Weighted A* Algorithms for Unsupervised Feature Selection with Provable Bounds on Suboptimality

Hiromasa Arai, Ke Xu, Crystal Maung, Haim Schweitzer
{Hiromasa.Arai, ke.xu}@utdallas.edu Crystal.Maung@gmail.com, hschweitzer@utdallas.edu
Department of Computer Science, University of Texas (Dallas)
800 W Campbell Road, Richardson, TX 75080

Abstract

Identifying a small number of features that can represent the data is believed to be NP-hard. Previous approaches exploit algebraic structure and use randomization. We propose an algorithm based on ideas similar to the Weighted A* algorithm in heuristic search. Our experiments show this new algorithm to be more accurate than the current state of the art.

1 The problem

Feature selection is a standard dimensionality reduction technique. Given data items described in terms of n features, the goal is to select $k<n$ features such that the reduced k dimensional vectors are useful for the task at hand. A standard criterion is to select k features that can be used to approximate all n features by a linear combination of the selected features (e.g., (Golub and Van-Loan 1996; Dasgupta et al. 2007)).

Let X be an $m\times n$ data matrix of m items (rows), each defined in terms of n features (columns). We wish to estimate X by a linear combination of k of its columns: $X \approx S_k A$. The matrix $S_k = (x_{s_1}, \ldots, x_{s_k})$ is formed by the selected columns of X, and A is the coefficients matrix. In Frobenius norm the approximation error is:

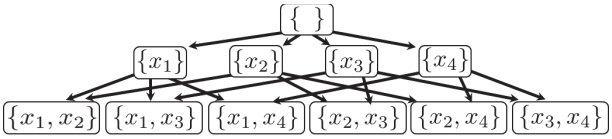
$$E = \min_{A,S_k} \|X - S_k A\|_F^2 \tag{1}$$

Finding S_k that minimizes (1) is also known as the "*Column Subset Selection Problem*" (CSSP) in numerical linear algebra (e.g., (Golub and Van-Loan 1996; Boutsidis, et. al. 2009)). It is believed to be NP-hard (Çivril 2014).

2 Heuristic search for the optimal solution

Let S_k^* denote a (best) selection of k columns that minimizes the error in (1), and let $E(S_k^*)$ be the corresponding (minimal) error. The eigenvalues of the matrix XX^T can be used to obtain bounds on $E(S_k^*)$. Let E_k^* denote the error of the best approximation of X in terms of a rank-k matrix. Then the following inequalities (Golub and Van-Loan 1996; Deshpande and Rademacher 2010) hold:

$$E_k^* \le E(S_k^*) \le (k+1)E_k^* \tag{2}$$



0. Put the root node into the fringe F.
1. While F is nonempty and no solution found:
  1.1 Pick n_i with the smallest f_i' from F.
  1.2 If n_i has k columns return it as the solution.
  1.3 Otherwise:
    1.3.1 Add n_i to the closed list C.
    1.3.2 Examine all children n_j of n_i.
    1.3.2.1 If n_j is in C or F do nothing.
    1.3.2.2 Otherwise put n_j in F.

Figure 1: Example of the subsets graph and the details of the generic heuristic search algorithm.

Calculating E_k^* is easy, since the columns of the best rank-k matrix are the k dominant eigenvectors of XX^T, scaled by the eigenvalues. From this it follows that: $E_k^* = \sum_{t=k+1}^{m} \lambda_t$, where λ_t is the t-largest eigenvalue.

A recent study (Arai et al. 2015) has shown how to model the CSSP as a graph search, and then apply the classic A*. Their algorithm, that we call **CSSP-A***, uses (2) to compute heuristic estimates. It is guaranteed to find the optimal solution, runs much faster than exhaustive search, but much slower than other algorithms. It operates on a graph, where nodes correspond to column subsets. There is an edge from subset s_1 to subset s_2 if adding one column to s_1 creates s_2. An example is shown in Figure 1.

3 Weighted A* algorithms for the CSSP

Let n_i be a node in the graph with the corresponding subset S_i. Set $j = |S_i|$ and define:

$g_i \triangleq$ Error of estimating X using the j columns in S_i.

$f_i \triangleq$ Error of estimating X using the j columns in S_i and additional "best possible" $k-j$ vectors.

$h_i \triangleq g_i - f_i$.

Let g_* be the smallest possible error in Equation (1). It was shown by (Arai et al. 2015) that running the algorithm in Fig.1 with $f_i' = g_i - h_i$ is guaranteed to find an optimal solution with error g_*. Since this is similar to the classic A*, one

Name	f'_i	Suboptimality Bound
CSSP-WA*-g	$g_i - h_i + \epsilon g_i$	$g'_* \leq g_* + \epsilon g_0$
CSSP-WA*-h	$g_i - h_i + \epsilon h_i$	$g'_* \leq g_* + \epsilon h_0$
CSSP-WA*-b	$g_i - h_i + \epsilon b_i$	$g'_* \leq (1 + \epsilon(k+1))g_*$

Figure 2: Three CSSP-WA* algorithms

may expect an improved run time by the classic Weighted A* algorithm , which replaces h_i with $(1+\epsilon)h_i$ (Pearl 1984). Unfortunately we have found experimentally that the resulting algorithm runs even slower than the CSSP-A*. However, we discovered that assigning weights differently produces the desired improved runtime, with precise bounds on suboptimality. Specifically, we propose to use the following heuristic for a Weighted A* algorithm for the CSSP:

$$f'_i = g_i - h_i + \epsilon v_i$$

with $\epsilon \geq 0$, $v_i \geq 0$. We call such an algorithm a **CSSP-WA***.

3.1 Suboptimality

Let n_* be an optimal solution node, and let g_* be the corresponding (optimal) error value. Suppose $v_i \geq 0$ can be computed at each node, and $f'_i{=}g_i{-}h_i{+}\epsilon v_i$.

Theorem 1: If f'_i is used by the algorithm in Figure 1, it will terminate with a solution satisfying: $g'_* \leq g_* + \epsilon v_{\max}$, where $v_{\max} = \max_i v_i$. See (Arai et al. 2016) for the proof.

The theorem is useful for proving bounds on suboptimality, but by itself it does not provide guidance as to what constitutes a useful v_i. The following corollary suggests that v_i should be chosen monotonically decreasing along any path. Under this condition the bound becomes tighter during the run of the algorithm.

Corollary 1: Let g_*, g'_*, v_i be as above. Suppose v_i is monotonically decreasing along any path. Let n_r be a visited node in the path to the solution, and let v_r be the value computed at n_r. Then: $g'_* \leq g_* + \epsilon v_r$. (Proof: Apply Theorem-1 to the subgraph rooted at n_r.)

We note that the guarantee in the corollary may be much stronger than the guarantee provided by the theorem since v_r may be much smaller than v_0, the value of v_i at the root.

Three choices for v_i are shown in Figure 2: g_i, h_i, and b_i. The function b_i is computed in a similar way to the upper bound in (2) and gives multiplicative bounds on suboptimality (see (Arai et al. 2016) for details). It can be shown that the functions g_i, h_i produce the same algorithm with different range values of ϵ.

3.2 Computing the heuristics

In calculating f_i, g_i, h_i we follow (Arai et al. 2015). Let S_i be the columns at node n_i, and set $j = |S_i|$. Compute the matrix Q_i, an orthogonal basis to S_i. Compute: $X_i = X - Q_i Q_i^T X$, and $B_i = X_i X_i^T$. Let $\lambda_1 \geq \ldots \geq \lambda_m$ be the eigenvalues of B_i. Then: $g_i = \sum_{t=1}^{m} \lambda_t = \|X_i\|_F^2$, $h_i = \sum_{t=1}^{j} \lambda_t, \quad f_i = \sum_{t=j+1}^{m} \lambda_t$.

4 Experimental results

We tested and compared our algorithms to the state of the art and the CSSP-A* on several datasets. We describe here a comparison on the TechTC01 dataset of size (163$\times$29,261). The results of the CSSP-WA* variants were compared to the following two algorithms: 1. The **GKS**, considered one of the best among the pure algebraic algorithms (Golub and Van-Loan 1996). 2. The randomized two-stage-method that we call **LEV_GE** (Boutsidis, et. al. 2009). It is considered one of the top randomized algorithms. The results are shown in figures 3,4. Observe that the CSSP-WA*-b always come as best in Figure 3. In Figure 4, observe that the CSSP-WA* found the optimal solution in each case. (The CSSP-WA*-g and the CSSP-WA*-h also find the optimal solution in this case.) Observed that the CSSP-WA* variants were much faster than the CSSP-A*.

k	GKS/(time)	LEV_GE/(time)	WA*-b/(time)
20	7.1E+05/(0.02)	8.7E+05/(0.67)	**6.9E+05**/(0.80)
60	1.5E+05/(0.04)	1.7E+05/(1.93)	**1.3E+05**/(5.68)
100	2.8E+04/(0.06)	2.8E+04/(3.15)	**2.1E+04**/(14.47)

Figure 3: Error and time(min) comparison(WA* uses ϵ=0.5.)

k	A*	WA*-b
3	{6, 84, 121}	{6, 84, 121}
4	{6, 84, 121, 329}	{6, 84, 121, 329}
5	{6, 84, 121, 329, 5376}	{6, 84, 121, 329, 5376}

Figure 4: Column indices selected by A* and WA*-b(ϵ=0.5)

5 Concluding remarks

The unsupervised feature selection problem that we consider, the CSSP, is a practical problem that was heavily studied for over 50 years. Our solution, using ideas from heuristic search, is more accurate than all currently available practical approaches. This comes at the expense of a longer running time. Still, our approach should be a preferred choice when an increase in running time can be tolerated.

References

Arai, H.; Maung, C.; and Schweitzer, H. 2015. Optimal column subset selection by A-Star search. In *AAAI'15*, 1079–1085.

Arai, H.; Maung, C.; Xu K.; and Schweitzer, H. 2016. Unsupervised feature selection by heuristic search with provable bounds on suboptimality. In *AAAI'16*.

Boutsidis, C.; Mahoney, M. W.; and Drineas, P. 2009. An improved approximation algorithm for the column subset selection problem. In *SODA'09*, 968–977.

Çivril, A. 2014. Column subset selection problem is ughard. *Journal of Computer and System Sciences* 80(4):849–859.

Dasgupta et al. 2007. Feature selection methods for text classification. In *KDD'07*, 230–239.

Deshpande, A., and Rademacher, L. 2010. Efficient volume sampling for row/column subset selection. In *FOCS'10*, 329–338.

Golub, G. H., and Van-Loan, C. F. 1996. *Matrix computations*.

Pearl, J. 1984. *Heuristics*. Addison-Wesley.