly that size; (2) given a quality threshold, we want to find the largest subset with at least that quality. We formalize these two approaches below.

Problem 1. *Given an $m \times n$ matrix A and a natural number $k < n$, find the k-column subset with the largest CRO.*

Problem 2. *Given an $m \times n$ matrix A and a threshold $0 < \tau \leq 1$, find the largest column subset with CRO at least τ.*

These problems are hard to optimize.

Theorem 1. *Problems 1 and 2 are* **NP***-hard.*

Proof. We first treat problem 1. Consider an instance of Clique with n vertices and adjacency matrix A. Let Δ denote the maximum degree over the input graph's nodes, and define $W = A + \Delta I$. Note that W is positive semidefinite. We define $s(k) = \frac{\lambda_1(C+\Delta I)}{k\Delta}$, where C is the adjacency matrix of the k-clique. Observe that $s(k)$ is the maximum value of that ratio over adjacency matrices of graphs of size k. Thus, by solving problem 1 for each value of $k = 3, \ldots, n$ we would find the largest clique in the given instance.

We now turn to problem 2. Observe that $\rho(A) = \frac{\lambda_1(A^TA)}{tr(A^TA)}$. Thus, an algorithm that solves problem 2 finds the largest principal submatrix of a positive semidefinite matrix with a value of said ratio above the given threshold. Consider an instance of Clique with n vertices and adjacency matrix A. Let Δ denote the maximum degree over the input graph's nodes, and define $W = A + \Delta I$. Note that W is positive semidefinite. There are n possible clique sizes, and to each of them corresponds a values of $\frac{\lambda_1(C)}{tr(C)}$, where C is the principal submatrix of A corresponding to the clique. We define $s(k) = \frac{\lambda_1(C+\Delta I)}{k\Delta}$, where C is the adjacency matrix of the k-clique. If we could run an algorithm to solve problem 2 in polynomial time, we could run it n times on the given Clique instance, one for each value of $s(k), k = 3, \ldots, n$. Consider the i-th of these runs produces a matrix C_i. By lemma 1, the subgraph corresponding to C_i is guaranteed to contain a clique of size i. Furthermore, we can easily find such subgraph by removing vertices so as to preserve the value of at least $s(k)$. Observe that all non-clique graphs of size k have a value of this ratio smaller than $s(k)$. □

4 ALGORITHMS

In this section we propose algorithms to approximately optimize the aforementioned objectives. Our goal is to provide algorithms that are efficient enough for practical use, scalable and, taking our hardness results from the previous section into account, with quality guarantees.

We first address problem 1. We propose an efficient algorithm —Algorithm 1— with quality guarantees in the case where the columns of the input matrix are unit-normed. The algorithm greedily builds a subset for every column, and outputs the one with largest CRO. The design of this algorithm is based on an existential result for column subset selection which follows from a theorem due to Deshpande et al. [10]:

Theorem 2. *(Deshpande et al)[10] Let A be a matrix and A_k its best rank-k approximation. There exists a subset of k columns of A, forming matrix C, such that the projection of A onto the span of C satisfies*

$$\|A - CC^{+}A\|_F^2 \leq (k+1)\|A - A_k\|_F^2.$$

This is relevant in our context because the CRO measures the quality of the best rank-1 approximation. Therefore, the theorem guarantees that any column subset C can be approximated by one of its columns at a relative loss of at most $2(1-\rho(C))$. This leads to our approximation guarantee for Algorithm 1.

Theorem 3. *Given a matrix A with columns scaled to unit norm, containing a k-column submatrix C with $\rho(C) = \tau$, algorithm 1 outputs a k-column submatrix D with $\rho(D) \geq 2\tau - 1$.*

Proof. By expanding the term $\|A - A_k\|_F^2$ in Theorem 2, we easily conclude that there is a column v in C such that

$$\|C - (v^T v)^{-1} v v^T C\|_F^2 \leq 2(1-\tau)\|C\|_F^2.$$

Let D be the matrix corresponding to the set computed by Algorithm 1 for v. Since the columns of A are unit-norm,

$$\sum_i \frac{(v^T v)^{-1}\left(v^T D^{(i)}\right)^2}{\|D\|_F^2} \geq \sum_i \frac{(v^T v)^{-1}\left(v^T C^{(i)}\right)^2}{\|C\|_F^2},$$

which implies

$$\frac{\|D - (v^T v)^{-1} v v^T D\|_F^2}{\|D\|_F^2} \leq \frac{\|C - (v^T v)^{-1} v v^T C\|_F^2}{\|C\|_F^2} \leq 2(1-\tau). \tag{1}$$

Let u denote the top left singular vector of C. Note that we can express the CRO of C as follows:

$$\rho(C) = 1 - \frac{\|C - uu^T C\|_F^2}{\|C\|_F^2} = \sum_i \frac{(u^T C^{(i)})^2}{\|C\|_F^2}. \tag{2}$$

So we can write

$$\rho(D) \geq \sum_i \frac{\left(v^T D^{(i)}\right)^2}{\|D\|_F^2} = \frac{\|(v^T v)^{-1} v v^T D\|_2^2}{\|D\|_F^2}.$$

From Eq. (1) we have

$$\frac{\|(v^T v)^{-1} v v^T D\|_2^2}{\|D\|_F^2} \geq \frac{\|(v^T v)^{-1} v v^T C\|_2^2}{\|C\|_F^2} \text{ and}$$

$$1 - \frac{\|(v^T v)^{-1} v v^T C\|_F^2}{\|C\|_F^2} \leq 2(1-\tau).$$

Thus, simple manipulations yield $\rho(D) \geq 2\tau - 1$. □

Algorithm 1 Best-k

Input: matrix A

Compute $W = A^T A$
for $i = 1 \ldots n$ **do**
  $S_i \leftarrow \{i\}$
  while $|S_i| < k$ **do**
    $S_i \leftarrow S_i \cup \text{argmax}_{j \notin S_i} W_{ij}^2 / W_{jj}$
  end while
end for
Output S_i maximizing $\rho(A_{S_i})$

Algorithm 2 Largest

Input: matrix A, threshold τ

Compute $W = A^T A$
for $i = 1 \ldots n$ **do**
  $S_i \leftarrow \{i\}$
  while $\sum_{j \in S_i} W_{ii}^{-2} W_{ij}^2 / \|A_{S_i}\|_F^2 \geq \tau$ **do**
    $S_i \leftarrow S_i \cup \text{argmax}_{j \notin S_i} W_{ij}^2 / W_{jj}$
  end while
  Remove last item added to S_i to ensure $\rho(S_i) \geq \tau$
end for
Output $\{S_i\}_{i=1}^n$

We now adress problem 2. As opposed to problem 1, which asks for a subset of size exactly k, problem 2 puts no such restriction. Thus, we focus on efficiently finding a subset with quality over the given threshold. To do this, as before we will build one subset per column. However, now we keep increasing its size until we are no longer above the threshold. Since computing the CRO for each visited subset would be costly, we rely on a lower bound. In particular, observe that for any matrix C and any of its columns, say v, it is

$$\rho(C) = \sum_i \frac{\left(u^T C^{(i)}\right)^2}{\|C\|_F^2} \geq \sum_i \frac{\left((v^T v)^{-1} v^T C^{(i)}\right)^2}{\|C\|_F^2}.$$

This bound is straightforward to compute as we incrementally build the subsets, and it makes it unnecessary to successively compute the CRO. Thus, it results in significant computational savings while guaranteeing the output of subsets with CRO at least τ. The procedure is summarized as Algorithm 2.

Our algorithms have several advantages. First, they can be easily and efficiently implemented resorting to established software packages for matrix operations. Second, they are fully parallelizable over data columns, so it is possible to deal with

very high-dimensional data efficiently. Third, the fact that all computations rely on the entries of the matrix A^TA makes it straightforward to incorporate kernel functions into the process, which can be useful for revealing non-linear factors.

4.1 Setting the subset size.

When dealing with problem 1 in practice, it might be of interest to estimate what might be a good choice for k. If we can afford to compute the singular values of our matrix, we can derive a helpful inequality.

Proposition 1. *Given a matrix A of rank n, the best possible CRO of a column subset of size k is upper-bounded by*

$$1 - \frac{\sum_{i=n-k+2}^{n} \sigma_i^2(A)}{\|C\|_F^2},$$

where C is the submatrix of A composed by the k columns of largest norm.

Proof. The interlacing inequalities of singular values guarantee that $\sigma_i(C) \geq \sigma_{n-k+i}(A)$. The result follows easily from

$$\rho(C) = 1 - \frac{\sum_{i=2}^{k} \sigma_i^2(C)}{\|C\|_F^2}.$$

□

4.2 Setting the quality threshold.

Another question that might arise in practical settings is this: when addressing problem 2, what should τ be? We show we can efficiently obtain the maximum possible value of this parameter. First, we show that given any matrix, we can find a subset of its columns with equal or better CRO.

Lemma 1. *Given a matrix A of n columns, there exists a submatrix S of $n-1$ columns of A such that $\rho(S) \geq \rho(A)$.*

Proof. Let A_i denote the matrix resulting from removing the i-th column from matrix A. We can consider two cases:

(1) The following holds for all i: the dominant left singular vector of A_i is u_1, that is, it remains unchanged.
(2) The dominant left singular vector of A_i is different from u_1 for at least one $i \in \{1, \ldots, n\}$.

Let $\beta_{1j} = \sigma_1^2(A) - \sigma_1^2(A_j)$. In the first case, we can write

$$\sum_j \beta_{1j} = \sum_j \sigma_1^2(A) - \sigma_1^2(A_j) = \sum_j \sigma_1^2 - \|u_1^T A_j\|_2^2.$$

In the second case, we have the following inequality:

$$\sum_j \beta_{1j} \leq \sum_j \sigma_1^2(A) - \|u_1^T A_j\|_2^2.$$

This follows from the fact that $\sigma_1^2(A_j)$ maximizes $\|x^T A_j\|_2^2$ over all unit vectors x. In addition, it is

$$\sum_j \sigma_1^2 - \|u_1^T A_j\|_2^2 = \sigma_1^2.$$

This follows from the fact that $\|u_1^T A_j\|_2^2$ is equal to σ_1^2 minus the "contribution" of v_j (the removed column) to it, that is

$$\|u_1^T A_j\|_2^2 = \sigma_1^2 - (u_1^T v_j)^2$$
$$\Leftrightarrow \sum_j \|u_1^T A_j\|_2^2 = n\sigma_1^2 - \|u_1^T A\|_2^2 = (n-1)\sigma_1^2.$$

So we finally have $\sum_j \beta_{1i} \leq \sigma_1^2$. Note that this holds with equality in the first case. Now, let us denote the i-th column of A as v_i. Assume that the removal of column v_i results in a subset with smaller CRO. Then,

$$\frac{\beta_{1j}}{\sigma_1^2} > \frac{\|v_j\|_2^2}{\|A\|_F^2}.$$

In contradiction with the statement of the proposition, assume the above holds for all i. Then we have $\sum_j \beta_{1j} > \sigma_1^2$. However, this contradicts inequality 4.2 (and the corresponding equality for the first case), and is thus absurd. □

It follows that the maximum possible CRO of a column subset of A (of more than one column) is attained by a 2-column subset. Therefore, an upper bound on the attainable CRO can be obtained by computing this value for all column pairs (and as we will show in section 4.3, we do not even need to compute all pairs). The next results show that in the case of No-CRO, we can benefit from additional computational gains.

Lemma 2. *Let A be a matrix comprised of two columns, denoted x and y respectively. If $\|x\| = \|y\| = 1$ then*

$$\bar{\rho}(A) = \left(\frac{(1 + sgn(x^Ty)x^Ty)}{\sqrt{2(1 + sgn(x^Ty)(x^Ty))}} \right)^2$$

Proof. Let u denote the first left singular vector of A. Since x and y are unit-normed, it is

$$u = \frac{x + sgn(x^Ty)y}{\|x + sgn(x^Ty)y\|}.$$

Note that $\bar{\rho}(A) = \frac{\|uu^Tv\|_2^2 + \|uu^Tv\|_2^2}{2}$. If we denote $s = sgn(x^Ty)$, we have

$$\|uu^Tx\|_2^2 + \|uu^Ty\|_2^2 = x^Tuu^Tx + y^Tuu^Ty$$
$$= (x^Tu)^2 + (y^Tu)^2 = \left(\frac{x^T(x+sy)}{\|x+sy\|} \right)^2 + \left(\frac{y^T(x+sy)}{\|x+sy\|} \right)^2$$
$$= 2\left(\frac{1 + sx^Ty}{\|x+sy\|} \right)^2 = 2\left(\frac{(1+sx^Ty)}{\sqrt{2+2s(x^Ty)}} \right)^2.$$

□

So given a matrix A, we can easily obtain the No-CRO for all pairs of columns by computing the product A^TA.

4.3 Bounding CRO pairs.

For the purpose of our algorithms, we can benefit from further computational savings by addressing the following question: given three vectors u, v, w and the values of u^Tv, u^Tw, can we say something about v^Tw? The answer is given by the following result.

Lemma 3. *Given three vectors u, v, w, define*

$$\beta_{u,v,w} = \sqrt{\left(\|v\|_2^2 - \left(\frac{u^T v}{\|u\|}\right)^2\right)\left(\|w\|_2^2 - \left(\frac{u^T w}{\|u\|}\right)^2\right)}.$$

Then

$$v^T w \in \left[\frac{u^T v u^T w}{\|u\|_2^2} - \beta_{u,v,w}, \frac{u^T v u^T w}{\|u\|_2^2} + \beta_{u,v,w}\right].$$

Proof. W.l.o.g. we can rotate all three vectors so that u is aligned with the first vector of the canonical basis of $\mathbb{R}^n$, i.e. $u = \alpha e_1$ for some $\alpha \in \mathbb{R}$. Then, $v_1 = \|u\|_2^{-1} u^T v$ and $w_1 = \|u\|_2^{-1} u^T w$. It then becomes clear that $w^T v$ is maximized when $v_{2:n} \propto w_{2:n}$, and minimized when $v_{2:n} \propto -w_{2:n}$, bounded using the Cauchy-Schwarz inequality by the values given in the statement of the result. □

To illustrate how these bounds can be employed we propose an example. Suppose we compute $u^T A$, yielding the inner products of u with the rest of the matrix columns. The value of $u^T v$ is large, while $u^T w$ is small. The upper bound stated above is thus small. As a result, we conclude that the columns v, w do not constitute a high-CRO submatrix and therefore we can safely discard the pair in all subsequent computations.

4.4 Column-based approximations.

The proposed approach is well-suited to exploratory applications, as it produces sets of variables that can be explained well using a single linear factor. However, it is still conceivable that some domain experts will prefer models formulated in terms of the actual variables, rather than an abstract factor. Fortunately, the design of our algorithms implies that we can accomplish this goal with guarantees. To argue why, we first state a result from the column subset selection literature.

Theorem 4. *(Guruswami-Sinop)[15] Given $A \in \mathbb{R}^{m \times n}$, and positive integers $k \leq r$, one can find a set C of r columns, deterministically using at most $O(rnm^\omega \log m)$ many arithmetic operations (where $\omega$is the exponent of matrix multiplication), such that*

$$\|A - CC^+ A\|_F^2 \leq \frac{r+1}{r+1-k}\|A - A_k\|_F^2,$$

where A_k is the best rank-k approximation of A.

This translates to our setting as follows. Consider a matrix C, formed by a column-subset of A, satisfying $1 - \rho(C) \leq \epsilon$. This means $\|C\|_F^2 - \sigma_1^2(C) \leq \epsilon\|C\|_F^2$. Suppose we run algorithm 2 with $\tau = 1 - \epsilon$ and obtain subsets $\{S_i\}_{i=1}^n$, each of them forming a matrix $C_i = A_{S_i}$. We can then run the algorithm of Guruswami-Sinop [15] on each C_i to obtain r_i columns of C_i, say R_i, satisfying

$$\|C_i - C_{R_i} C_{R_i}^+ C_i\|_F^2 \leq (1 + 1/r)\,\epsilon\|A_{S_i}\|_F^2.$$

That is, from each subset we can pick a small number of columns and obtain small multiplicative error with respect to the best rank-1 approximation. This means that if we set the target CRO to be high, we can approximate each variable subset with a model based on a few of the original variables.

4.5 Stability.

In applications, one usually deals with noisy or missing data. In case of the latter, the problem may be partially overcome by means of imputation. How sensitive is CRO to perturbations? Will the results change significantly depending on the imputation strategy? We address these questions.

The CRO is the reciprocal of the *stable rank* [8], a robust variant of the matrix rank. While the rank is integral and can change abruptly, the stable rank changes smoothly, in proportion to the magnitude of the perturbation.

This can be formalized resorting to elementary results from perturbation theory. Let $\tilde{A} = A + E$ be a perturbed realization of A. A classical inequality due to Weyl [30] guarantees $(\sigma_1(A) - \sigma_1(\tilde{A}))^2 \leq \|E\|^2$. Thus, $\sigma_1(A) - \|E\|_2 \leq \sigma_1(\tilde{A}) \leq \sigma_1(A) + \|E\|_2$.

Furthermore, it is straightforward to show that $\|A + E\|_F^2 = \|A\|_F^2 + \|E\|_F^2 + 2tr(A^T E)$. Following John Von Neumann [24] we assert $|tr(A^T E)| \leq \sum_i \sigma_i(A)\sigma_i(E)$, and thus

$$\frac{\sigma_1(A) - \|E\|_2}{\|A\|_F^2 + \|E\|_F^2 + 2\sum_i \sigma_i(A)\sigma_i(E)} \leq \rho(\tilde{A}) \leq \frac{\sigma_1(A) + \|E\|_2}{\|A\|_F^2 + \|E\|_F^2 - 2\sum_i \sigma_i(A)\sigma_i(E)}.$$

That is, if the perturbation is small, then the value of the CRO will not change substantially. The practical implications are significant. For instance, if we have to deal with a few missing data, the choice of imputation strategy will not have a strong impact on the results of our methods.

4.6 Complexity analysis and scalability.

Our algorithms require computing a matrix product in time $O(mn^2)$, then $nT_{top(k,n)}$ computations, where $T_{top(k,n)}$ is the time required to find the top k items in a list of size n. For the first step, we can benefit from existing efficient and parallelized implementations for matrix multiplication. The second phase can be fully parallelized over the matrix columns, bringing the second stage down to $\frac{n}{p}T_{top(k,n)}$ for p parallel processors.

This compares favourably to the algorithm of Kim & Choi, as the CRO for each cluster pair can be computed in parallel, but the merging sequence needs to be computed sequentially.

4.7 A greedy alternative.

Our algorithms are efficient and, as we will show, scale to data sets with thousands of dimensions. However, in the presence of millions of columns, running our algorithms can be too demanding for some practical applications, which might require near-instant response. While random sampling can aid in sidestepping this issue, practitioners might be averse to discarding data in some settings.

Fortunately, our ideas can be combined with existing approaches to obtain more efficient algorithms. Here we describe one example. We consider column subset selection, which as explained in the introduction is the well-known problem of choosing the best c columns to approximate the rest of the input matrix. We can find a CRO subset of size k by solving the algorithm with $c = 1$, and then forming a subset with the output column and the $k - 1$ columns that are best approximated

by it. We can find successive sets by repeating this process, either removing only the first column to allow overlap, or the k to avoid it. As we will show in our experiments, this method can be faster than our main algorithms, but provides slightly worse results in general.

5 EXPERIMENTS

We validate our algorithms through various experiments on different data sets. We wish to evaluate the efficiency of our algorithms, the quality of their output and their validity as a tool for exploratory analysis and dimensionality reduction.

Data. We employ the following publicly available data sets.

- `Crime`[1]: Socioeconomic, crime and law enforcement data on different U.S. communities.
- `S&P`[2]: Historical stock data for S&P 500 companies.
- `Gas`[3]: Measurements from 16 chemical sensors exposed to 6 different gases at various concentration levels.
- `Gene`[4]: Random extraction of gene expressions of patients having different types of tumor
- `Opportunity`[5]: Human Activity Recognition from Wearable, Object, and Ambient Sensors
- `URLRep`[6] For scalability tests. We sparsify the matrix W by removing entries smaller than 0.2.

We scale input matrices so that columns are unit-norm, and remove redundant or non-varying columns.

Baselines. To the best of our knowledge, our problem formulations are novel and no algorithms have been proposed in the literature to tackle them. Therefore, we design strong baselines based on existing methods for similar purposes.

- KIM&CHOI: the hierarchical clustering algorithm of Kim & Choi [20], which first creates one cluster per column, and iteratively merges the cluster pair with highest CRO among all possible pairs. To obtain a subset of size k, we run the merging process until one such cluster is found. We use our own Python implementation.
- KMEANS: the algorithm by Boutsidis et al. [6], which runs k-means clustering on the data columns and then selects columns from each cluster. The algorithm takes the number of clusters c as a parameter. To obtain a subset of size k, we run it for decreasing values of $c = \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \ldots$ until one such cluster is obtained. If the cluster is larger, we trim it to exactly k columns. For the k-means step we use the Python scikit-learn library.

Our algorithms were implemented in Python, using matrix operations and parallelization where possible[7]. We will also employ the greedy alternative described in section 4.7. For the column subset selection part we use the algorithm described by Farahat et al. [12]. In particular, we use a Python rewrite of the authors' Matlab implementation. We will refer to this method as GCSS.

Dataset	Train	Test	Variables
`Crime`	1772	443	125
`S&P`	1007	252	471
`Gas`	10310	3600	128
`Gene`	640	161	20532
`Opportunity`	695499	173888	135
`URLRep`	2396130	-	928809

All code was run on a server with 16 Intel Xeon Gold 6248 CPU cores @ 2.50GHz and 8 GB of RAM per core. We will first evaluate the performance of the different methods in terms of output quality and running time, and then choose the best-performing ones for subsequent tests.

5.1 Quality of the results.

We first evaluate the quality of the best subset found by each method. We set $k = 2, 4, 8, 16, 32, 64$. For the higher-dimensional dataset `Gene` we set k up to 2048. The results are shown in figure 2 (top). All methods manage to find good CRO subsets, but our algorithm is consistently the best one. Interestingly, our GCSS-based method is a close second. We omit KIM&CHOI and KMEANS on `Gene`, as these methods took too long to complete and are thus not competitive on high-dimensional data.

5.2 Running time.

Figure 2 (bottom) shows running times. Our method is almost always the fastest. In the case of `Gene`, GCSS is slightly faster, although the running times of both methods are close.

5.3 Finding multiple low-rank subsets.

To further evaluate the ability of the algorithms to find CRO subsets, we select the two best-performing ones (BEST-K and GCSS) from the previous experiment and test their ability to find multiple such subsets. We first fix k and run each method. We then take the best located subset, remove the corresponding columns from the data, and proceed to run the algorithm again. In figure 3 we report the CRO of the successively found subsets. We also report averages over all these successive runs. The subsets found by our method are of smoothly decreasing quality, while GCSS is very unstable, and in some cases fails to find top-quality ones when they are available. Notice e.g. the sharp drops in `Gas`, $k = 4$, when our method is still able to find subsets of CRO above 0.98.

5.4 Applications

We now consider some possible applications of our algorithm.

5.4.1 Feature selection on a budget. We consider the case where computational and storage capabilities restrict the size of predictive models. As an illustration, consider we deploy n sensors. After some time collecting the readings, we need to deploy a new sensor grid, but due to a low budget we decide to retrieve d of the original ones, so they can be part of the new deployment. To select d sensors we decide to pick the

[1] http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized
[2] https://www.kaggle.com/camnugent/sandp500/data
[3] http://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations
[4] http://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq
[5] http://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition
[6] http://archive.ics.uci.edu/ml/datasets/URL+Reputation
[7] Source code: https://osf.io/h6dqj/?view_only=a1aa466e2d0a4e308e82f19763c43611.

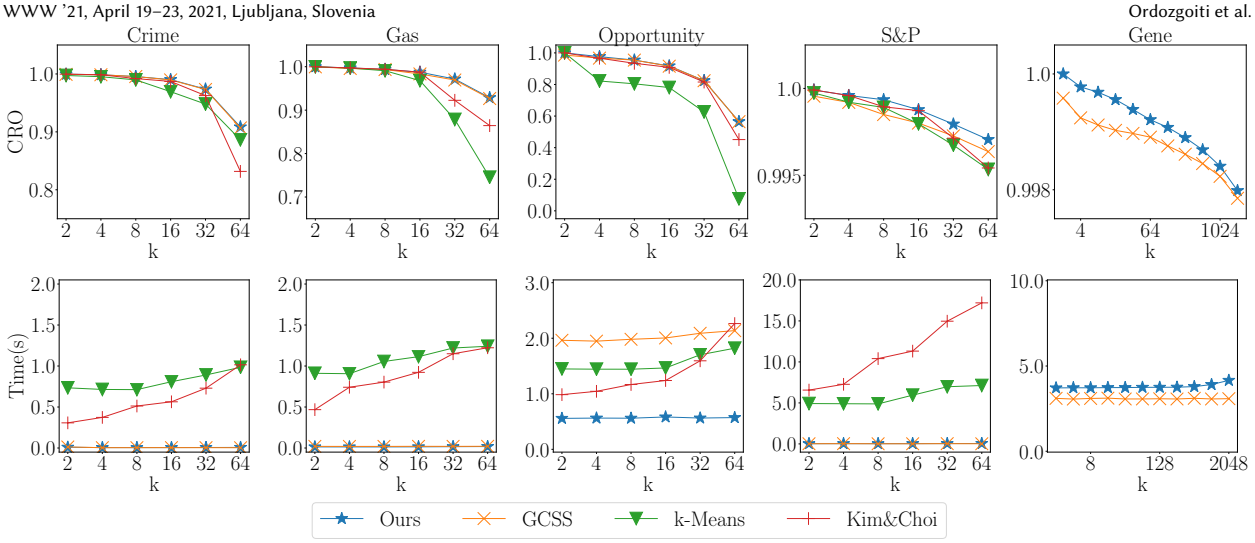

Figure 2: Top: CRO of best located subsets. Bottom: running time.

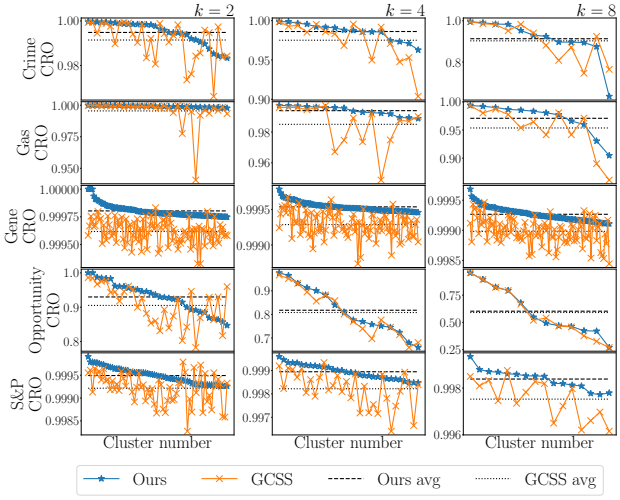

Figure 3: CRO of successively located subsets. We find different numbers of clusters depending on dimensionality and k, from 12 (`Crime`, $k = 8$) to 100 (`Gene`).

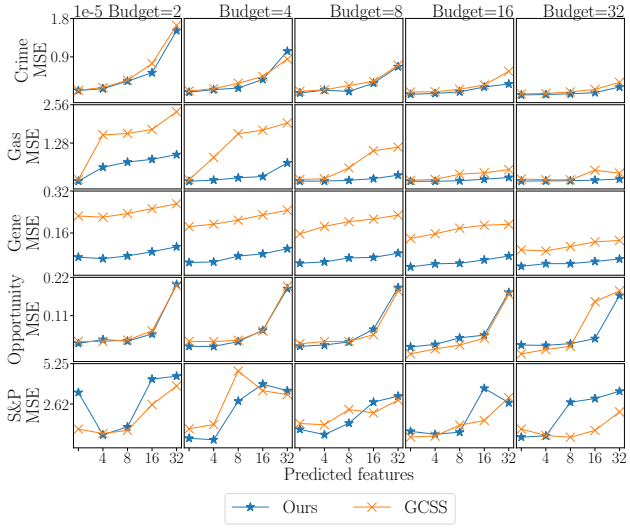

Figure 4: Prediction error.

most redundant ones, that is, those whose readings can be best predicted using the readings of others. Suppose the predictive models are to be deployed on low-capacity systems, and must thus be small, say of size k, not much larger than 1. We could accomplish this using a column subset selection method to find k sensors. However, the resulting model would be the best to predict the remaining $n - k$ sensors, while we are only interested in predicting d of them. A high-CRO subset of size $k + d$ is likely to be the better option, as it simultaneously reveals a subset of d predictable sensors and a good set of k sensors to predict their values.

We evaluate this task as follows. For each data set we consider combinations of budget (k, number of columns to build the predictive model with) and the number of predicted columns (d), both taking values in $\{2, 4, 8, 16, 32\}$. For Best-k, we find a CRO subset of size $k + d$ and then use column subset selection to choose which ones will conform the predictive model (exhaustively, when the subset is small enough, greedily otherwise). For GCSS, we run the algorithm to find k columns and then choose the d ones that are best predicted.

We report the mean squared error (MSE) of the predicted columns in Fig. 4. First, note how the idea to use CRO subsets to perform feature selection with very small predictive models is effective, as the errors are small. Our method generally does better, sometimes by a wide margin, with some exceptions. Note that as k grows, GCSS is expected to do better.

Table 1: A four-question subset corresponding to trust in politics found by LARGEST. Below, left, countries where this subset has CRO $\geq$ 0.75. Right, countries where the subset excludes Trust in the EP, even after lowering τ to 0.7.

Trust in (1) country's parliament, (2) politicians, (3) political parties and (4) European Parliament (EP).	
Include trust in EP	Exclude trust in EP
Belgium, Denmark, Finland, France, Netherlands, Slovenia	Spain, Ireland, Poland, Portugal

5.4.2 Case study: exploratory analysis of survey data. The European Social Survey (ESS[8]) is a cross-national survey that collects data on a variety of attitudes and opinions towards social, economic, and political issues. In these surveys, each social and political attitude, a latent construct, is measured with a set of questions designed to form a consistent scale of, e.g., attitudes toward immigrants or political trust. Differences in the composition of factors in different countries provide useful information about the content of attitudes., and an interesting starting point for future research. Because the questions in cross-national surveys are a priori designed such that responses are strongly correlated, our algorithms can be employed to determine the extent to which the one-factor structures expected by the designers are present in the data.

We run LARGEST with $\tau = 0.75$ and compare the results across 23 countries covered by the ESS Round 8. One of the clearest attitudes is trust in political institutions, such as the parliament, legal system, political parties, the European Parliament, etc. The factor is identified in all countries. However, in some EU-member states trust in the European Parliament is part of the "political trust" factor, which suggests that EU institutions are perceived as part of the country's politics on par with the national parliament. In other countries, primarily in Central and Southern Europe, trust in EP is not part of the trust in institutions factor, which suggests lower identification with the European Community. This is illustrated in table 1. The results suggest that the proposed algorithms can be useful for exploratory analysis, as they can reveal strong latent component without resorting to prespecified factor structure, as required, for instance, by Factor-Analysis-based methods.

5.5 Scalability.

We evaluate BEST-K on random samples of URLRep. For this experiment we employed 64 cores. If there are high-CRO subsets, hitting one of their columns should be enough to uncover them, or at least subsets of similar quality. We set $k = 4, 8, 12, 16$ and sample up to 20% of the rows of W. The results in Fig. 5, averaged over 10 runs, show we can find good subsets in just a few seconds, by sampling between 5% and 20% of the columns. The cases $k = 12, 16$ likely require larger samples, as large, good subsets are less likely to exist.

[8]https://www.europeansocialsurvey.org/

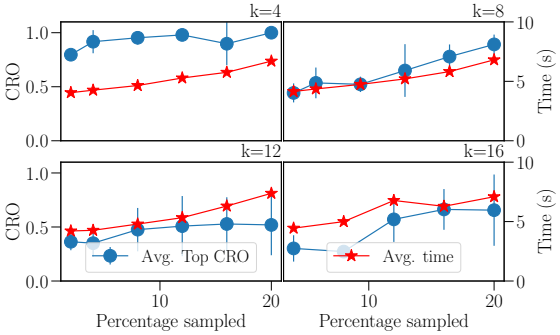

Figure 5: Results on URLRep.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have studied the problem of finding subsets of variables that can be approximated by a single factor. We approach this task as the problem of finding submatrices close to rank one. We have proposed two problem formulations, which we have characterized in terms of computational complexity, and proposed efficient, scalable algorithms with approximation guarantees. Our experiments on a variety of real data sets show how the proposed methods run efficiently and produce results of high quality. In the future it would be interesting to improve the approximation guarantees if possible, and to identify other application domains. In addition, we intend to study the use of kernel matrices to compute inner products between columns, so as to find non-linearly low-rank subsets.

ACKNOWLEDGMENTS

This work was supported by the Academy of Finland project AIDA (317085), the EC H2020RIA project "SoBigData++" (871042), and the Polish National Agency for Academic Exchange within the Bekker programme, number PPN/BEK/2019/1/00133.

REFERENCES

[1] Richard E Bellman. 2015. *Adaptive control processes: a guided tour.* Princeton university press.
[2] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful?. In *International conference on database theory.* Springer, 217–235.
[3] Kenneth A. Bollen. 1989. *Structural Equations with Latent Variables.* John Wiley & Sons, Inc., Hoboken, NJ, USA. 432–447 pages. https://doi.org/10.1002/9781118619179
[4] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. 2014. Near-optimal column-based matrix reconstruction. *SIAM J. Comput.* 43, 2 (2014), 687–717.
[5] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. 2009. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms.* Society for Industrial and Applied Mathematics, 968–977.
[6] Christos Boutsidis, Jimeng Sun, and Nikos Anerousis. 2008. Clustered subset selection and its applications on it service metrics. In *Proceedings of the 17th ACM conference on Information and knowledge management.* ACM, 599–608.
[7] Ali Çivril. 2014. Column subset selection problem is ug-hard. *J. Comput. System Sci.* 80, 4 (2014), 849–859.
[8] Michael B Cohen, Jelani Nelson, and David P Woodruff. 2015. Optimal approximate matrix product in terms of stable rank. *arXiv preprint arXiv:1507.02268* (2015).
[9] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391.

[10] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. 2006. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 1117–1126.

[11] Michael Elad and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing* 15, 12 (2006), 3736–3745.

[12] Ahmed K Farahat, Ali Ghodsi, and Mohamed S Kamel. 2011. An efficient greedy method for unsupervised feature selection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 161–170.

[13] Brian F. French and W. Holmes Finch. 2008. Multigroup Confirmatory Factor Analysis: Locating the Invariant Referent Sets. *Structural Equation Modeling: A Multidisciplinary Journal* 15, 1 (jan 2008), 96–113. https://doi.org/10.1080/10705510701758349

[14] Ming Gu and Stanley C Eisenstat. 1996. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing* 17, 4 (1996), 848–869.

[15] Venkatesan Guruswami and Ali Kemal Sinop. 2012. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1207–1214.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.

[17] Yoo Pyo Hong and C-T Pan. 1992. Rank-revealing QR factorizations and the singular value decomposition. *Math. Comp.* 58, 197 (1992), 213–232.

[18] Ian Jolliffe. 2002. *Principal component analysis*. Wiley Online Library.

[19] Karl G Jöreskog. 1969. A general approach to confirmatory maximum likelihood factor analysis. *Psychometrika* 34, 2 (1969), 183–202.

[20] Yong-Deok Kim and Seungjin Choi. 2007. A method of initialization for nonnegative matrix factorization. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Vol. 2. IEEE, II–537.

[21] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788.

[22] Michael W Mahoney and Petros Drineas. 2009. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences* 106, 3 (2009), 697–702.

[23] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. 1999. Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*. 536–542.

[24] Leon Mirsky. 1975. A trace inequality of John von Neumann. *Monatshefte für mathematik* 79, 4 (1975), 303–306.

[25] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[26] Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research* 37, 23 (1997), 3311–3325.

[27] Berkant Savas and Inderjit S Dhillon. 2016. Clustered Matrix Approximation. *SIAM J. Matrix Anal. Appl.* 37, 4 (2016), 1531–1555.

[28] Yaroslav Shitov. 2017. Column subset selection is NP-complete. *arXiv preprint arXiv:1701.02764* (2017).

[29] L L Thurstone. 1947. *Multiple-factor analysis; a development and expansion of The Vectors of Mind*. University of Chicago Press, Chicago, IL, US. xix, 535–xix, 535 pages.

[30] Hermann Weyl. 1912. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Math. Ann.* 71, 4 (1912), 441–479.

[31] Hui Zou, Trevor Hastie, and Robert Tibshirani. 2006. Sparse principal component analysis. *Journal of computational and graphical statistics* 15, 2 (2006), 265–286.