A Fast Greedy Algorithm for Generalized Column Subset Selection

Ahmed K. Farahat, Ali Ghodsi, and Mohamed S. Kamel
University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
{afarahat,aghodsib,mkamel}@uwaterloo.ca

Abstract

This paper defines a generalized column subset selection problem which is concerned with the selection of a few columns from a source matrix A that best approximate the span of a target matrix B. The paper then proposes a fast greedy algorithm for solving this problem and draws connections to different problems that can be efficiently solved using the proposed algorithm.

1 Generalized Column Subset Selection

The Column Subset Selection (CSS) problem can be generally defined as the selection of a few columns from a data matrix that best approximate its span [2–5, 10, 15]. We extend this definition to the generalized problem of selecting a few columns from a source matrix to approximate the span of a target matrix. The generalized CSS problem can be formally defined as follows:

Problem 1 (Generalized Column Subset Selection) *Given a source matrix $A \in \mathbb{R}^{m \times n}$, a target matrix $B \in \mathbb{R}^{m \times r}$ and an integer l, find a subset of columns $\mathcal{L}$ from A such that $|\mathcal{L}| = l$ and*

$$\mathcal{L} = \arg\min_{\mathcal{S}} \|B - P^{(\mathcal{S})}B\|_F^2,$$

where $\mathcal{S}$ is the set of the indices of the candidate columns from A, $P^{(\mathcal{S})} \in \mathbb{R}^{m \times m}$ is a projection matrix which projects the columns of B onto the span of the set $\mathcal{S}$ of columns, and $\mathcal{L}$ is the set of the indices of the selected columns from A.

The CSS criterion $\mathbf{F}(\mathcal{S}) = \|B - P^{(\mathcal{S})}B\|_F^2$ represents the sum of squared errors between the target matrix B and its rank-l approximation $P^{(\mathcal{S})}B$. In other words, it calculates the Frobenius norm of the residual matrix $F = B - P^{(\mathcal{S})}B$. Other types of matrix norms can also be used to quantify the reconstruction error [2, 3]. The present work, however, focuses on developing algorithms that minimize the Frobenius norm of the residual matrix. The projection matrix $P^{(\mathcal{S})}$ can be calculated as $P^{(\mathcal{S})} = A_{:\mathcal{S}} \left(A_{:\mathcal{S}}^T A_{:\mathcal{S}}\right)^{-1} A_{:\mathcal{S}}^T$, where $A_{:\mathcal{S}}$ is the sub-matrix of A which consists of the columns corresponding to $\mathcal{S}$. It should be noted that if $\mathcal{S}$ is known, the term $\left(A_{:\mathcal{S}}^T A_{:\mathcal{S}}\right)^{-1} A_{:\mathcal{S}}^T B$ is the closed-form solution of least-squares problem $T^* = \arg\min_T \|B - A_{:\mathcal{S}}T\|_F^2$.

2 A Fast Greedy Algorithm for Generalized CSS

Problem 1 is a combinatorial optimization problem whose optimal solution can be obtained in $O\left(\max\left(n^l mrl, n^l ml^2\right)\right)$. In order to approximate this optimal solution, we propose a fast greedy algorithm that selects one column from A at a time. The greedy algorithm is based on a recursive formula for the projection matrix $P^{(\mathcal{S})}$ which can be derived as follows.

Lemma 1 *Given a set of columns $\mathcal{S}$. For any $\mathcal{P} \subset \mathcal{S}$, $P^{(\mathcal{S})} = P^{(\mathcal{P})} + R^{(\mathcal{R})}$, where $R^{(\mathcal{R})} = E_{:\mathcal{R}} \left(E_{:\mathcal{R}}^T E_{:\mathcal{R}}\right)^{-1} E_{:\mathcal{R}}^T$ is a projection matrix which projects the columns of $E = A - P^{(\mathcal{P})}A$ onto the span of the subset $\mathcal{R} = \mathcal{S} \setminus \mathcal{P}$ of columns.*

Proof Define $D = A_{:\mathcal{S}}^T A_{:\mathcal{S}}$. The projection matrix $P^{(\mathcal{S})}$ can be written as $P^{(\mathcal{S})} = A_{:\mathcal{S}} D^{-1} A_{:\mathcal{S}}^T$. Without loss of generality, the columns and rows of $A_{:\mathcal{S}}$ and D can be rearranged such that the first sets of rows and columns correspond to $\mathcal{P}$. Let $S = D_{\mathcal{R}\mathcal{R}} - D_{\mathcal{P}\mathcal{R}}^T D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}}$ be the Schur complement [17] of $D_{\mathcal{P}\mathcal{P}}$ in D, where $D_{\mathcal{P}\mathcal{P}} = A_{:\mathcal{P}}^T A_{:\mathcal{P}}$, $D_{\mathcal{P}\mathcal{R}} = A_{:\mathcal{P}}^T A_{:\mathcal{R}}$ and $D_{\mathcal{R}\mathcal{R}} = A_{:\mathcal{R}}^T A_{:\mathcal{R}}$. Using the block-wise inversion formula [17], D^{-1} can be calculated as

$$D^{-1} = \begin{bmatrix} D_{\mathcal{P}\mathcal{P}}^{-1} + D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}} S^{-1} D_{\mathcal{P}\mathcal{R}}^T D_{\mathcal{P}\mathcal{P}}^{-1} & -D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}} S^{-1} \\ -S^{-1} D_{\mathcal{P}\mathcal{R}}^T D_{\mathcal{P}\mathcal{P}}^{-1} & S^{-1} \end{bmatrix}$$

Substituting with $A_{:\mathcal{S}}$ and D^{-1} in $P^{(\mathcal{S})} = A_{:\mathcal{S}} D^{-1} A_{:\mathcal{S}}^T$, the projection matrix can be simplified to

$$P^{(\mathcal{S})} = A_{:\mathcal{P}} D_{\mathcal{P}\mathcal{P}}^{-1} A_{:\mathcal{P}}^T + \left(A_{:\mathcal{R}} - A_{:\mathcal{P}} D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}}\right) S^{-1} \left(A_{:\mathcal{R}}^T - D_{\mathcal{P}\mathcal{R}}^T D_{\mathcal{P}\mathcal{P}}^{-1} A_{:\mathcal{P}}^T\right) . \tag{1}$$

The first term of the right-hand side is the projection matrix $P^{(\mathcal{P})}$ which projects vectors onto the span of the subset $\mathcal{P}$ of columns. The second term can be simplified as follows. Let E be an $m \times n$ residual matrix which is calculated as: $E = A - P^{(\mathcal{P})} A$. The sub-matrix $E_{:\mathcal{R}}$ can be expressed as

$$E_{:\mathcal{R}} = A_{:\mathcal{R}} - P^{(\mathcal{P})} A_{:\mathcal{R}} = A_{:\mathcal{R}} - A_{:\mathcal{P}} \left(A_{:\mathcal{P}}^T A_{:\mathcal{P}}\right)^{-1} A_{:\mathcal{P}}^T A_{:\mathcal{R}} = A_{:\mathcal{R}} - A_{:\mathcal{P}} D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}} .$$

Since projection matrices are idempotent, then $P^{(\mathcal{P})} P^{(\mathcal{P})} = P^{(\mathcal{P})}$ and

$$E_{:\mathcal{R}}^T E_{:\mathcal{R}} = \left(A_{:\mathcal{R}} - P^{(\mathcal{P})} A_{:\mathcal{R}}\right)^T \left(A_{:\mathcal{R}} - P^{(\mathcal{P})} A_{:\mathcal{R}}\right) = A_{:\mathcal{R}}^T A_{:\mathcal{R}} - A_{:\mathcal{R}}^T P^{(\mathcal{P})} A_{:\mathcal{R}} .$$

Substituting with $P^{(\mathcal{P})} = A_{:\mathcal{P}} \left(A_{:\mathcal{P}}^T A_{:\mathcal{P}}\right)^{-1} A_{:\mathcal{P}}^T$ gives

$$E_{:\mathcal{R}}^T E_{:\mathcal{R}} = A_{:\mathcal{R}}^T A_{:\mathcal{R}} - A_{:\mathcal{R}}^T A_{:\mathcal{P}} \left(A_{:\mathcal{P}}^T A_{:\mathcal{P}}\right)^{-1} A_{:\mathcal{P}}^T A_{:\mathcal{R}} = D_{\mathcal{R}\mathcal{R}} - D_{\mathcal{P}\mathcal{R}}^T D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}} = S .$$

Substituting $\left(A_{:\mathcal{P}} D_{\mathcal{P}\mathcal{P}}^{-1} A_{:\mathcal{P}}^T\right)$, $\left(A_{:\mathcal{R}} - A_{:\mathcal{P}} D_{\mathcal{P}\mathcal{P}}^{-1} D_{\mathcal{P}\mathcal{R}}\right)$ and S with $P^{(\mathcal{P})}$, $E_{:\mathcal{R}}$ and $E_{:\mathcal{R}}^T E_{:\mathcal{R}}$ respectively, Equation (1) can be expressed as

$$P^{(\mathcal{S})} = P^{(\mathcal{P})} + E_{:\mathcal{R}} \left(E_{:\mathcal{R}}^T E_{:\mathcal{R}}\right)^{-1} E_{:\mathcal{R}}^T .$$

The second term is the projection matrix $R^{(\mathcal{R})}$ which projects vectors onto the span of $E_{:\mathcal{R}}$. This proves that $P^{(\mathcal{S})}$ can be written in terms of $P^{(\mathcal{P})}$ and R as $P^{(\mathcal{S})} = P^{(\mathcal{P})} + R^{(\mathcal{R})}$ ■

Given the recursive formula for $P^{(\mathcal{S})}$, the following theorem derives a recursive formula for $\mathbf{F}(\mathcal{S})$.

Theorem 2 *Given a set of columns $\mathcal{S}$. For any $\mathcal{P} \subset \mathcal{S}$, $\mathbf{F}(\mathcal{S}) = \mathbf{F}(\mathcal{P}) - \left\|R^{(\mathcal{R})} F\right\|_F^2$, where $F = B - P^{(\mathcal{P})} B$ and $R^{(\mathcal{R})}$ is a projection matrix which projects the columns of F onto the span of the subset $\mathcal{R} = \mathcal{S} \setminus \mathcal{P}$ of columns of $E = A - P^{(\mathcal{P})} A$*

Proof By definition, $\mathbf{F}(\mathcal{S}) = \left\|B - P^{(\mathcal{S})} B\right\|_F^2$. Using Lemma 1, $P^{(\mathcal{S})} B = P^{(\mathcal{P})} B + R^{(\mathcal{R})} B$. The term $R^{(\mathcal{R})} B$ is equal to $R^{(\mathcal{R})} F$ as $E_{:\mathcal{R}}^T B = E_{:\mathcal{R}}^T F$. To prove that, multiplying $E_{:\mathcal{R}}^T$ by $F = B - P^{(\mathcal{P})} B$ gives $E_{:\mathcal{R}}^T F = E_{:\mathcal{R}}^T B - E_{:\mathcal{R}}^T P^{(\mathcal{P})} B$. Using $E_{:\mathcal{R}} = A_{:\mathcal{R}} - P^{(\mathcal{P})} A_{:\mathcal{R}}$, the expression $E_{:\mathcal{R}}^T P^{(\mathcal{P})}$ can be written as $E_{:\mathcal{R}}^T P^{(\mathcal{P})} = A_{:\mathcal{R}}^T P^{(\mathcal{P})} - A_{:\mathcal{R}}^T P^{(\mathcal{P})} P^{(\mathcal{P})}$. This is equal to 0 as $P^{(\mathcal{P})} P^{(\mathcal{P})} = P^{(\mathcal{P})}$ (an idempotent matrix). Substituting in $\mathbf{F}(\mathcal{S})$ and using $F = B - P^{(\mathcal{P})} B$ gives

$$\mathbf{F}(\mathcal{S}) = \left\|B - P^{(\mathcal{P})} B - R^{(\mathcal{R})} F\right\|_F^2 = \left\|F - R^{(\mathcal{R})} F\right\|_F^2$$

Using the relation between Frobenius norm and trace, $\mathbf{F}(\mathcal{S})$ can be simplified to

$$\mathbf{F}(\mathcal{S}) = \text{tr}\left(\left(F - R^{(\mathcal{R})} F\right)^T \left(F - R^{(\mathcal{R})} F\right)\right) = \text{tr}\left(F^T F - F^T R^{(\mathcal{R})} F\right) = \|F\|_F^2 - \left\|R^{(\mathcal{R})} F\right\|_F^2$$

Using $\mathbf{F}(\mathcal{P}) = \|F\|_F^2$ proves the theorem. ■

Using the recursive formula for $\mathbf{F}(\mathcal{S} \cup \{i\})$ allows the development of a greedy algorithm which at iteration t selects column p such that

$$p = \arg\min_i \mathbf{F}(\mathcal{S} \cup \{i\}) = \arg\max_i \left\|P^{(\{i\})} F\right\|_F^2 .$$

Let $G = E^T E$ and $H = F^T E$, the objective function $\left\|P^{(\{i\})}F\right\|_F^2$ can be simplified to

$$\left\|E_{:i}\left(E_{:i}^T E_{:i}\right)^{-1} E_{:i}^T F\right\|_F^2 = \text{tr}\left(F^T E_{:i}\left(E_{:i}^T E_{:i}\right)^{-1} E_{:i}^T F\right) = \frac{\left\|F^T E_{:i}\right\|^2}{E_{:i}^T E_{:i}} = \frac{\|H_{:i}\|^2}{G_{ii}} .$$

This allows the definition of the following greedy generalized CSS problem.

Problem 2 (Greedy Generalized CSS) *At iteration t, find column p such that*

$$p = \arg\max_i \quad \frac{\|H_{:i}\|^2}{G_{ii}}$$

where $H = F^T E$, $G = E^T E$, $F = B - P^{(\mathcal{S})}B$, $E = A - P^{(\mathcal{S})}A$ and $\mathcal{S}$ is the set of columns selected during the first $t-1$ iterations.

For iteration t, define $\boldsymbol{\delta} = G_{:p}$, $\boldsymbol{\gamma} = H_{:p}$, $\boldsymbol{\omega} = G_{:p}/\sqrt{G_{pp}} = \boldsymbol{\delta}/\sqrt{\boldsymbol{\delta}_p}$ and $\boldsymbol{\upsilon} = H_{:p}/\sqrt{G_{pp}} = \boldsymbol{\gamma}/\sqrt{\boldsymbol{\delta}_p}$. The vectors $\boldsymbol{\delta}^{(t)}$ and $\boldsymbol{\gamma}^{(t)}$ can be calculated in terms of A, B and previous $\boldsymbol{\omega}$'s and $\boldsymbol{\upsilon}$'s as

$$\boldsymbol{\delta}^{(t)} = A^T A_{:p} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_p^{(r)} \boldsymbol{\omega}^{(r)}, \qquad \boldsymbol{\gamma}^{(t)} = B^T A_{:p} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_p^{(r)} \boldsymbol{\upsilon}^{(r)} . \tag{2}$$

The numerator and denominator of the selection criterion at each iteration can be calculated in an efficient manner without explicitly calculating H or G using the following theorem.

Theorem 3 *Let $\boldsymbol{f}_i = \|H_{:i}\|^2$ and $\boldsymbol{g}_i = G_{ii}$ be the numerator and denominator of the greedy criterion function for column i respectively, $\boldsymbol{f} = [\boldsymbol{f}_i]_{i=1..n}$, and $\boldsymbol{g} = [\boldsymbol{g}_i]_{i=1..n}$. Then,*

$$\boldsymbol{f}^{(t)} = \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ \left(A^T B\boldsymbol{\upsilon} - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\upsilon}^{(r)T}\boldsymbol{\upsilon}\right)\boldsymbol{\omega}^{(r)}\right)\right) + \|\boldsymbol{\upsilon}\|^2(\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)},$$

$$\boldsymbol{g}^{(t)} = \left(\boldsymbol{g} - (\boldsymbol{\omega} \circ \boldsymbol{\omega})\right)^{(t-1)},$$

where $\circ$ represents the Hadamard product operator.

In the update formulas of Theorem 3, $A^T B$ can be calculated once and then used in different iterations. This makes the computational complexity of these formulas $O(nr)$ per iteration. The computational complexity of the algorithm is dominated by that of calculating $A^T A_{:p}$ in (2) which is of $O(mn)$ per iteration. The other complex step is that of calculating the initial $\boldsymbol{f}$, which is $O(mnr)$. However, these steps can be implemented in an efficient way if the data matrix is sparse. The total computational complexity of the algorithm is $O(\max(mnl, mnr))$, where l is the number of selected columns. Algorithm 1 in Appendix A shows the complete greedy algorithm.

3 Generalized CSS Problems

We describe a variety of problems that can be formulated as a generalized column subset selection (see Table 1). It should be noted that for some of these problems, the use of greedy algorithms has been explored in the literature. However, identifying the connection between these problems and the problem presented in this paper gives more insight about these problems, and allows the efficient greedy algorithm presented in this paper to be explored in other interesting domains.

Column Subset Selection. The basic column subset selection [2–4, 10, 15] is clearly an instance of the generalized CSS problem. In this instance, the target matrix is the same as the source matrix $B = A$ and the goal is to select a subset of columns from a data matrix that best represent other columns. The greedy algorithm presented in this paper can be directly used for solving the basic CSS problem. A detailed comparison of the greedy CSS algorithm and the state-of-the-art CSS methods can be found at [11]). In our previous work [13, 14], we successfully used the proposed greedy algorithm for unsupervised feature selection which is an instance of the CSS problem. We used the greedy algorithm to solve two instances of the generalized CSS problem: one is based on selecting features that approximate the original matrix $B = A$ and the other is based on selecting features that approximate a random partitioning of the features $B_{:c} = \sum_{j \in \mathcal{P}_c} A_{:j}$. The proposed greedy

Table 1: Different problems as instances of the generalized column subset selection problem.

Method	Source	Target
Generalized CSS	A	B
Column Subset Selection	Data matrix A	Data matrix A
Distributed CSS	Data matrix A	Random subspace $A\Omega$
SVD-based CSS	Data matrix A	SVD-based subspace $U_k\Sigma_k$
Sparse Approximation	Atoms D	Target vector $\mathbf{y}$
Simultaneous Sparse Approximation	Atoms D	Target vectors $\left[\mathbf{y_{(1)}}, \mathbf{y_{(2)}}, ...\mathbf{y_{(r)}}\right]$

algorithms achieved superior clustering performance in comparison to state-of-the-art methods for unsupervised feature selection.

Distributed Column Subset Selection. The generalized CSS problem can be used to define distributed variants of the basic column subset selection problem. In this case, the matrix B is defined to encode a concise representation of the span of the original matrix A. This concise representation can be obtained using an efficient method like random projection. In our recent work [12], we defined a distributed CSS based on this idea and used the proposed greedy algorithm to select columns from big data matrices that are massively distributed across different machines.

SVD-based Column Subset Selection. Çivril and Magdon-Ismail [5] proposed a CSS method which first calculates the Singular Value Decomposition (SVD) of the data matrix, and then selects the subset of columns which best approximates the leading singular values of the data matrix. The formulation of this CSS method is an instance of the generalized CSS problem, in which the target matrix is calculated from the leading singular vectors of the data matrix. The greedy algorithm presented in [5] can be implemented using Algorithm 1 by setting $B = U_k\Sigma_k$ where U_k is a matrix whose columns represent the leading left singular vectors of the data matrix, and Σ_k is a matrix whose diagonal elements represent the corresponding singular values. Our greedy algorithm is however more efficient than the greedy algorithm of [5].

Sparse Approximation. Given a target vector and a set of basis vectors, also called atoms, the goal of sparse approximation is to represent the target vector as a linear combination of a few atoms [20]. Different instances of this problem have been studied in the literature under different names, such as variable selection for linear regression [8], sparse coding [16, 19], and dictionary selection [6, 9]. If the goal is to minimize the discrepancy between the target vector and its projection onto the subspace of selected atoms, the sparse approximation can be considered an instance of the generalized CSS problem in which the target matrix is a vector and the columns of the source matrix are the atoms. Several greedy algorithms have been proposed for sparse approximation, such as basic matching pursuit [18], orthogonal matching pursuit [21], the orthogonal least squares [7]. The greedy algorithm for generalized CSS is equivalent to the orthogonal least squares algorithm (as defined in [1]) because at each iteration it selects a new column such that the reconstruction error after adding this column is minimum. Algorithm 1 can be used to efficiently implement the orthogonal least squares algorithm by setting $B = \mathbf{y}$, where $\mathbf{y}$ is the target vector. However, an additional step will be needed to calculate the weights of the selected atoms as $\left(A_{:\mathcal{S}}^T A_{:\mathcal{S}}\right)^{-1} A_{:\mathcal{S}}^T \mathbf{y}$.

Simultaneous Sparse Approximation. A more general sparse approximation problem is the selection of atoms which represent a group of target vectors. This problem is referred to as simultaneous sparse approximation [22]. Different greedy algorithms have been proposed for simultaneous sparse approximation with different constraints [6, 22]. If the goal is to select a subset of atoms to represent different target vectors without imposing sparsity constraints on each representation, simultaneous sparse approximation will be an instance of the greedy CSS problem, where the source columns are the atoms and the target columns are the input signals.

4 Conclusions

We define a generalized variant of the column subset selection problem and present a fast greedy algorithm for solving it. The proposed greedy algorithm can be effectively used to solve a variety of problems that are instances of the generalized column subset selection problem.

References

[1] T. Blumensath and M. E. Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. 2007. Unpublished Manuscript.

[2] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near optimal column-based matrix reconstruction. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'11)*, pages 305 –314, 2011.

[3] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*, pages 968–977, 2009.

[4] C. Boutsidis, J. Sun, and N. Anerousis. Clustered subset selection and its applications on it service metrics. In *Proceedings of the Seventeenth ACM Conference on Information and Knowledge Management (CIKM'08)*, pages 599–608, 2008.

[5] A. Çivril and M. Magdon-Ismail. Column subset selection via sparse approximation of SVD. *Theoretical Computer Science*, 421(0):1 – 14, 2012.

[6] V. Cevher and A. Krause. Greedy dictionary selection for sparse representation. *Journal of Selected Topics in Signal Processing*, 5(5):979–988, 2011.

[7] S. Chen, S. A. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[8] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 45–54, 2008.

[9] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning, (ICML'11)*, pages 1057–1064, 2011.

[10] P. Drineas, M. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 316–326. Springer, 2006.

[11] A. K. Farahat. *Greedy Representative Selection for Unsupervised Data Analysis*. PhD thesis, University of Waterloo, 2012.

[12] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel. Distributed column subset selection on MapReduce. In *Proceedings of the Thirteenth IEEE International Conference on Data Mining (ICDM'13)*, 2013. In Press.

[13] A. K. Farahat, A. Ghodsi, and M. S. Kamel. An efficient greedy method for unsupervised feature selection. In *Proceedings of the Eleventh IEEE International Conference on Data Mining (ICDM'11)*, pages 161 –170, 2011.

[14] A. K. Farahat, A. Ghodsi, and M. S. Kamel. Efficient greedy feature selection for unsupervised learning. *Knowledge and Information Systems*, 35(2):285–310, 2013.

[15] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 370 –378, 1998.

[16] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19 (NIPS'06)*, pages 801–808. MIT, 2006.

[17] H. Lütkepohl. *Handbook of Matrices*. John Wiley & Sons Inc, 1996.

[18] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[19] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by VI? *Vision Research*, 37(23):3311–3326, 1997.

[20] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.

[21] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[22] J. Tropp, A. Gilbert, and M. Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

Appendix A

Algorithm 1 Greedy Generalized Column Subset Selection

Input: Source matrix A, Target matrix B, Number of columns l
Output: Selected subset of columns $\mathcal{S}$

1: Initialize $\boldsymbol{f}_i^{(0)} = \|B^T A_{:i}\|^2$, $\boldsymbol{g}_i^{(0)} = A_{:i}^T A_{:i}$ for $i = 1 \dots n$
2: Repeat $t = 1 \to l$:
3: $\quad p = \arg\max_i \boldsymbol{f}_i^{(t)} / \boldsymbol{g}_i^{(t)}, \;\; \mathcal{S} = \mathcal{S} \cup \{p\}$
4: $\quad \boldsymbol{\delta}^{(t)} = A^T A_{:p} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_p^{(r)} \boldsymbol{\omega}^{(r)}$
5: $\quad \boldsymbol{\gamma}^{(t)} = B^T A_{:p} - \sum_{r=1}^{t-1} \boldsymbol{\omega}_p^{(r)} \boldsymbol{\upsilon}^{(r)}$
6: $\quad \boldsymbol{\omega}^{(t)} = \boldsymbol{\delta}^{(t)} / \sqrt{\boldsymbol{\delta}_p^{(t)}}$, $\boldsymbol{\upsilon}^{(t)} = \boldsymbol{\gamma}^{(t)} / \sqrt{\boldsymbol{\delta}_p^{(t)}}$
7: $\quad$ Update $\boldsymbol{f}_i$'s, $\boldsymbol{g}_i$'s (Theorem 3)

Proof of Theorem 3

Let $\mathcal{S}$ denote the set of columns selected during the first $t-1$ iterations, $F^{(t-1)}$ denote the residual matrix of B at the start of the t-th iteration (i.e., $F^{(t-1)} = B - P^{(\mathcal{S})} B$), and p be the column selected at iteration t. From Lemma 1, $P^{(\mathcal{S} \cup \{p\})} = P^{(\mathcal{S})} + R^{(\{p\})}$. Multiplying both sides with B gives $P^{(\mathcal{S} \cup \{p\})} B = P^{(\mathcal{S})} B + R^{(\{p\})} B$. Subtracting both sides from B and substituting $B - P^{(\mathcal{S})} B$, and $B - P^{(\mathcal{S} \cup \{p\})} B$ with $F^{(t-1)}$ and $F^{(t)}$ respectively gives $F^{(t)} = \left(F - R^{(\{p\})} B\right)^{(t-1)}$.

Since $R^{(\{p\})} B = R^{(\{p\})} F$ (see the proof of Theorem 2), $F^{(t)}$ can be calculated recursively as

$$F^{(t)} = \left(F - R^{(\{p\})} F\right)^{(t-1)}.$$

Similarly, $E^{(t)}$ can be expressed as

$$E^{(t)} = \left(E - R^{(\{p\})} E\right)^{(t-1)}.$$

Substituting with F and E in $H = F^T E$ gives

$$H^{(t)} = \left(\left(F - R^{(\{p\})} F\right)^T \left(E - R^{(\{p\})} E\right)\right)^{(t-1)} = \left(H - F^T R^{(\{p\})} E\right)^{(t-1)}.$$

Using $R^{(\{p\})} = E_{:p} \left(E_{:p}^T E_{:p}\right)^{-1} E_{:p}^T$, and given that $\boldsymbol{\omega} = G_{:p} = E^T E_{:p} / \sqrt{E_{:p}^T E_{:p}}$ and $\boldsymbol{\upsilon} = H_{:p} = F^T E_{:p} / \sqrt{E_{:p}^T E_{:p}}$, the matrix H can be calculated recursively as

$$H^{(t)} = \left(H - \boldsymbol{\upsilon} \boldsymbol{\omega}^T\right)^{(t-1)}.$$

Similarly, G can be expressed as

$$G^{(t)} = \left(G - \boldsymbol{\omega} \boldsymbol{\omega}^T\right)^{(t-1)}.$$

Using these recursive formulas, $\boldsymbol{f}_i^{(t)}$ can be calculated as

$$\begin{aligned}
\boldsymbol{f}_i^{(t)} &= \left(\|H_{:i}\|^2\right)^{(t)} = \left(\|H_{:i} - \boldsymbol{\omega}_i \boldsymbol{\upsilon}\|^2\right)^{(t-1)} \\
&= \left((H_{:i} - \boldsymbol{\omega}_i \boldsymbol{\upsilon})^T (H_{:i} - \boldsymbol{\omega}_i \boldsymbol{\upsilon})\right)^{(t-1)} \\
&= \left(H_{:i}^T H_{:i} - 2\boldsymbol{\omega}_i H_{:i}^T \boldsymbol{\upsilon} + \boldsymbol{\omega}_i^2 \|\boldsymbol{\upsilon}\|^2\right)^{(t-1)} \\
&= \left(\boldsymbol{f}_i - 2\boldsymbol{\omega}_i H_{:i}^T \boldsymbol{\upsilon} + \boldsymbol{\omega}_i^2 \|\boldsymbol{\upsilon}\|^2\right)^{(t-1)}.
\end{aligned}$$

Similarly, $\boldsymbol{g}_i^{(t)}$ can be calculated as

$$\boldsymbol{g}_i^{(t)} = G_{ii}^{(t)} = \left(G_{ii} - \boldsymbol{\omega}_i^2\right)^{(t-1)} = \left(\boldsymbol{g}_i - \boldsymbol{\omega}_i^2\right)^{(t-1)}.$$

Let $\boldsymbol{f} = [\boldsymbol{f}_i]_{i=1..n}$and $\boldsymbol{g} = [\boldsymbol{g}_i]_{i=1..n}$, $\boldsymbol{f}^{(t)}$ and $\boldsymbol{g}^{(t)}$ can be expressed as

$$\begin{aligned} \boldsymbol{f}^{(t)} &= \left(\boldsymbol{f} - 2\left(\boldsymbol{\omega} \circ H^T\boldsymbol{v}\right) + \|\boldsymbol{v}\|^2\left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}, \\ \boldsymbol{g}^{(t)} &= \left(\boldsymbol{g} - \left(\boldsymbol{\omega} \circ \boldsymbol{\omega}\right)\right)^{(t-1)}, \end{aligned} \tag{3}$$

where $\circ$ represents the Hadamard product operator.

Using the recursive formula of H, the term $H^T\boldsymbol{v}$ at iteration $(t-1)$ can be expressed as

$$H^T\boldsymbol{v} = \left(A^TB - \Sigma_{r=1}^{t-2}\left(\boldsymbol{\omega}\boldsymbol{v}^T\right)^{(r)}\right)\boldsymbol{v} = A^TB\boldsymbol{v} - \Sigma_{r=1}^{t-2}\left(\boldsymbol{v}^{(r)T}\boldsymbol{v}\right)\boldsymbol{\omega}^{(r)}$$

Substituting with $H^T\boldsymbol{v}$ in (3) gives the update formulas for $\boldsymbol{f}$ and $\boldsymbol{g}$. ■