Supplementary Material for Fast Algorithms for Learning with Long N-grams via Suffix Tree Based Matrix Multiplication

Hristo S. Paskov

1 Modified Sparse Matrix Format

The standard compressed sparse column (CSC) format for a sparse $M \times N$ matrix X consisting of nz non-zero entries stores three arrays:

1. The jc array, an array of size $N+1$ such that jc$[i+1]-$ jc$[i]$ gives the number of non-zero entries in column i.
2. The ir array, an array of size nz in which indices jc$[i], \ldots,$ jc$[i+1]-1$ contain the row ids of the non-zero entries in column i.
3. The x array, a double array of size nz containing the non-zero entries of X in the same order that they are listed in the ir array.

This matrix format is inefficient when storing frequency data since we know all entries in x are non-negative integers. Moreover, the number of bits needed to store each index in the jc array is $\lceil \log_2 nz \rceil$ which can be significantly larger than $\lceil \log_2 U^X \rceil$ where U^X is the largest number of non-zero elements in any column. Our modified CSC format simply replaces the jc array with an integer array of size N that stores the number of non-zero elements in each column and it replaces x by an integer array of frequency counts. This modifications can lead to substantial savings when appropriate.

2 Examples for N-Gram and Node Matrix Inefficiencies

We start with a canonical example from the suffix tree literature which highlights the inefficiency of the N-gram matrix. Suppose that the document corpus consists of a single document $D_1 = c_1 c_2 \ldots c_n$ of n distinct characters, i.e. $c_i \neq c_j$ if $i \neq j$. There are $\frac{n^2+n}{2}$ distinct substrings in this document, so the N-gram matrix pertaining to all possible N-grams is a row vector of $\frac{n^2+n}{2}$ ones. In contrast, the node matrix $\mathcal{X}$ only consists of n entries pertaining to every distinct character. Direct multiplication with X requires $\Theta(n^2)$ operations whereas multiplication with $\mathcal{X}$ requires $\Theta(n)$ operations.

Next, to show that the node matrix can be inefficient, consider a document corpus comprised of K documents and an alphabet of K distinct characters $c_1, \ldots, c_K$. The i^{th} document $D_i = c_1 c_2 \ldots c_i$ is comprised of the first i characters of the alphabet and the total corpus length is $n = \frac{K^2+K}{2}$. By inspecting the structure of the suffix tree $\mathcal{T}_\mathcal{C}$ for this corpus, it is possible to show that both the all N-grams matrix X and all N-grams node matrix $\mathcal{X}$ have $\Theta(K^3)$ non-zero entries and thus require $\Theta(n\sqrt{n})$ memory to store and $\Theta(n\sqrt{n})$ operations to multiply.

In particular, consider the branch β_1 corresponding to suffix $D_K[1]$, i.e. the suffix consisting of K characters and equal to the entire document D_K. Note that there is a document D_i equalling every *prefix* $[i]D_K = c_1 c_2 \ldots c_i$ of D_K. By construction, for $i = 1, \ldots, K-1$, every occurrence of the substring $[i]D_K$ in $\mathcal{C}$ is either followed by c_{i+1} (for example in document D_{i+1}) or is the end of a document (i.e. D_i). This structure implies that β_1 contains $K-1$ internal nodes pertaining to the first $K-1$ characters in $D_K[1]$ and that the edge labels connecting these nodes contain a single character. For $i < K$ the internal node pertaining to character c_i has two children: a leaf indicating the end of document D_i and another internal node corresponding to character c_{i+1}. The final node in β_1 has character label c_K and is a leaf signalling the end of D_K. If we count this node (for simplicity), the node pertaining to character i appears in exactly $K-i+1$ documents, so the column for substring $[i]D_K$ in the (all) node matrix $\mathcal{X}$ contains $K-i+1$ non-zero entries. The K prefixes of D_K each pertain to a node in β_1 and have a column in $\mathcal{X}$ with a total of

$$\sum_{i=1}^{K}(K-i+1) = \frac{K^2+K}{2}$$

non-zero entries.

The other strings in the corpus are formed in a similar manner by looking at the prefixes of $c_i \ldots c_K$, i.e. all prefixes of every suffix of D_K. Note that the corpus length is $n = \frac{K^2+K}{2}$ and there are n distinct substrings, equivalence classes, and nodes in $\mathcal{T}_{\mathcal{C}}$ (that correspond to these equivalence classes) so $\mathcal{X}$ has n columns. By iterating our earlier reasoning we see that branch β_k corresponds to (all prefixes of) suffix $D_K[k]$ and it accounts for k of these nodes. In total these k nodes contribute

$$\sum_{i=1}^{k}(k-i+1) = \frac{k^2+k}{2} \qquad (1)$$

non-zero entries to $\mathcal{X}$.

By summing equation (1) from $k = 1, \dots, K$ we find that $\mathcal{X}$ has $\Theta(K^3)$, i.e. $\Theta(n\sqrt{n})$, non-zero entries and therefore is as inefficient as the naïve all N-grams matrix!

3 Proof of Theorem 4

Suppose that f is $\mathcal{J}$-PI where $\mathcal{J} = \{\zeta_1, \dots, \zeta_m\}$ and let X^* be the set of minimizers of $\min_{x\in\mathbb{R}^d} f(x)$. If X^* is empty then our proof is trivial, so we assume that X^* is not empty. The central idea behind our proof is that X^* must contain a Cartesian product of permutahedrons (Ziegler, 1995). In particular, given a finite vector $a \in \mathbb{R}^n$, the permutahedron $\mathbb{P}(a) \subset \mathbb{R}^n$ on a is the polyhedron formed by taking the convex hull of all $n!$ n-vectors whose entries are some permutation of the entries of a.

In order to see how this relates to f, let $x \in X^*$ be optimal and let x_{ζ_k} denote the $n_k = |\zeta_k|$ entries in x with indices in ζ_k. Since f is $\mathcal{J}$-PI, it follows that f's value remains unchanged if we permute the x_{ζ_k} arbitrarily. In fact, by definition, if $\hat{x}$ is the vector formed by arbitrarily permuting the entries within each $\zeta_k \in \mathcal{J}$, then $f(x) = f(\hat{x})$ so $\hat{x} \in X^*$ is optimal as well. Assume, without loss of generality, that $\zeta_1 = \{1, \dots, n_1\}, \zeta_2 = \{n_1+1, \dots, n_1+n_2\}$ and so on and define

$$\mathcal{Q} = \mathbb{P}(x_{\zeta_1}) \times \mathbb{P}(x_{\zeta_2}) \times \cdots \times \mathbb{P}(x_{\zeta_m}).$$

Our reasoning shows that any $z \in \mathcal{Q}$ is optimal and hence $\mathcal{Q} \subset X^*$.

Now consider the centroid of $\mathcal{Q}$, $\mu \in \mathbb{R}^d$. The centroid of $\mathbb{P}(a)$ for $a \in \mathbb{R}^n$ is simply the n-vector with $\frac{1}{n}\sum_{i=1}^n a_i$ in every entry (Ziegler, 1995). Moreover, since $\mathcal{Q}$ is a Cartesian product of polyhedra, its centroid is given by stacking the centroids of its constituent polyhedra. Let $\eta \in \mathbb{R}^m$ have its entries be $\eta_k = \frac{x_{\zeta_k}^T \mathbf{1}}{n_k}$, i.e. the mean of the elements in x_{ζ_k} and define $V \in \{0,1\}^{d\times m}$ to be the binary matrix in which column k has ones in indices ζ_k and is all 0 otherwise. It follows that $\mu = V\eta$, and since $\mu \in \mathcal{Q} \subset X^*$, there must be a minimizer of f whose entries are identical in each of the ζ_k.

This reasoning then shows that constrained problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad x \in \operatorname{col} V. \qquad (2)$$

is a constrained convex problem (with a linear constraint) and therefore has a minimum that is *lower bounded* by the minimum of our original (unconstrained) problem. By construction of μ, we see that it satisfies the linear constraint and is an optimal point for both problems. It follows, then, that the minimizers of the problem in equation (2) are a subset of X^*. Moreover, solving equation (2) will always provide a minimizer of the original optimization problem.

We can then replace the subspace constraint by noting that $x \in \operatorname{col} V$ if and only if $x = Vz$ for some $z \in \mathbb{R}^d$. This leads to a problem which is equivalent to the problem in (2), namely

$$\underset{z\in\mathbb{R}^m}{\text{minimize.}} \quad f(Vz) \qquad (3)$$

It follows that we obtain a minimizer of our original problem simply by setting $x = Vz$, i.e. $x_i = z_k$ where $i \in \zeta_k$. Importantly, equation (3) is a smaller minimization problem over m variables and not d terms. We note that this proof is entirely geometric and the details of how problem (3) might further be reduced algebraically are problem dependent. QED.

References

Ziegler, Günter M. Lectures on polytopes. Vol. 152. Springer Science & Business Media, 1995.