

Singular Value Decomposition Mathematics and Machine Learning Linear Algebra

Machine Learning

What is an intuitive explanation of singular value decomposition (SVD)?

Ad by Udacity

Learn natural language processing the project-based way.

Build a portfolio of projects with machine translation, neural networks & speech recognition!

Learn more at udacity.com

19 Answers



Jason Liu, Studies Computational Math and Data Analysis..

Updated Sep 5 2017 · Upvoted by Xinyu Zhao, [Ph.D Machine Learning & Magnetic Resonance Imaging, Auburn University \(2016\)](#) and Huang Xiao, [Machine Learning researcher at Technische Universität München](#) · Author has **66** answers and **345.2k** answer views

There is a bit of math in the beginning of this post but I also wrote a quick MATLAB program that visualizes what SVD can do to an image.

In the context of data analysis, the idea is to use a rank reduced approximation of a dataset to generalize some of the properties/structures.

Related Questions

[What is an intuitive explanation of the singular values from an SVD?](#)

[When and where do we use SVD?](#)

[What is the purpose of Singular Value Decomposition?](#)

[What is an intuitive explanation of the relation between PCA and SVD?](#)

[What are the interpretations of the individual matrices of SVD \(singular value decomposition\)?](#)

[What is an intuitive explanation for PCA?](#)

[What is the best way to describe the importance of "singular value decomposition"?](#)

[How is singular value decomposition useful for waveform analysis? How does it compare to other methods of analysis?](#)

[What is a good resource to learn about topics like singular value decomposition?](#)

[What is the relation between singular value decomposition and principal component analysis?](#)

[+ Ask New Question](#)

Still have a question? Ask your own!

What is your question?

Ask

To find a SVD of A , we must find V , Σ and U such that:

$$A = U\Sigma V^T$$

1. V must diagonalize $A^T A$

1.1. v_i are eigenvectors of $A^T A$.

2. Σ where Σ_{ii} are singular values of A .

3. U must diagonalize AA^T

3.1 u_i are eigenvectors of AA^T .

If A has rank r then:

1. $v_1 \cdots v_r$ forms an orthonormal basis for the range of A^T

2. $u_1 \cdots u_r$ form an orthonormal basis for the range of A

3. Rank of A is equal to the number of nonzero entries of Σ . From the form of this factorization

We see that we can express A another way, it can be shown that A can be written as a sum of Rank = 1 matrixes.

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

We know that by construction σ_i monotonic decreasing, the significance/weight of the n^{th} term decreases. This means that the summation to $k < r$ is an approximation \hat{A} of rank k for the matrix A .

 Still have a question? Ask your own!

What is your question?

Ask

But what does this mean?

It means that we can take a list of R unique vectors, and approximate them as a linear combination of K unique vectors.

Take this example, the image below is an image made of 400 unique row vectors.

 Still have a question? Ask your own!


What is your question?

Ask



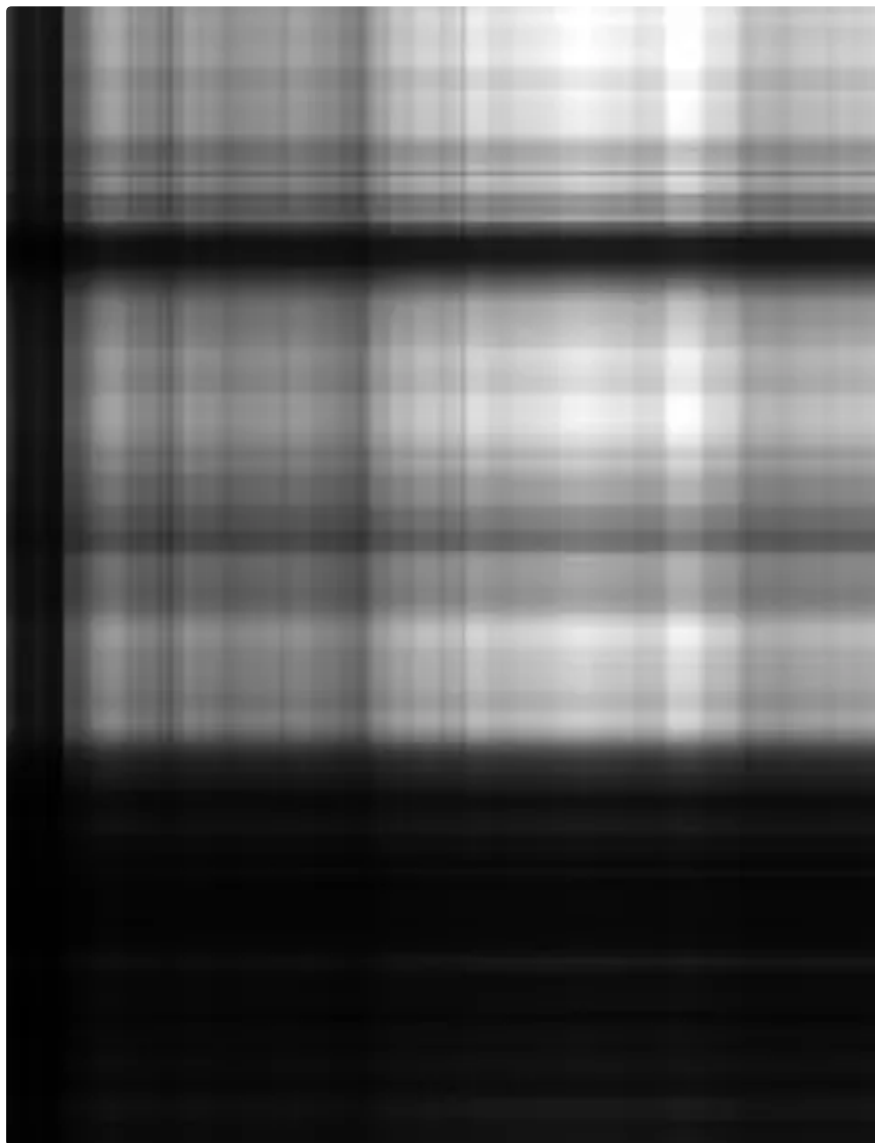
SVD is my favourite algorithm, and Feynman is my favourite scientist.

What happens if I take the first singular vector?

 Still have a question? Ask your own!

What is your question?

Ask



Notice that each row of pixels is the same... just different 'brightness'

Essentially, each row can now be written as $R_i = c_i \cdot SV_1$ where c_i is the

 Still have a question? Ask your own!

What is your question?

Ask

What happens if I take the first two singular vectors?



Now, each row can be written as the sum of two vectors

 Still have a question? Ask your own!

What is your question?

Ask

k=10



Can you believe it? With only 10 unique vectors I can almost make out the

 Still have a question? Ask your own!

What is your question?

Ask



There you have it.

Using 50 unique values and you get a decent representation of what 400 unique

 Still have a question? Ask your own!

What is your question?

[Ask](#)

Vector Space of things.

Think of another example. Say I'm netflix. And I have 100 million users and each one has watched the same 500 movies.

What I can say is that each person can be characterized in the basis of R^n where $n = 100$. So each user is in the vector space of movies.

How do I reason about this data? We have no way to 'reduce' the dimension of the vector space of movies. How do I get insight?

Maybe I'll take the SVD truncated at ... k=50.

What does this mean? This means that I'm going to find the best 50 vectors that might be a good representation of the matrix. I'm going to find the singular vectors, what I do is mathematically the following statement.

Its a change of basis from the movie basis to the first k singular vector basis. The interpretation could be that each user can be equal to

$$user = \sum^{50} c_i \cdot SV_i$$

Where SV is the singular vectors.

We can consider those 50 singular values to be 'types of users'. We may discover that SV_3 has high ratings on sports movies while SV_2 has very high ratings on horror movies.

This allows us to consider only 50 types of users and use those 50 to generalize the 500 (tada, you have dimensionality reduction)

 Still have a question? Ask your own!

What is your question?

Ask

SV₃ = sports

...

When we look at the singular values (instead of the singular vectors) we notice that a user has the values [0,4,2,0..]

This means that user = $4 \cdot SV_2 + 2 \cdot SV_3 + \dots$

Translation : This user likes horror movies and maybe watches sports too.

We get to rewrite people from the movie basis to the movie genre basis. That's mathematical, it works pretty well too.

Other examples could be from word basis to the topic basis or clothes basis to fashion style basis.

85.7k Views · View Upvoters

Promoted by Figure Eight

Chatbots are not search bars.

They need interactions, conversations, & to coordinate joint actions. Ensure yours is worth talking to.

Download at figure-eight.com

Related Questions

More Answers Below

What is an intuitive explanation of the singular values from an SVD?

 Still have a question? Ask your own!

What is your question?

Ask

What are the interpretations of the individual matrices of SVD (singular value decomposition)?

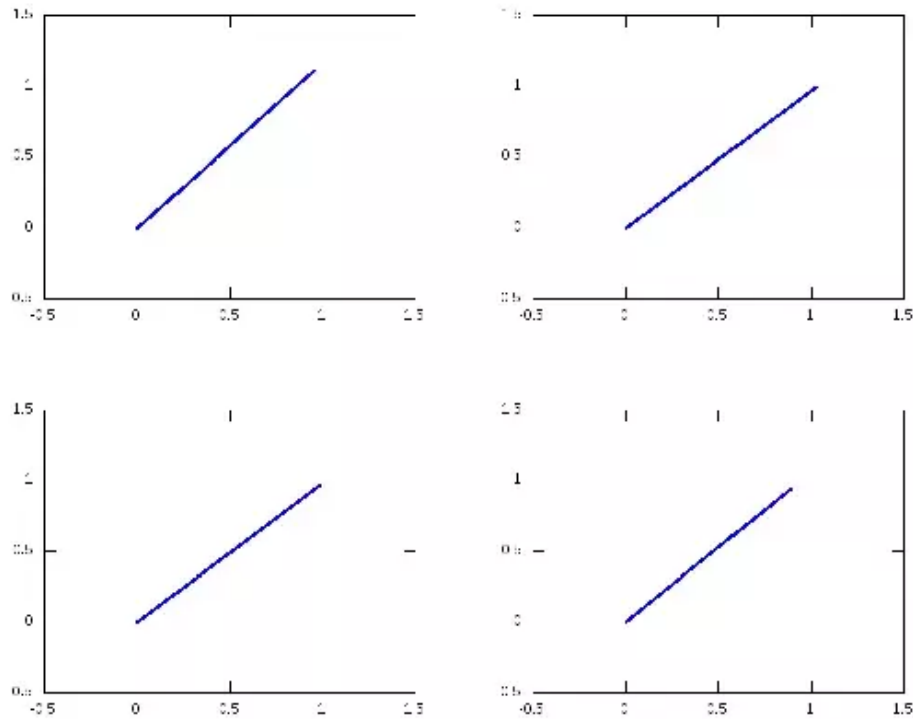
[+ Ask New Question](#)



Jerry McMahan, PhD in Applied Mathematics from North Carolina State University

Answered Oct 24 2014 · Author has **90** answers and **175.4k** answer views

Look at the lines in this figure.



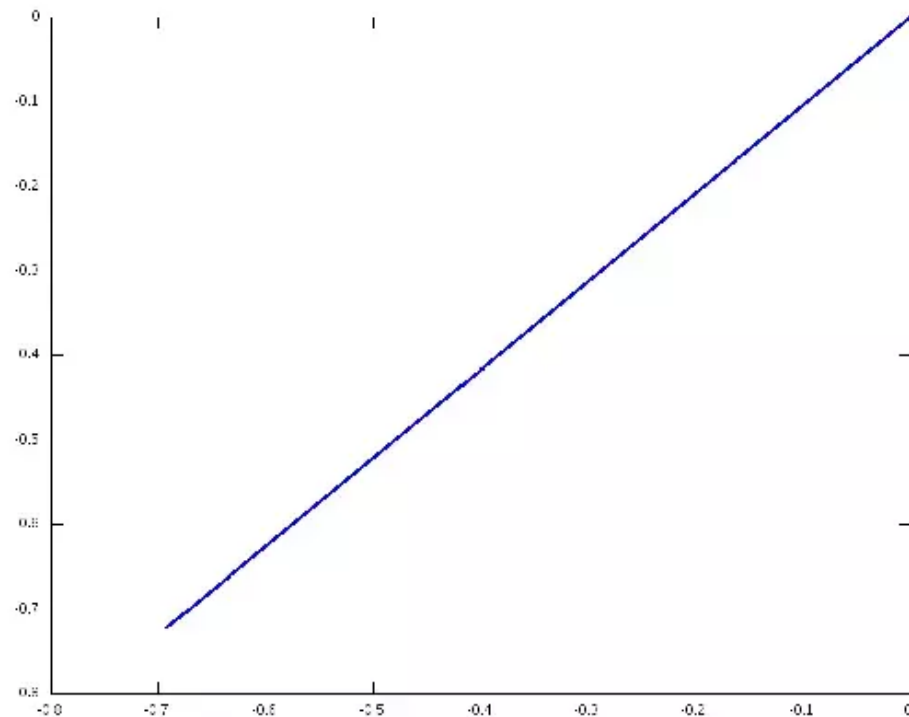
These are plots of four vectors in 2D. It is difficult to see, but they are all slightly different. Despite these differences, we can see a pattern in them. They're all

[Still have a question? Ask your own!](#)

What is your question?

[Ask](#)

take the SVD. The singular values are about 2.788 and 0.1084. There are two left-singular vectors that correspond to these values. Look at the first one, which is the one corresponding to the 1st singular value 2.788:

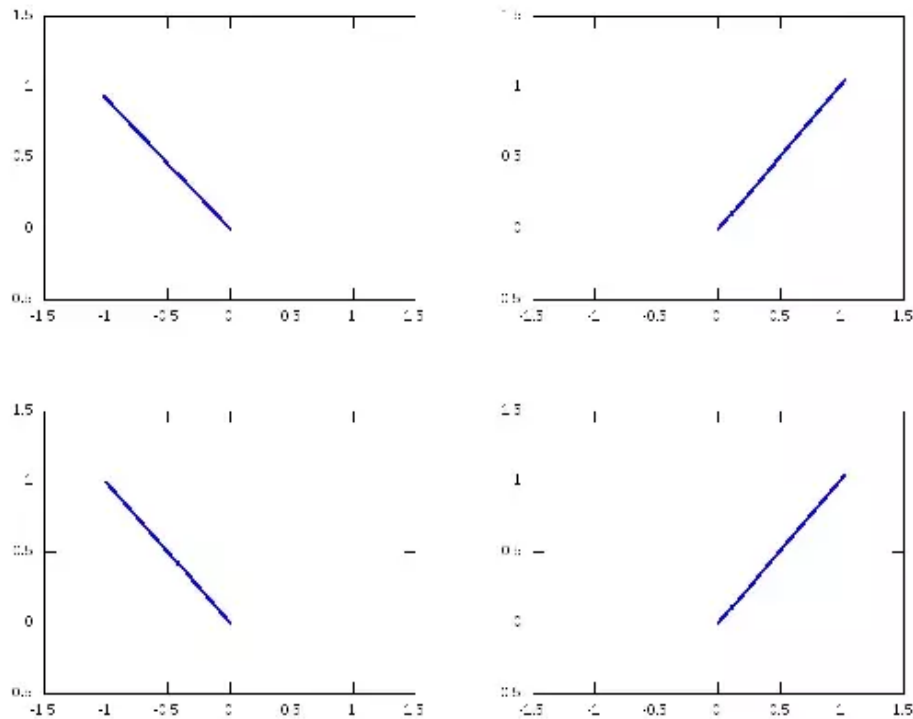


Notice this line is a unit vector (i.e., length 1) which has an angle that is about the same as the four vectors above. The SVD has extracted the pattern, i.e., it has taken four 2-D lines with approximately the same angle and represented them with a line of a similar angle. By setting the first singular value so much larger than the second (i.e., $2.788 > 0.1084$), the SVD has indicated to us that our data lies mostly along this line with a little bit of variation (this can be made more

 Still have a question? Ask your own!

What is your question?

Ask

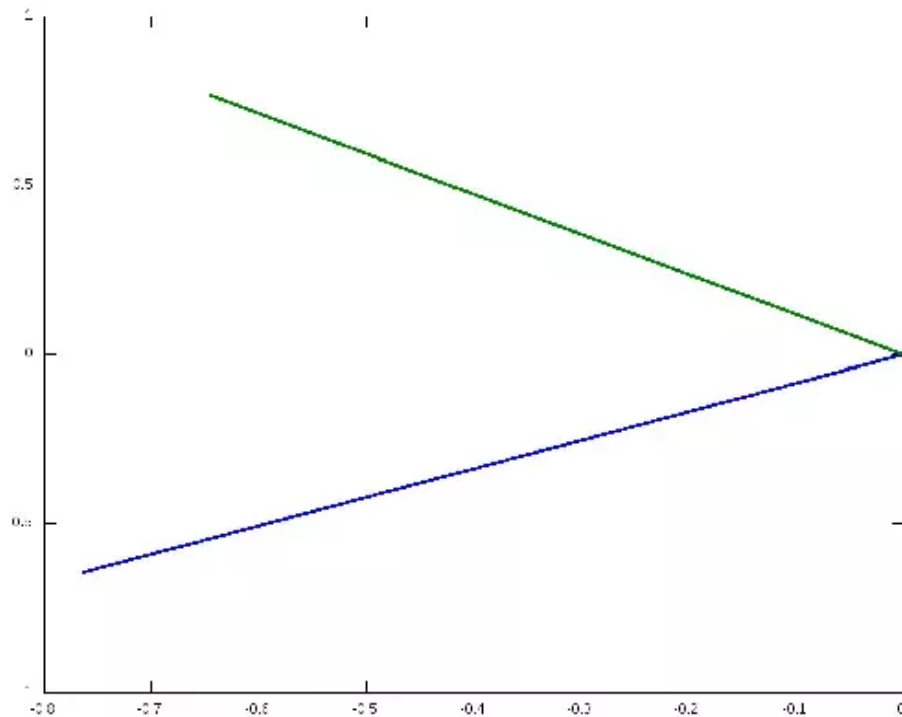


They're similar to before, only now two of the lines are approximately orthogonal to the other two (i.e., at 90 degree angles to each other - the angles don't look as close to 90 degrees visually as they actually are, because of the scaling of the axes). Doing the same as above, the SVD gives us singular values of about 2.073 and 1.976. This time we'll plot both left singular vectors rather than just the first:

 Still have a question? Ask your own!

What is your question?

Ask



Since the singular values are nearly the same in magnitude, the SVD is indicating to us that both of these vectors are equally important in representing a sample in our data so we cannot approximate with just one line. This makes sense - we have samples in our data which are at 90 degree angles to the other data and of similar magnitude - i.e., they are roughly as far away as vectors of approximately the same magnitude can possibly be in terms of the Euclidean distance.

What the SVD is doing is finding "patterns" in data. This is too ambiguous, as

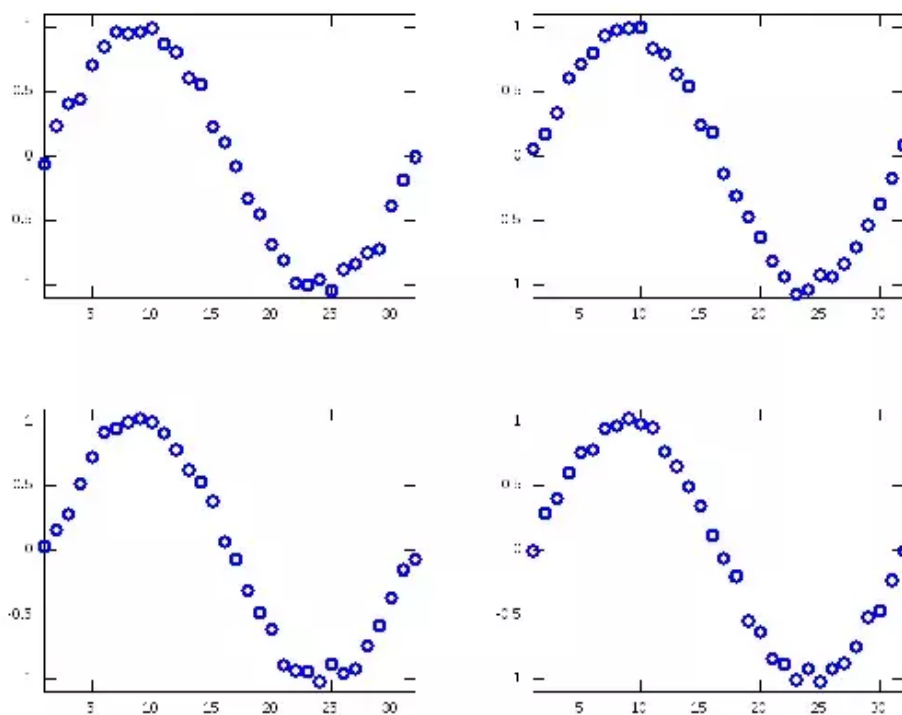
 Still have a question? Ask your own!

What is your question?

Ask

the data (first left singular-vector). After that, it adds another line to form a plane that most closely represents the data. If you have higher-dimensional data, this continues. How well this works depends on something like the "angle" between your samples (the "samples" referring to the four different lines).

In higher dimensions, this is visualized in different ways, but it can still be seen. For instance, let's take a particular set of four 32-dimensional vectors and plot each of the 32 components on the y-axis with the index of the vector on the x-axis.

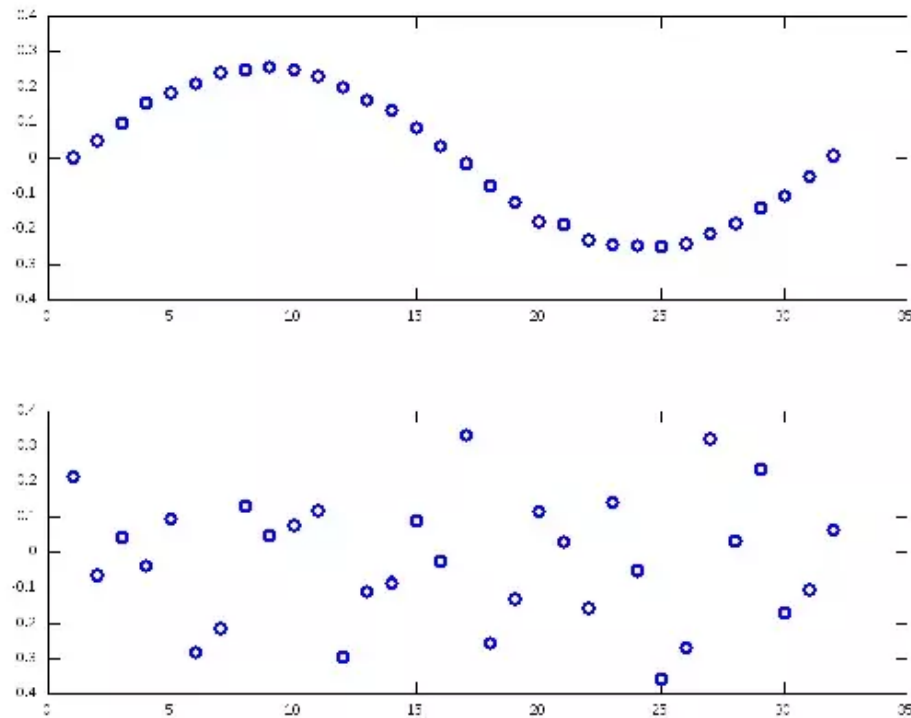


 Still have a question? Ask your own!

What is your question?

Ask

singular-vectors, plotted the same way as above, are



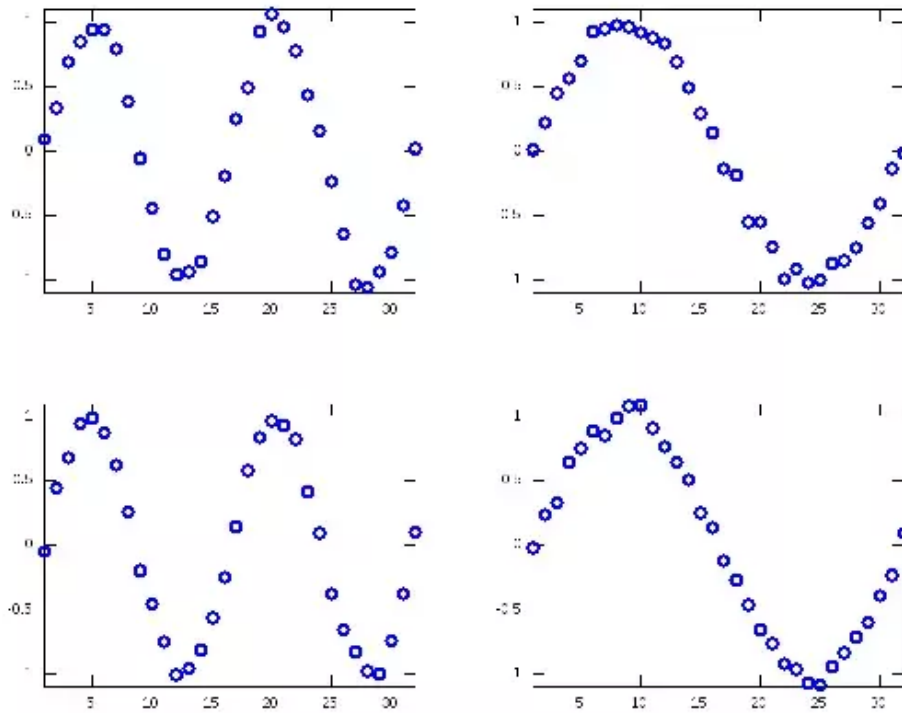
Notice how the first left singular-vector extracts the pattern that we see in our data and the second left singular-vector looks like random points. The SVD is doing the same thing it did before. The large first singular value compared to the other singular values tells us the data lies mostly along the first left singular-vector.

For a higher-dimensional example with orthogonal data, look at the following data vectors.

 Still have a question? Ask your own!

What is your question?

Ask

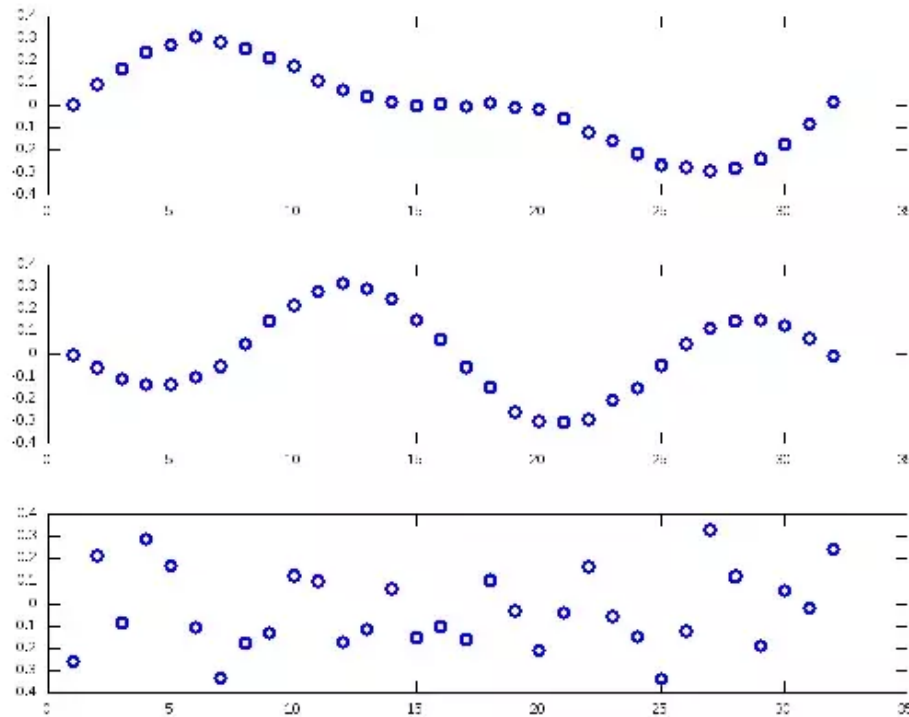


The SVD results in singular values 5.650, 5.547, 0.3671, 0.2464. The first three left singular-vectors are

 Still have a question? Ask your own!

What is your question?

Ask



Notice how regular the first two left singular-vectors are while the last one looks like random values again. Since the first two singular values are roughly equal in magnitude and much larger than the others, the data will be fairly close to a vector represented by a linear combination of the first two.

There are a variety of ways to interpret these results, but an especially popular one lately is the statistical one. The angle between vectors relates to the correlation of random variables. In particular, when random variables are represented as vectors, then correlated variables are lines with similar angles

 Still have a question? Ask your own!

What is your question?

Ask

correlated, since wealthy people can afford better medical care (among other reasons). Uncorrelated variables are those which have no relationship. For instance, if I flip a coin, and it lands on heads, this tells me nothing about who will win the Olympic gold medal in the 100-meter dash in 2016, so these two events are uncorrelated. One way to intuitively interpret the SVD, then, is that it identifies correlations in data. This is useful for all kinds of things. For instance, say 99% of people who rate Buffy the Vampire Slayer 5 stars on Netflix also rate Firefly 5 stars. Now suppose some man has rated Buffy the Vampire Slayer 5 stars but hasn't seen Firefly. Then the SVD can be used to automatically determine that Netflix should recommend that this poor guy remedy his unfortunate situation and watch Firefly already.

There are also ideas related to "smoothness" of functions that show how the SVD is helpful. Without worrying what "smoothness" means, the SVD can be shown to have rapidly decreasing singular values for highly smooth functions. This makes the SVD useful for compression algorithms because many real-life functions are "smooth".

This glosses over a lot of stuff and is more of a series of examples than an explanation, but hopefully it's helpful.

29.5k Views · View Upvoters

Promoted by Segment

What is Machine Learning? Understand the tech behind the hype.

Learn how leading companies use data to deliver better customer experiences.

Download this free guide

 Still have a question? Ask your own!

What is your question?

Ask



Answered Oct 28 2017

The answers here so far are great. However, I think the original question says “intuitive” and many of the answers here are complex. I’ll try to wave my hands some here to make things more visual. One way to understand the SVD is that it finds a transformation from an ellipsoid to the unit sphere. In linear algebra we study linear transformations of coordinate systems. Take any bases and think of each vector as an axis of a high dimensional ellipse. These axes are not necessarily orthogonal, but if we could find a way to “straighten” them out we’d end up with a sphere. In a hand-wavy way, this is essentially what SVD is doing.

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

The above decomposition can be interpreted as two rotations sandwiched around an arbitrary scaling of each axis which induces a shearing of the unit sphere. If we invert this transformation we should get back to the unit sphere and the identity matrix:

$$M^{-1} = (U\Sigma V^{\top})^{-1} = V^{-\top} \Sigma^{-1} U^{-1} = V\Sigma^{-1} U^{\top}$$

The above is called the [pseudoinverse](#) of M and it generalizes matrix inverses to non-square matrices. It would be nice to be able to do this whole operation with a collection of linearly dependent vectors too since it would allow us to solve for over constrained systems. SVD enables us to overconstrain operations like fitting models to real world data enabling us to provide more samples than the dimensionality of the data vectors. This allows us to train a model using all of our data finding the “best” fit in the linear least squares sense. The SVD has been a workhorse for the computer vision and machine learning communities

 Still have a question? Ask your own!

What is your question?

Ask

3.8k Views · View Upvoters



Duane Rich, Sr Data Scientist at LendUp

Updated May 20 · Upvoted by Luis Argerich, Data Science Professor at UBA since 1997.
and Apurv Verma, ML@GATech

I see most of these answers focus on SVD as a means of approximating/reconstructing a matrix \mathbf{A} as a sum of scaled outer products. No problem there, but what if we want to think of \mathbf{A} as a linear transformation? SVD gives us insight into the *function* that is $\mathbf{A}\mathbf{x} : \mathcal{R}^n \Rightarrow \mathcal{R}^m$, so much so that the relation has been crowned **The Fundamental Theorem of Linear Algebra**. With some math, we'll gain some real intuition.

SVD in a Nutshell

Let's be precise. Let \mathbf{A} be a rank r matrix with m rows and n columns. SVD tells

 Still have a question? Ask your own!

What is your question?

Ask

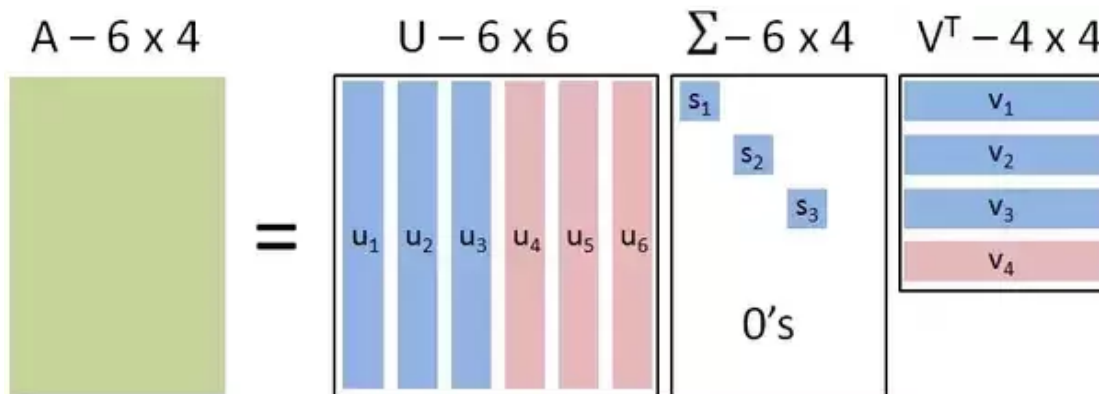
where:

1. \mathbf{U} is a square orthonormal $m \times m$ matrix. Its columns are the left singular vectors.
2. $\mathbf{\Sigma}$ is a diagonal $m \times n$ matrix with r positive values starting from the top left. These are the singular values.
3. \mathbf{V}^T is a square orthonormal $n \times n$ matrix. The rows are the right singular vectors.

Feel enlightened? Me neither, but we'll get there.

An Example and an Analogy

Let's say \mathbf{A} has $r = 3$, $m = 6$ and $n = 4$. You could visualize it like this:



Seems to be a lot of dimensions here huh? Why do we need 6 linear independent columns of \mathbf{U} to rebuild the 4 columns of \mathbf{A} ? Well, we don't - the red vectors will contribute nothing because they're multiplied by zero ultimately. However,

Still have a question? Ask your own!

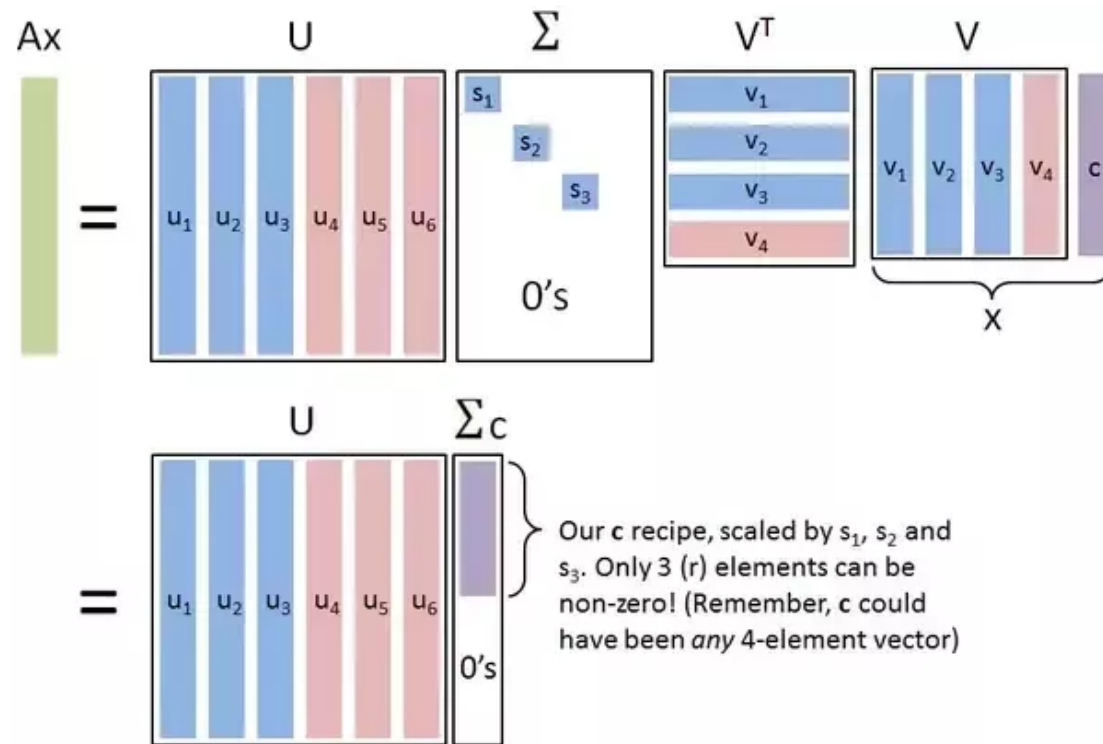
What is your question?

Ask

given \mathbf{x} as $\mathbf{V}\mathbf{c}$. Think of the vector \mathbf{c} as a *recipe* for making \mathbf{x} out of the *ingredients* that are \mathbf{V} 's columns. Now, let's apply our function:

$$\mathbf{Ax} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{c} = \mathbf{U}\mathbf{\Sigma}\mathbf{c}$$

which looks like:



Stare at that. $\mathbf{\Sigma}\mathbf{c}$ is just our \mathbf{x} recipe scaled element-wise by the singular values, leaving only 3 (r) non-zero values. With this readjusted recipe, the output $\mathbf{U}\mathbf{\Sigma}\mathbf{c}$ is just our recipe applied, but *with the ingredients as \mathbf{U} 's columns*.

I'll say that differently. \mathbf{Ax} is a linear combination of the left singular vectors

Still have a question? Ask your own!

What is your question?

[Ask](#)

1. the recipe for making \mathbf{x} out of the right singular vectors (columns of \mathbf{V}). So if \mathbf{x} is made up a lot of \mathbf{v}_2 , then the output will have a lot of \mathbf{u}_2 .
2. the 3 singular values, which scale elements of \mathbf{c} and determine which aren't crushed to zero.

That's the main idea, but its part of a bigger picture...

The Fundamental Theorem of Linear Algebra

Baked into this is an extremely important connection: SVD tells us how the *row space* (all vectors you could make with a linear combination of the *rows* of \mathbf{A}) relates to the *column space* (all vectors you could make with a linear combination of the *columns*). Specifically, its that magic number 3 - this reflects the dimension of *both* the column space and the row space. As it relates to the function output \mathbf{Ax} , there are only 3 dimensions (those of the row space) within the 4 dimensions that \mathbf{x} lives whereby movements will result in movements in the output \mathbf{Ax} , which consequently can only move in 3 dimensions (those of the column space). So imagine I gave you two vectors \mathbf{x}_1 and \mathbf{x}_2 which have recipes \mathbf{c}_1 and \mathbf{c}_2 , but the only difference was their loading for \mathbf{v}_4 . Well their outputs \mathbf{Ax}_1 and \mathbf{Ax}_2 would be the same because that loading is multiplied by zero. That dimension doesn't matter! In fact, that space (in this case, the 1 dimensional line along \mathbf{v}_4) has a name - the *null space* of \mathbf{A} .

We've now accounted for all 4 dimensions where \mathbf{x} lives. It would be nice to account for all dimension of \mathcal{R}^6 as well. Luckily, it's easy. We know the output \mathbf{Ax} must be a linear combination of the first 3 (blue) columns of \mathbf{U} (everything else is multiplied by zero) and therefore can only move in 3 dimensions. The

 Still have a question? Ask your own!

What is your question?

Ask

So we've accounted for all dimensions of \mathcal{R}^4 and \mathcal{R}^6 , with a few named subspaces. Let's review them:

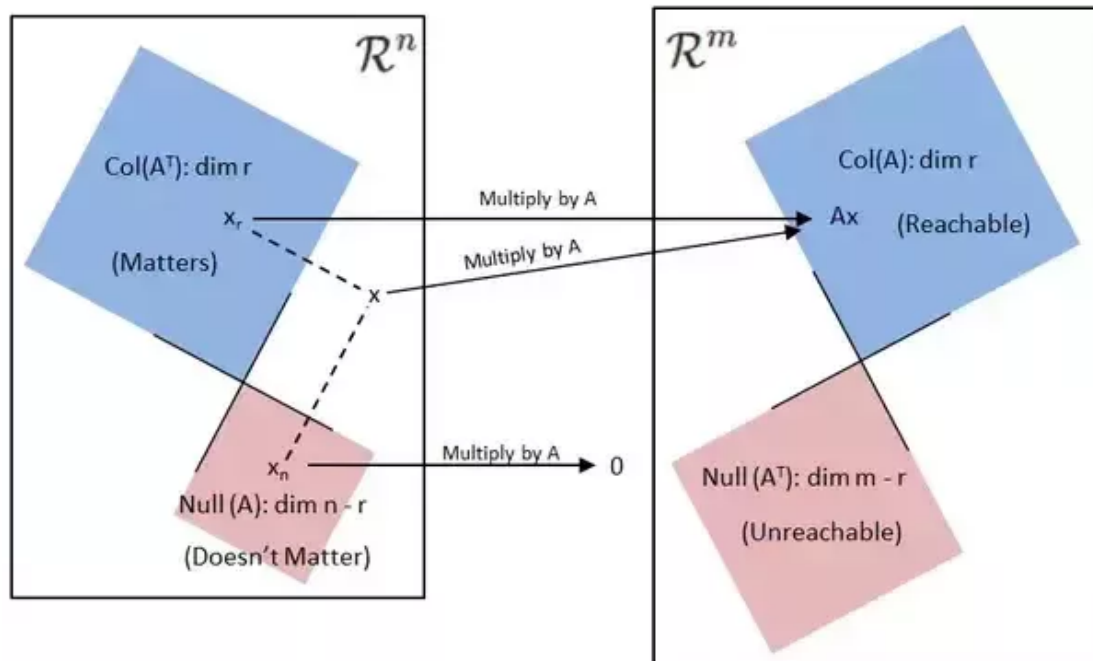
1. *null space* of \mathbf{A} , call it $\text{null}(\mathbf{A})$ - subspace of \mathcal{R}^4 where movements make no difference to the output \mathbf{Ax} . Think of it as the 'Doesn't matter' space.
2. *column space* of \mathbf{A} , call it $\text{col}(\mathbf{A})$ - subspace of \mathcal{R}^6 where \mathbf{Ax} could land with a properly chosen \mathbf{x} . Think of it as the 'Reachable' space.
3. *null space* of \mathbf{A}^T , call it $\text{null}(\mathbf{A}^T)$ - subspace of \mathcal{R}^6 where \mathbf{Ax} could never touch. Think of it as the 'Unreachable' space.
4. *row space* of \mathbf{A} , which is also the *column space* of \mathbf{A}^T , so call it $\text{col}(\mathbf{A}^T)$ - subspace of \mathcal{R}^4 which drives the difference in \mathbf{Ax} . Think of it as the 'Matters' space.

Starting to see some symmetry here? That's intentional. But first, let's think of this relation graphically. It'll help to think of \mathbf{x} as being made up of two parts, that which exists in $\text{col}(\mathbf{A}^T)$ (call it \mathbf{x}_r) and that which exists in $\text{null}(\mathbf{A})$ (call it \mathbf{x}_n). So we have $\mathbf{x} = \mathbf{x}_n + \mathbf{x}_r$.

 Still have a question? Ask your own!


What is your question?

Ask



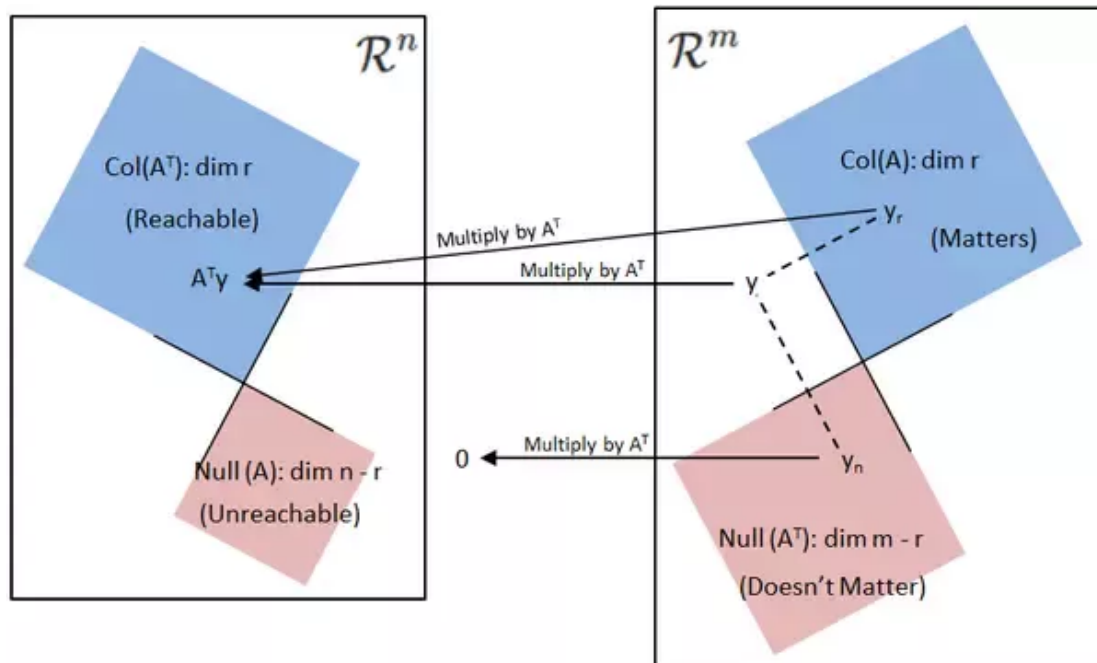
We see that $A\mathbf{x}$ and $A\mathbf{x}_r$ map to the same point. In other words, only the piece of \mathbf{x} that resides in the column space of A^T matters for determining its output, as we've seen previously.

But what if we consider a point \mathbf{y} in \mathcal{R}^m and apply A^T as a function to get $A^T\mathbf{y}$? Well, all these subspaces remain and we have a familiar, reversed relation:

 Still have a question? Ask your own!

What is your question?

Ask



Comparing these two graphics, we see the equivalence:

1. The 'Unreachable' space with multiplication by \mathbf{A} is the 'Doesn't matter' space for multiplication by \mathbf{A}^T and visa versa.
2. The 'Matters' space for multiplication by \mathbf{A} is the 'Reachable' space for multiplication \mathbf{A}^T and visa versa.

Wrapping It Up

SVD tells us how to decompose a matrix into sets of orthonormal vectors which span \mathcal{R}^n and \mathcal{R}^m . Depending on the singular values these are paired with, we know what subspace of \mathcal{R}^n matters (and doesn't matter) for driving $\mathbf{A}\mathbf{x}$ within the reachable space of \mathcal{R}^m . Specifically, $\mathbf{A}\mathbf{x}$ will be a linear combination of the

 Still have a question? Ask your own!

What is your question?

Ask

be the same. Considering multiplication by \mathbf{A}^T tells us we can think of 'Reachable' spaces as 'Matters' spaces and 'Unreachable' as 'Doesn't Matter' if we view from a different direction.

Sources

1. Gilbert Strang has an [intuitive article](#) on this. The last graphic was his idea for a cool representation.

7.2k Views · View Upvoters



Vinod Mamtani, works at OnLive

Answered Feb 1 2012 · Upvoted by Sean Owen, [Director, Data Science @ Cloudera](#)

Originally Answered: What is SVD in layman's terms?

SVD stands for Singular Value Decomposition. It's a method for matrix decomposition/factorization. The simplest way to visualize and understand how SVD is useful is to think in terms of Principal Component Analysis(PCA)/dimensionality reduction.

Let's say you want to reduce data X from n -dimensions to k -dimensions where $k < n$. In order to do this, plot the data points in an n -dimensional space. Now, draw a k -dimensional hyper-plane and project all data points onto this hyper-plane. The projection of the original data points leads to a new set of points in a reduced dimension k . It is desirable to find a hyper-plane such that the projected distance is as short as possible. This is exactly what PCA helps achieve and where SVD comes in. In order to visualize this example, think of $n=3$ and $k=2$.



Still have a question? Ask your own!

What is your question?

Ask

eigenvectors.

$[U, S, V] = \text{SVD}(\text{Sigma})$.

Some interesting notes on the result of factorization:

- The columns of matrix U form the eigenvectors of Sigma.
- The S matrix is a diagonal matrix. The values of the diagonal are called eigen values and are in descending order.

The matrix U is an $n \times n$ matrix. If you reduce the number of column vectors to k (first k), then you have obtained the k-dimensional hyper-plane in this example. The values of S give you the amount of variance retained by this reduction. To be precise, the ratio of sum of top k values along the diagonal S matrix to sum of all values along the diagonal S matrix gives the amount of variance.

Let's call the matrix U with reduced column vectors as U_{reduced} . The data set in the reduced dimension can be obtained as: $\text{transpose}(U_{\text{reduced}}) * X$

For a full treatment on SVD, check out this link: <http://nimbledais.com/?p=57>

Hope this helps.

13.2k Views · View Upvoters



Sean Owen, Director, Data Science @ Cloudera

Answered Aug 13 2012 · Author has **667** answers and **2.9m** answer views

Originally Answered: What is SVD in layman's terms?

Vinod's answer is entirely correct. Let me try a different approach that may

 Still have a question? Ask your own!

What is your question?

Ask

application of matrix factorization, like PCA. I don't know if you can explain the SVD as separate from NNMF without specialist knowledge, but you can describe a core thing it is used for.

PCA can help explain observations for very many particular things in terms of very few general things, and, that matches how many things in the world work, which is useful. If I go to your CD shelf, I'll see 100 different albums, from a world of a million albums. Maybe I see John Coltrane's "A Love Supreme" and Miles Davis's "Kind of Blue". (These happen to be famous jazz albums.)

However I don't believe you'd describe your musical preferences this way, by listing 100 albums. You'd likely say "I like Jazz." That's not only more efficient to say, but communicates more -- you likely have some affinity for ten thousand other Jazz records.

If we didn't actually think and 'like' things in terms of genres, it'd be a lot harder to reason about tastes. Every album would be an island unto itself and say little about your preference for others. But, because we have the underlying idea of "Jazz", suddenly by knowing these are "Jazz" albums I have a world of more informed guesses about your interest in other Jazz albums like by Charles Mingus.

PCA is trying to find those underlying features, "genres" in the case of music. It



Still have a question? Ask your own!

What is your question?

Ask

item-feature associations (i.e. how much each album is a Jazz album, a Rock album). We also get a third output saying how relatively important these are in explaining tastes.


From there you can do useful things, not least of which are things like recommendation, filling in your CD shelf, with albums that this model predicts you like. You can efficiently compare albums' / users' similarity in terms of few features. You can even decide to throw out or add genres (keep more/less of S) to create a more or less nuanced model of genres.

The SVD is most of the machinery that enables the above. How it operates is not possible to explain to the layman, nor maybe useful to explain. What it does (well, what PCA does) definitely can be.

24.7k Views · View Upvoters

Related Questions

- What is an intuitive explanation for PCA?
- What is the best way to describe the importance of "singular value decomposition"?
- How is singular value decomposition useful for waveform analysis? How does it compare to other methods of analysis?
- What is a good resource to learn about topics like singular value decomposition?

 Still have a question? Ask your own!

What is your question?

Ask


What is the singular value decomposition useful for?

In singular value decomposition (SVD), is the order of the matrix rows and columns important?

Linear Algebra: How do I interpret singular value decomposition (SVD) for visualization?

Is there a generalized version of the complex singular value decomposition?

+ Ask New Question

 Still have a question? Ask your own!

What is your question?

Ask