New developments in nonnegative matrix factorization

Suresh Venkatasubramanian

March 13, 2013

1 Introduction

Let us start with three different puzzles.

Shadow polytopes Let's play shadow games. I want to create an outline of a specific (convex) polygonal shape on a wall.

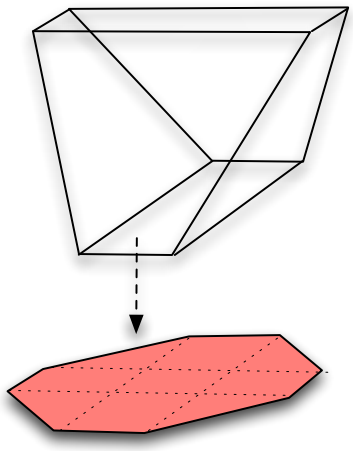

Figure 1: The regular octagon as the shadow of a polytope with 6 faces[5]

> *What is the minimum (face) complexity of a three-dimensional (or even higher-dimensional) polytope whose shadow on the wall is the shape I need ?*

Positive explanations The *singular value decomposition* is often used to determine a small collection of points that can linearly "explain" a data set. When we write an $n \times d$ matrix

M as $M = UV$, where V is $k \times d$, we're saying that we can "explain" the points in M using the (fewer) points in V.

But this explanation might involve negative combinations of points, and this can be hard to interpret. A *positive* combination of points can be viewed as a convex combination (after scaling), which can in turn be interpreted as a probability distribution over points.

Can we design a concise positive explanation of a point set ?

Communication Complexity Alice and Bob have d-bit inputs x and y respectively, and wish to compute $f(x, y)$ by communicating information between each other. The *deterministic* communication complexity of this problem is the minimum amount of (deterministic) communication needed to determine $f(x, y)$. This can be lower bounded by the *nondeterministic* communication complexity, in which both Alice and Bob are given a "certificate" that they can use to verify their answer.

How do we compute the nondeterministic communication complexity of a function ?

All of these problems require the same notion: the *nonnegative rank* of a matrix (and a nonnegative matrix factorization (NMF)).

1.1 Nonnegative Matrix Factorization

To understand the NMF, it's helpful to start with the singular value decomposition. Let M be an $n \times d$ real-valued[1] matrix. Then the *singular value decomposition* (SVD) of M is given by

$$M = U\Sigma V^\top$$

where U is an $n \times n$ orthonormal matrix, V is $d \times d$ orthonormal, and Σ is a rectangular diagonal matrix of nonnegative reals called *singular values*.

There's a natural geometric interpretation of the SVD if we think of M as a linear transform. It says that the operation of applying A can be viewed as doing a d-dimensional rotation (V), an axis scaling and lifting (Σ) and then an n dimensional rotation (U).

On the other hand, suppose we think of M as a collection of n d-dimensional points (expressed as row vectors). If the rank of M is $r(M) = r$, then Σ has k nonzero values. This means that we can rewrite M as

$$M = AW$$

where A is $n \times k$ and W is $k \times d$. Note that $k \leq \min(n, d)$. We can reinterpret the points in M as being "generated" by a collection of k points in W. In particular, each point in M

[1] The SVD is defined for complex-valued matrices as well, but we don't need that level of generality here.

can be written as a *linear* combination of the k points in W (where the matrix A describes the coefficients).

Moreover, if we wished to represent the data by a "small" collection of basis vectors (say $r < k$), we can merely take the top r singular values of Σ and zero out the rest. Suppose the resulting rank-r matrix is $M' = U\Sigma' V^\top$. The key property of the SVD is that M' minimizes the quantity $\|M - X\|_F$ over all rank-r matrices X, and so this is a good *low-rank* approximation of the data. This is perhaps the most surprising and wonderful aspect of the SVD: that it allows us to solve the optimization $\min \|M - X\|_F$ over all rank-k matrices X (which is a nonconvex constraint). Here $\|M\|_F = \mathrm{Tr}(M^\top M) = \sum m_{ij}^2$ is the Frobenius norm; the difference $\|X - M\|_F^2$ is the sum of squares of the Euclidean distances between corresponding points.

NMF One problem of using the SVD as a feature transformation is that the features are hard to interpret. Often, a goal of feature transformation and dimensionality reduction is to express a complex concept as a combination of simpler concepts. The SVD allows us to express each point of the data as a *linear* combination of exemplars, but what does it really mean to say that a document can be expressed as the *difference* of two other documents ?

Note that this is really a problem with the signs. If all weights are positive, then the combination (normalized) can be interpreted stochastically: we can interpret an object is the (scaled) expected value of a probability distribution over exemplars. This quest for *interpretability* has led researchers to investigate other ways[9, 10] to factorize a collection of points, and one of the more popular ones is called the *nonnegative matrix factorization*.

Definition 1.1 (Nonnegative Matrix Factorization (NMF))**.** *Given an $m \times n$ matrix M with nonnegative entries, find matrices $A \in \mathbb{R}_+^{m \times k}, W \in \mathbb{R}_+^{n \times k}$ with nonnegative entries such that*

$$M = AW^\top$$

The minimum value of k such that this factorization exists is called $\mathrm{rank}_+(M)$, the *nonnegative rank* of A. Obviously, $r(M) \leq \mathrm{rank}_+(M)$.

1.2 Back to the puzzles.

We've already seen the connection between nonnegative matrix factorization and probabilistic factorization of a set of points. What about the other puzzles ?

Shadow Polygons Our first puzzle was about constructing shadows of polygons. If I have a regular n-gon in the plane, how many facets can I manage with ? It turns out that for n-gons, you can do this with $O(\log n)$ facets[5, 2, 7] and this is tight. For non-regular n-gons this is a lot harder, and you might need $\Omega(\sqrt{n})$ facets, or at least $\Omega(\sqrt{n}/\sqrt{\log n})$ facets for "well-behaved" non-regular polygons[5].

In general, the notion we're looking at is called *the extension complexity* of a polygon (or a polytope in general).

Definition 1.2 (Extension Complexity)**.** *Let P be a polytope in $\mathbb{R}^d$. The* extension complexity *of P, denoted as $xc(P)$, is the minimum number of facets in a polytope $Q \in \mathbb{R}^{d+k}$ such that P is the projection of Q onto $\mathbb{R}^d$.*

The extension complexity of a polytope is a very important notion in optimization. It is part of a larger set of results that roughly say that "lifting reduces complexity". Just like the Veronese embedding linearizes complex boundaries and allows us to use linear reasoning, a small extension complexity means that we can describe a collection of constraints as a projection of many fewer constraints in a higher dimensional space. This idea is the basis for much of the recent excitement in lift-and-project schemes, and was used to show recently that there is no polynomial-sized lifted linear program for the travelling salesman problem[4].

Computing the extension complexity appears to be difficult because one might have to enumerate over a large class of higher dimensional polytopes. A beautiful result of Yannakakis characterized the extension complexity of a polytope in an elegant way.

Let P be given by the m facet-defining inequalities $\mathbf{A_i x} \leq b_i$. Let the vertices of P be $\mathbf{v_1}, \ldots, \mathbf{v_n}$. Define the *slack matrix* S as

$$S_{ij} = b_i - \mathbf{A_i v_j}$$

Note that S is nonnegative since all vertices are feasible.

Theorem 1.1 ([14])**.** *$xc(P) = rank_+(S)$.*

Communication Complexity Let M be a Boolean matrix. Suppose we select a subset of rows and columns. We get a submatrix of M that we call a *combinatorial rectangle*. Let the *rectangle complexity* $\mathrm{rc}(M)$ be the number of combinatorial rectangles required to cover all the 1s in M.

Now suppose that M is a $2^d \times 2^d$ matrix where M_{ij} is the value of the (Boolean) function $f(i, j)$. The nondeterministic communication complexity of a protocol for computing f is precisely $\log \mathrm{rc}(M)$[8].

If for a nonnegative matrix M we have $\mathrm{rank}_+(M) = r$, then by definition there must exist nonnegative vectors $\mathbf{u_i}, \mathbf{v_i}, 1 \leq i \leq r$ such that

$$M = \sum_i \mathbf{u_i v_i}^\top$$

Intuitively, you can think of each term in the summation as "covering" some elements of M. This intuition connects nonnegative matrices and communication complexity. In particular, Yannakakis[14] showed:

Theorem 1.2.

$$rc(M) \leq rank_+(M)$$

2 Computing the NMF

As the above examples show, the nonnegative rank of a matrix (and its nonnegative factorization) are important quantities that connect a purely geometric notion with fundamental problems in theoretical computer science.

Unlike the SVD however, the NMF is very hard to compute ! While a number of heuristics have proposed for computing a good factorization, the true complexity remained unknown until 2008, when Steve Vavasis[13] showed that the following decision problem was NP-complete:

Definition 2.1 (EXACT NMF)**.** *Given an $m \times n$ nonnegative matrix M of* promised *rank k, do there exist nonnegative matrices $A \in \mathbb{R}^{m \times k}$ and $W \in \mathbb{R}^{k \times n}$ such that $M = AW$?*

The key reduction is to show that EXACT NMF is equivalent to a shape fitting problem called INTERMEDIATE SIMPLEX. The proof is then completed by showing that INTERMEDIATE SIMPLEX is NP-complete by a reduction from 3SAT.

Definition 2.2 (INTERMEDIATE SIMPLEX)**.** *Given a polyhedron $Ax \geq \mathbf{b}$ where $A \in \mathbb{R}^{n \times (k-1)}$ and $[A\ \mathbf{b}]$ has rank k, and a set of m points $S \subset P$ that is affinely independent, is there a $(k-1)$-simplex Δ such that $S \subset \Delta \subset P$?*

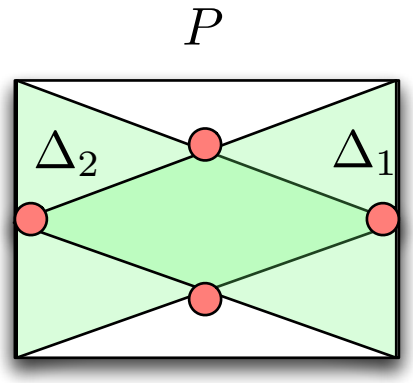

Figure 2: Illustration of INTERMEDIATE SIMPLEX. Two possible simplices Δ_1 and Δ_2 can cover the marked points S while staying within the polytope P [13].

It's worth pointing out that the equivalent of the NMF with the INTERMEDIATE SIMPLEX problem shows that the problem can be solved easily for rank 2 matrices, since in this case the set of points lie on a line, and the covering simplex is merely the endpoints of the points. This fact was exploited by Cohen and Rothblum[3] to find an NMF for rank-2 matrices.

But an algorithm that (say) ran in time polynomial in n and exponential in k remained elusive till last year, when Arora et al. [1] presented the first algorithm that computes the NMF for constant rank matrices in time polynomial in m and n.

Their approach was interesting, because it used a rather heavy (and familiar) hammer: the quantifier elimination method of Tarski and Seidenberg. A detailed exposition of current results in quantifier elimination and the first-order theory of the reals, while fascinating[2], is a little out of scope here. The main results we need can be summarized as follows.

Theorem 2.1. *Given s polynomial constraints of total degree d on r variables, we can find a feasible point in the semi-algebraic set defined by these constraints in time $O(sd)^{O(r)}$.*

It is not hard to set the NMF problem in this framework. Fixing a rank r, the nonnegativity constraints on A and W are linear, and the matrix product constraint is degree two. However, the number of variables required is $O(nr + mr)$, yielding an algorithm exponential in all three size parameters.

Notice that the dependence on the number of constraints and the total degree is still polynomial. This suggests that if the high dimensional polynomial system can be "factored" into a collection of many low-dimensional systems (even exponentially many in r!), we could obtain an algorithm that has only polynomial dependence on m and n.

The meat of their work addresses how to do this using only $2r^2$ variables. The idea is to construct two $r \times r$ matrices that encode the key elements of the desired matrices A, W: at least enough to reconstruct them. The constraints can be expressed as low-degree polynomials in the entries of these matrices, and then they apply the quantifier elimination hammer.

It's helpful to see how this works in the special case when A and W have full (column or row) rank. Since M has rank r, there is a basis of r vectors for its columns and another basis of r vectors for its rows. Let this first basis be $U \in \mathbb{R}^{n \times r}$, and the second basis be $V \in \mathbb{R}^{m \times r}$. This means that we can write $M = AW = UM' = M''V^\top$. Since A has rank r, it has a pseudoinverse A^+ such that $A^+A = I_r$, and similarly for W.

Now we can extract W from M by left-multiplying by A^+: $W = A^+UM'$. Similarly, we can write $A = M''V^\top W^+$. The key observation now is that $A^+U = T$ is an $r \times r$ matrix, and so is $V^\top W^+ = T'$. We can therefore express the search for A and W in terms of a search for T' and T'' such that for any M' and M'' constructed from arbitrary column and row bases for M, the following equations hold:

- TM' is nonnegative, and so is $M''T'$
- $M''T'TM' = M$.

[2] And maybe a topic for a future column !

The first condition is linear, and the second is a degree-two polynomial, and so the result follows.

When A and W do not have full rank, the pseudoinverse can't be used, and the algorithm gets much more complicated. But even there, the core idea is to identify small $(r \times r)$ matrices that control the decomposition.

More recently, Moitra[11] showed that even this viewpoint is not optimal. Through a careful examination of the decomposition, he showed that the number of variables could be reduced to $O(r^2)$ from $O(r^2 2^r)$, yielding an algorithm that runs in time $O((nm)^{r^2})$.

Optimality. Quantifier elimination is a powerful (and blunt) tool for algorithm design. While it's useful for showing that an algorithm must exist, it's essentially a brute force procedure that builds the arrangement of the polynomial constraints. So could there be a more efficient algorithm ?

It turns out that the answer is NO. Arora et al. show that assuming the Exponential Time Hypothesis (ETH)[6] (that SAT cannot be solved in sub-exponential time), the NMF cannot be computed in $(nm)^{o(r)}$ time. The proof of hardness comes via a reduction from the d-SUM problem, which was shown to be ETH-hard by Pătraşcu and Williams[12].

Approximation algorithms. Given the hardness of computing the NMF exactly, it's natural to wonder about approximations. There are different ways to pose the question: one version studied by Arora et al. [1] looks to approximate the resulting product.

Problem 2.1. *Given a nonnegative $n \times m$ matrix M, integer r and $\delta > 0$ such that $\text{rank}_+(M) = r$. Find nonnegative matrices A, W such that*

$$\|M - AW\|_F \leq \delta \|M\|_F$$

They show that with $\delta = O((r\epsilon^2)^{1/4})$, they can find the desired matrices in time $2^{\text{poly}(r \log(1/\epsilon))}\text{poly}(n, m)$.

However, if the goal (like in data mining) is to extract the A, W matrices themselves, then there's a different approximation question to ask.

Problem 2.2. *Given a nonnegative $n \times m$ matrix M such that $\text{rank}_+(M) = r$ achieved by matrices A, W, and given $\delta > 0$, find matrices A', W' such that $\|A - A'\|_F \leq \delta \|A\|_F$ and $\|W - W'\|_F \leq \delta \|W\|_F$.*

And finally, one might merely want to know $\text{rank}_+(M)$.

Problem 2.3. *Can we approximate $\text{rank}_+(M)$?*

Note that both of these questions are easy to solve (exactly) for the *rank*.

Next Steps So what *can* we do with the NMF ? To what extent can we apply the vast machinery of matrix approximations, sketching and sampling to tackle computing such factorizations ? Is there a streaming algorithm to estimate the NMF approximately ? One piece of negative news in this light comes from Moitra's paper, where he shows that a matrix of dimension $3rn \times 3rn$ might have nonnegative rank $4r$, but any submatrix with n rows has nonnegative rank at most $3r$. This suggests that finding small witnesses (the essence of sketching methods) might be tricky.

The lower bound via the ETH suggests that we can't do much better than the best known bounds for computing the NMF, but maybe we can design more practical algorithms that don't go through the quantifier elimination hammer ? If we actually want to compute the factorization, then this issue is extremely important.

3 Conclusion

The idea of nonnegative rank is not new. The awareness that nonnegative rank has a structural role in complexity is also not new. What is new here is the interest in nonnegative rank and nonnegative decompositions as an algorithms problem, fueled by the interest coming from the world of data mining[9] (which I like to think of as "high dimensional geometry with code").

What's interesting about the NMF is that it further reinforces the folk wisdom in high dimensional geometry that "ℓ_1 is hard, and ℓ_2 is easy". One can think of the NMF as an "ℓ_1 variant of the SVD", and in that sense the increased complexity is not surprising.

References

[1] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the 44th symposium on Theory of Computing*, pages 145–162. ACM, 2012.

[2] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):pp. 193–205, 2001.

[3] J. Cohen and U. Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993.

[4] S. Fiorini, S. Massar, S. Pokutta, H. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In *Proceedings of the 44th symposium on Theory of Computing*, pages 95–106. ACM, 2012.

[5] S. Fiorini, T. Rothvoß, and H. Tiwary. Extended formulations for polygons. *Discrete & Computational Geometry*, pages 1–11, 2012.

[6] R. Impagliazzo and R. Paturi. The complexity of k-sat. *2012 IEEE 27th Conference on Computational Complexity*, 0:237, 1999.

[7] V. Kaibel and K. Pashkovich. Constructing extended formulations from reflection relations. *Integer Programming and Combinatoral Optimization*, pages 287–300, 2011.

[8] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 2006.

[9] D. Lee, H. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[10] M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[11] A. Moitra. An almost optimal algorithm for computing nonnegative rank. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1454–1464, 2013.

[12] M. Pătraşcu and R. Williams. On the possibility of faster sat algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1065–1075. Society for Industrial and Applied Mathematics, 2010.

[13] S. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.

[14] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.