

Algorithms for numerical analysis in high dimensions

Beylkin and Mohlenkamp referred to in Udo's paper using AG to decompose tensors for MP2 calculation

1 introduction

complexity of algorithms grows in dimension D exponentially in D. Even accessing a vector in dimension d requires M^d operations wher M is the number of entries in each direction. called curse of dimensionality. In this paper describe how to solve linear systems and show how to deal with antisymmetric fuctions. A number of problems solved by using separation of variables. This allows one to approximate a function with complexity growing only linearly in d and compute using only one-dimensional operations, avoiding exponential dependence. If simple separation of variables is not good enough there is no way, in that framework to improve the accuracy.

The extension is to make a function a sum of functions multiplied by a weighting coefficient. This is called separated representation. An accuracy goal is developed and the functions, weighting coefficients and number of terms is adapted to acheive the goal. This is a rich structure and is not well understood. It is not a projection onto a subspace but a nonlinear method to track a function in a hgih-dimensional space while using a small number of parameters.

Many LA operations can be performed by using the separated representation. Then one can perform operation in D dimensions using combinations of one dimensional operations so computational complexity scales linearly in D, formally. The complexity also depends on the separation rank, r.

The practical question is how to keep the separation rank close to optimal during the course of a numerical algorithm.

In order to use the separated rep for num analysis app, many algorithms and operations need to be translated into this framework. Many standard methods, like gaussian elimination, do not make sense in the separated representation. In this paper 2 approaches to solve a linear system shown. AG method for sparse matrices and least squares problem.

2 The separated representation

introduce the separated representation and discuss its properties.

clarifying information.

a function f in dimension d is is a map $f : \mathbf{R}^d \rightarrow \mathbf{R}$ from d-dimensional Euclidian space to the real numbers. f is written as $f(x_1, \dots, x_d)$ where $x_i \in \mathbf{R}$

A vector \mathbf{F} in dimension d is a discrete rep of a function in dimension d on a rectangular domain. written as $\mathbf{F} = F(j_1, \dots, j_d)$ where $j_i = 1, \dots, M_i$

A linear operator \mathcal{A} in dimension d is a linear map $\mathcal{A} : S \rightarrow S$ where S is a subspace of functions in dimension d.

A matrix \mathbb{A} in dimension d is a discrete representation of a linear operator in dimension d. written $\mathbb{A} = A(j_1, j'_1; \dots; j_d, j'_d)$ where $j_i = 1, \dots, M_i$

For simplicity we assume $M'_i = M_i = M$ for all i.

We define the separation representation as

$$\sum_{l=1}^r s_l F_1^l \otimes F_2^l \otimes \dots \otimes F_d^l$$

where s_l is a scalar and F_i^l is a vector in dimension 1 with entries $F_i^l(j_i)$, and unit norm. We require

$$\|F - \sum_{l=1}^r s_l F_1^l \otimes F_2^l \otimes \dots \otimes F_d^l\| \leq \epsilon$$

sl separation values and integer r the separation rank.

A matrix is defined similarly $\mathbb{A}_i^l = A_i^l(j_i, j'_i)$ replacing the vectors $\mathbf{F}_i^l(j_i)$

Definition 2.2: The condition number of the sum equation is the ratio

$$\kappa = \frac{\sum_{l=1}^r s_l^2)^{1/2}}{\|\mathbf{F}\|}$$

In order to maintain sig digits cannot allow κ to be too large. to achieve numerical it suffices to have

$$\kappa\mu\|\mathbf{F}\| \leq \epsilon$$

where μ is the machine roundoff.

Next one will show how to do basic operations of addition, inner product and matrix vector multiplication.

Other operations such as scalar multiplication, trace, Frobenius norm, matrix-matrix multiplication, etc follow a similar pattern.

Proposition 2.3

vector representation cost requires $\mathcal{O}(drM)$ entries to store.

vector addition $\hat{F} + \tilde{F}$: vectors in separate reps can be added by merging their representations as sums and resorting. there is no appreciable comp cost but the new sep rank is nominally the sum of the two ranks

$$r = \hat{r} + \tilde{r}$$

Vector inner product $\langle \hat{F}, \tilde{F} \rangle$ can be computed via inner product of two vectors

$$\langle \hat{F}, \tilde{F} \rangle = \sum_{\hat{l}=1}^{\hat{r}} \sum_{\tilde{l}=1}^{\tilde{r}} \hat{s}_{\hat{l}} \tilde{s}_{\tilde{l}} \langle \hat{F}_1^{\hat{l}}, \tilde{F}_1^{\tilde{l}} \rangle \dots \langle \hat{F}_d^{\hat{l}}, \tilde{F}_d^{\tilde{l}} \rangle$$

Matrix-vector multiplication $\mathbb{A}\mathbf{F}$ looks like

$$\mathbf{G} = \mathbb{A}\mathbf{F} = \sum_{\hat{l}=1}^{r_A} \sum_{\tilde{l}=1}^{r_F} s_{\hat{l}}^A s_{\tilde{l}}^F (AF)1 \otimes \dots \otimes (AF)d$$

the resulting separation rank is $r_G = r_A r_F$

The representation has been studied under the name of canonical decomposition and parallel factors analysis

look to references 14, 27, 28, 13, 12

Other reps have been proposed that are more easily studied, higher order SVD is generalization of SVD but as a consequence has computational complexity that depends exponentially on d . Configuration interaction (CI) methods use a representation that looks like the separation representation but with additional constraint that all the vectors are selected from a master orthonormal set.

2.1 Classes of functions

General functions in higher dimensions.

the class of complete asymptotically smooth functions is considered

The sparse grid approach in [46] considers function classes that allow hierarchically arranged efficient grids in high dimensions. each grid point corresponds to a separable basis function. This construction yields a separated representation with separation rank equal to the number of grid points.

No current methods to characterize a function with low separation rank.

3 Reducing the separation rank

To reduce the separation rank of a matrix \mathbb{A} we can treat the component matrices \mathbb{A}_i^l as vectors using the Frobenius inner product and norm.

For the timings presented the optimal solution is not necessary (doing the CP-ALS method that I already made) A nearly optimal representation will have similar computational cost and is easier to construct. A suboptimal representation is preferred when it has better conditioning.

3.1 Method

Solve the problem as you are doing optimize a guess, either calling the core routine to create a best solution until the solution F(which approximates G) does not change, or increase the rank approximation.

1. Alternating Least Squares

For a given r the best separated rep is that which minimizes the error $\|F - G\|$ subject to the condition number constraint. First algorithm ignores constraint on κ then in section section it will be taken into account.

min is a nonlinear least-squares problem. To solve exploit its multilinearity and use an ALS approach.

ALS is used in statistic for similar purpose in 21,28,29,8,15,40

input is a vector in separated representation (rather than a dense data vector in dimension d)

ALS Takes F and refines one direction k, at a time then "alternate" the direction k=1,...,d. To refine direction k, fix all other directions and solve for new F_k^l

example of linear least squares problem solved using standard linear algebra in 17

Taking the gradient of $\|F - G\|$ wrt the vector entries in direction k and setting =0, one obtains a linear system to solve for the updated vector entries.

For fixed K form the matrix B with entries

$$B(lh, lt) = \Pi_{i \neq k} < F_i^l t, F_i^l h >$$

for fixed coordinate jk, form the vector bjk with entries

$$b_{jk}(lh) = \sum_{l=1}^{r_G} s_l^G G_k^l(j_k) \Pi_{j \neq k} < G_i^l, F_i^{lh} >$$

Then solve the eigenvalue equation

$$BC_{j_k} = b_{j_k}$$

The norm of c is the singular value and the vector of F = c/s.

2. Controlling the condition number

The minimization can be ill-conditioned so we control the condition number κ Instead of minimizing the normal problem we minimized

$$\|F - G\|^2 + \alpha(\kappa(\|F\|))^2 = \|F - G\|^2 + \sum_{lt}^{rf} (s_{lt}^F)^2$$

where alpha is a small parameter. this is called regularization. The modification of the als algorithm is simple just a redefinition of a singular value term, just add alpha*Identity to B.

4 Solving a linear system.

How to solve a linear system $AF = G$ for F, where all objects are in the separated representation. One standard method is Gaussian elimination LU factorization. Can't use this in separated representation. Better with iterative algorithms. First approach apply one of the iterative methods designed for large sparse systems. Second approach to formulate the system as a least squares problem then solve this joint problem method similar to previously stated.

4.1 iterative algorithm

Under the assumption $\|\mathbb{I} - \mathbb{A}\| < 1$ one method for solving the system is the iteration

$$F_{k+1} = (\mathbb{I} - \mathbb{A})F_k + G$$

This requires only matrix-vector multiplication and vector addition both which can be performed at linear cost in d . Rank is required to be reduced since LA ops increase rank.

Steepest decent and conjugate gradient methods only require matrix vector multiplication, vector addition and vector inner product. One can construct their iterations such that

$$F_{k+1} = (I - c\mathbb{A}^*\mathbb{A})F_k + c\mathbb{A}^*G$$

where c is chosen to make $c\|\mathbb{A}^*\mathbb{A}\| < 1$.

4.2 Finding a low separation-rank solution to a linear system.

The problem of solving a linear system $AF = G \rightarrow \|AF - G\|$ can be cast as a minimization problem.

There is a constraint on the rank and therefore solve the linear system and find a reduced separation-rank representation for the solution. The algorithm is similar to the one discussed in section 3. There are some extra terms but if the matrix A