EFFICIENT CALCULATION OF POLYNOMIAL FEATURES ON SPARSE MATRICES

Andrew Nystrom
Savvysherpa Inc.
6200 Shingle Creek Pkwy
Suite 400
Minneapolis, MN 55430, USA
awnystrom@gmail.com

John Hughes
Department of Computer Science
Brown University
Providence, RI
jfh@cs.brown.edu

ABSTRACT

We provide an algorithm for polynomial feature expansion that can operate directly on sparse a matrix. For a vector $\vec{x}$ of a $N \times D$ matrix, the algorithm has time and space complexity $O(d^k D^k)$ where k is the polynomial order and d is the density of $\vec{x}$.

1 INTRODUCTION

Polynomial feature expansion is a tool long used in statistics for approximating nonlinear functions (see Gergonne (1974); Smith (1918)). While their use is widespread, the authors of this work are unaware of any improvements made to their calculation efficiency. In this work we provide an algorithm for calculating polynomial features for a row vector $\vec{x}$ of a $N \times D$ matrix with time and space complexity $O(d^k D^k)$ where k is the polynomial order and d is the density of $\vec{x}$. The density of a vector is the percent of elements that are nonzero, so $0 \leq d \leq 1$. The standard algorithm has time and space complexity $O(D^k N)$, so the added factor of d^k represents a significant reduction in time. The algorithm does not require the densification of the matrix, e.i. the matrix remains in sparse form, so the space complexity is also $O(d^k D^k)$ as opposed to $O(D^k)$.

2 ALGORITHM

The traditional method of calculating polynomial features for a vector $\vec{x}$ involves augmenting a feature for the product of each combination with repetition (without repetition for interaction features) of features in $\vec{x}$ of orders 2 to k. This method does not exploit the sparsity of a sparse matrix and will yield a product of zero any time one of the features involved in the product is zero. In a sparse matrix, products resulting in zero will be common. Since the default value of a sparse matrix is zero, these products are entirely unnecessary to compute.

The main idea behind our algorithm is to only compute products that do not involve zeros. In a compressed sparse row matrix, the columns containing nonzero data are the only columns that are stored. We can therefore iterate over products of combinations with repetition of order k of these columns for each row to calculate k-degree polynomial features.

While the idea is straightforward, there is yet one unaddressed caveat. Given a set of columns whose corresponding nonzero components were multiplied to produce a polynomial feature, where in the augmented polynomial vector does the result of the product belong? To adress this, we give a bijective mapping from the set of possible column index combinations with repetition of order k onto the column index space of the polynomial feature matrix. More precisely, this the mapping is of the following form:

$$(x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}) \rightarrowtail\!\!\!\!\twoheadrightarrow p_{i_0 i_1 \ldots i_{k-1}} \in \{0, 1, \ldots, \left(\!\!\binom{D}{k}\!\!\right)\} \forall (i_0, i_1, \ldots, i_{k-1}) \tag{1}$$

such that $0 \leq i_0 \leq i_1 \leq \cdots \leq i_{k-1} < D$ where $(x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}})$ are column indicies of a row vector $\vec{x}$ of an $N \times D$ input matrix and $p_{i_0 i_1 \ldots i_{k-1}}$ is a column index into the polynomial expansion vector for $\vec{x}$ and $\left(\binom{D}{k}\right)$ is the number of combinations with repetition of size k drawn from D objects. In general, and is equal to $\left(\binom{n}{k}\right) = \binom{n+k-1}{n-1} = \binom{n+k-1}{k}$, as given by Stanley (1986).

2.1 Construction of Mappings

We seek a map from matrix indices (i, j) (with $i < j$ and $0 \leq i < D$) to numbers $f(i, j)$ with $0 \leq f(i, j) < \frac{D(D-1)}{2}$, one that follows the pattern indicated by

$$\begin{bmatrix} x & 0 & 1 & 3 \\ x & x & 2 & 4 \\ x & x & x & 5 \\ x & x & x & x \end{bmatrix} \tag{2}$$

where the entry in row i, column j, displays the value $f(i, j)$.

To simplify slightly, we introduce a notation for the nth triangular number,

$$T_2(n) = \frac{n(n+1)}{2} \tag{3}$$

The subscript 2 indicates that these are triangles in two dimensions; we'll use $T_3(n)$ to indicate the nth tetrahedral number, and so on for higher dimensions.

Observe that in Equation 2, each entry in column j (for $j > 0$) lies in the range

$$T_2(j-1) \leq e < T_2(j). \tag{4}$$

And in fact, the entry in the ith row of that column is just $i + T_2(j-1)$. Thus we have

$$\begin{aligned} f(i, j) &= i + T_2(j-1) && (5)\\ &= i + \frac{(j-1)j}{2} && (6)\\ &= \frac{2i + j^2 - j}{2}. && (7) \end{aligned}$$

For instance, in column $j = 2$ in our example (the *third* column), the entries range from 1 to 2, while $T_2(j-1) = T_2(1) = 1$ and $T_2(j) = T_2(2) = 3$, and the entry in column $j = 2$, row $i = 1$ is $i + T_2(j-1) = 1 + 1 = 2$.

2.1.1 Other indices

With one-based indexing in both the domain and codomain, the formula above becomes

$$\begin{aligned} f_1(i, j) &= 1 + f(i-1, j-1) && (8)\\ &= 1 + f(i-1, j-1) && (9)\\ &= \frac{2 + 2(i-1) + (j-1)^2 - (j-1)}{2} && (10)\\ &= \frac{2i + j^2 - 3j + 2}{2} && (11) \end{aligned}$$

2.1.2 Polynomial Features

For polynommial features, we seek a map from matrix indices (i, j) (with $i \leq j$ and $0 \leq i < D$) to numbers $g(i, j)$ with $0 \leq f(i, j) < \frac{D(D+1)}{2}$, one that follows the pattern indicated by

$$\begin{bmatrix} 0 & 1 & 3 & 6 \\ x & 2 & 4 & 7 \\ x & x & 5 & 8 \\ x & x & x & 9 \end{bmatrix} \tag{12}$$

i.e., essentially the same task as before, except that the diagonal is included. One can regard all but the last column of entries in Equation 12 as corresponding to the entries in Equation 2, but shifted to the left. Thus the formula for $g(i,j)$ is simply the formula for f, shifted by 1, i.e.,

$$\begin{aligned} g(i,j) &= f(i,j+1) && (13)\\ &= \frac{2i+(j+1)^2-(j+1)}{2} && (14)\\ &= \frac{2i+j^2+j+1)}{2}. && (15) \end{aligned}$$

Alternatively, we can write this as

$$g(i,j) = i + T_2(j), \tag{16}$$

and get the same result.

2.1.3 HIGHER DIMENSIONS

To handle three-way interactions, we need to map triples of indices in a 3-index array to a flat list, and similarly for higher-order interactions.

For three indices, i,j,k, with $i<j<k$ and $0 \leq i,j,k < D$, we have a similar recurrence. Calling the mapping h, we have

$$h(i,j,k) = i + T_2(j-1) + T_3(k-2); \tag{17}$$

if we define $T_1(i) = i$, then this has the very regular form

$$h(i,j,k) = T_1(i) + T_2(j-1) + T_3(k-2); \tag{18}$$

and from this, the generalization to higher dimensions is straightforward. The formulas for "higher triangular numbers", i.e., those defined by

$$T_k(n) = \sum_{i=1}^{n} T_{k-1}(n) \tag{19}$$

for $k > 1$ can be determined inductively. For $k = 3$, the result is

$$\begin{aligned} T_3(n) &= \sum_{i=1}^{n} T_2(n) && (20)\\ &= \frac{n^3+3n^2+2n}{6}, && (21) \end{aligned}$$

so that the formula for 3-way interactions, with zero-based indexing, becomes

$$\begin{aligned} h(i,j,k) = 1 + (i-1) + \frac{(j-1)j}{2} + & && (22)\\ \frac{(k-2)^3+3(k-2)^2+2(k-2)}{6}. & && (23) \end{aligned}$$

2.1.4 HIGHER-DIMENSION POLYNOMIAL FEATURES

For the case where we include the diagonal in higher dimensions, we must shift j by 1, k by 2, and so on, and the formula becomes

$$\ell(i,j,k) = T_1(i) + T_2(j) + T_3(k), \tag{24}$$

with analogous formulas for higher degree polynomial interactions.

3 Analysis

3.1 Analytical

Calculating k-degree polynomial features via our method for a vector of dimensionality D and density d requires dD choose k (with repetition) products. The big O of the algorithm is therefore given by

$$
\begin{aligned}
O\left(\left(\binom{dD+k-1}{k}\right)\right) &= O\left(\frac{(dD+k-1)!}{k!(dD-1)!}\right) && (25)\\
&= O\left(\frac{(dD+k-1)(dD+k-2)\dots(dD)}{k!}\right) && (26)\\
&= O\left((dD+k-1)(dD+k-2)\dots(dD)\right) \text{ for } k \ll dD && (27)\\
&= O\left(d^k D^k\right) && (28)
\end{aligned}
$$

3.2 Empirical

4 Conclusion

References

JD Gergonne. The application of the method of least squares to the interpolation of sequences. *Historia Mathematica*, 1(4):439–447, 1974.

Kirstine Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 12(1/2):1–85, 1918.

Richard P Stanley. *What Is Enumerative Combinatorics?* Springer, 1986.