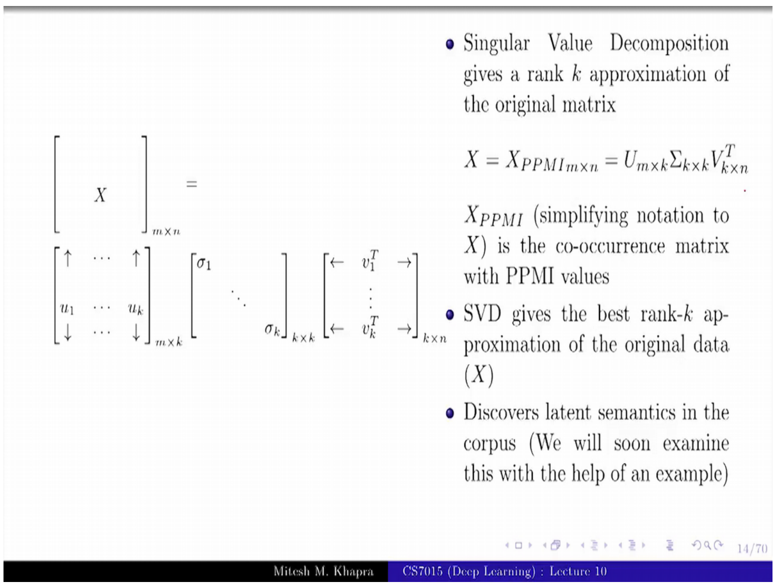
Deep Learning
Prof. Mitesh M Khapra
Department of Computer Science Engineering
Indian Institute of Technology, Madras

Module – 10.3
Lecture – 10
SVD for learning word representations

So, in this module we will talk about using SVD for learning word representations.

(Refer Slide Time: 00:18)



$$[X]_{m \times n} = \begin{bmatrix} \uparrow & \cdots & \uparrow \\ u_1 & \cdots & u_k \\ \downarrow & \cdots & \downarrow \end{bmatrix}_{m \times k} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k \times k} \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k \times n}$$

- Singular Value Decomposition gives a rank k approximation of the original matrix

$$X = X_{PPMI\, m \times n} = U_{m \times k} \Sigma_{k \times k} V^T_{k \times n}$$

X_{PPMI} (simplifying notation to X) is the co-occurrence matrix with PPMI values

- SVD gives the best rank-k approximation of the original data (X)
- Discovers latent semantics in the corpus (We will soon examine this with the help of an example)

14/70

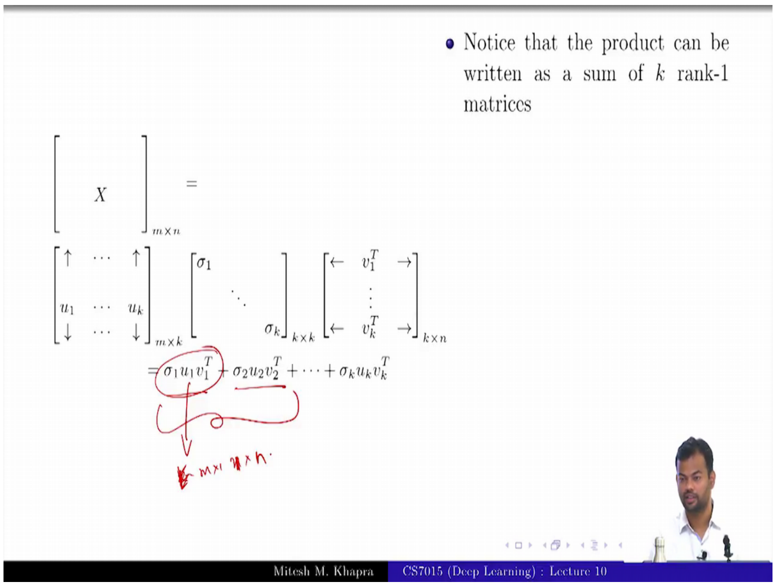
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, what does singular value decomposition? Do yeah these are all possible variants. So, people have tried various things and one of the PPMI one is the is the most reliable thing that is what is given. But you can think of, I mean you said one there are 10 different things which we can do for the co-occurrence matrix right, but this is the most popular and most stable thing to do.

Yeah what is the single value decomposition do, can you read it from the slide please. It gives the rank k approximation of the matrix. So, let me start defining a few things. So, from now on when I refer to the co-occurrence matrix I would mean the XPPMI matrix right which was the positive PMI which was replacing all negative PMIs by 0, and just do not have this nasty variable I will just call it as X.

So, from now on whenever I say X, I mean the positive PMI co-occurrence matrix. So, that is what this matrix is and we know that SVD gives us this reconstruction of the original matrix. And fine it gives us the best rank k approximation of the original matrix, and it discovers the latent semantics in the corpus. Everyone remembers this like that is what we were by we were using PC and SVD and auto encoders it was able to discover some latent semantics. And we will concretize this intuition with the help of our current example, but for now I just want you to recall that it helps in discovering the latent semantics.

(Refer Slide Time: 01:49)



- Notice that the product can be written as a sum of k rank-1 matrices

$$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}_{m\times n} =$$

$$\begin{bmatrix} \uparrow & \cdots & \uparrow \\ u_1 & \cdots & u_k \\ \downarrow & \cdots & \downarrow \end{bmatrix}_{m\times k} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k\times k} \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k\times n}$$

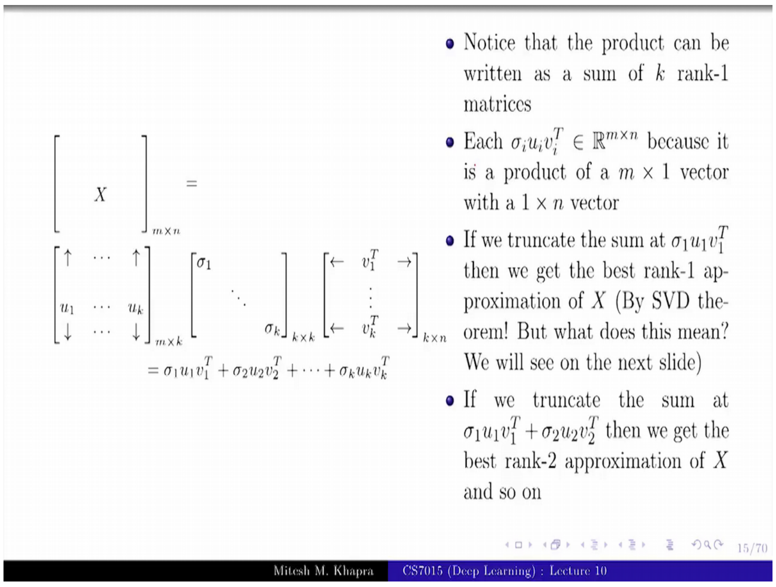
$$= \underbrace{\sigma_1 u_1 v_1^T}_{m\times 1 \; 1\times n} + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$$

Now, notice that this product and I think I have done this in one of the assignments or something can be written as a sum of the following products right. So, I can write it as sigma 1 u 1 v 1 transpose, sigma 2 u 2 v 2 transpose and so on. Can you tell me what this sum is this is the rank 2 approximation of the original matrix and I keep taking more terms I get more and more rank approximations of the original matrix? Now, and we all know that, we all hopefully know that what is the dimension of this? It is a scalar vector matrix, scalar vector matrix.

Student: (Refer Time: 02:27).

Ok. Now of course, you will say matrix, but what is the dimension of the matrix; why is it a matrix? It is an outer product of 2 vectors right this, what is the size of this? n cross 1 into n cross 1 so that; sorry 1 cross n that gives you n cross n matrix. Everyone gets this otherwise how is it a rank 1 approximation you have to get the original dimensions right. Everyone is clear with this is an outer product.

(Refer Slide Time: 02:49)



$$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}_{m\times n} =$$

$$\begin{bmatrix} \uparrow & \cdots & \uparrow \\ u_1 & \cdots & u_k \\ \downarrow & \cdots & \downarrow \end{bmatrix}_{m\times k} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k\times k} \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k\times n}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$$

- Notice that the product can be written as a sum of k rank-1 matrices
- Each $\sigma_i u_i v_i^T \in \mathbb{R}^{m\times n}$ because it is a product of a $m \times 1$ vector with a $1 \times n$ vector
- If we truncate the sum at $\sigma_1 u_1 v_1^T$ then we get the best rank-1 approximation of X (By SVD theorem! But what does this mean? We will see on the next slide)
- If we truncate the sum at $\sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T$ then we get the best rank-2 approximation of X and so on

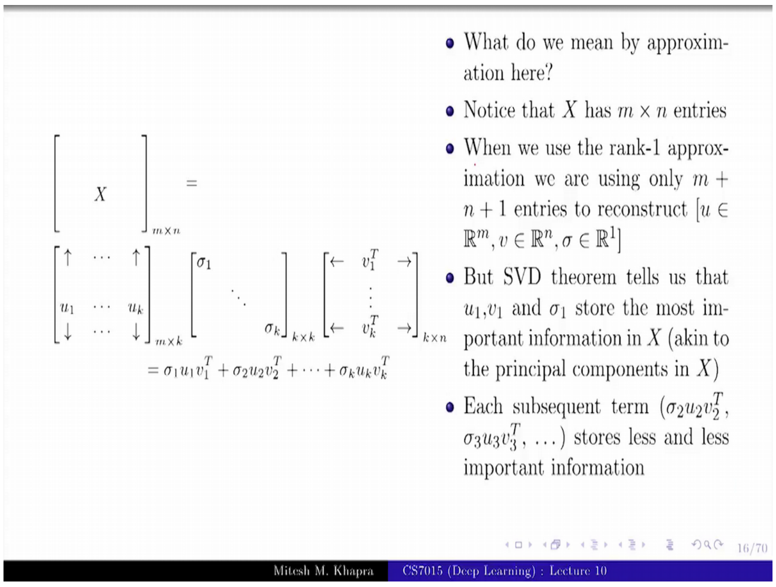
15/70

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

And it belongs to R m cross n. Ok and if we truncate the sum at the first term we get the rank 1 approximation and by SVD theorem we know that this is the best rank 1 approximation.

Now, what does this actually mean? That this is an approximation, what do we mean by that? So, we will see that on the next slide and similarly in the same way if we truncate it in the second term you get the same best rank 2 approximation.

(Refer Slide Time: 03:14)



- What do we mean by approximation here?
- Notice that X has $m \times n$ entries
- When we use the rank-1 approximation we are using only $m + n + 1$ entries to reconstruct $[u \in \mathbb{R}^m, v \in \mathbb{R}^n, \sigma \in \mathbb{R}^1]$
- But SVD theorem tells us that u_1,v_1 and σ_1 store the most important information in X (akin to the principal components in X)
- Each subsequent term ($\sigma_2 u_2 v_2^T$, $\sigma_3 u_3 v_3^T$, ...) stores less and less important information

$$\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}_{m \times n} = \begin{bmatrix} \uparrow & \cdots & \uparrow \\ u_1 & \cdots & u_k \\ \downarrow & \cdots & \downarrow \end{bmatrix}_{m \times k} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k \times k} \begin{bmatrix} \leftarrow & v_1^T & \rightarrow \\ & \vdots & \\ \leftarrow & v_k^T & \rightarrow \end{bmatrix}_{k \times n}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$$

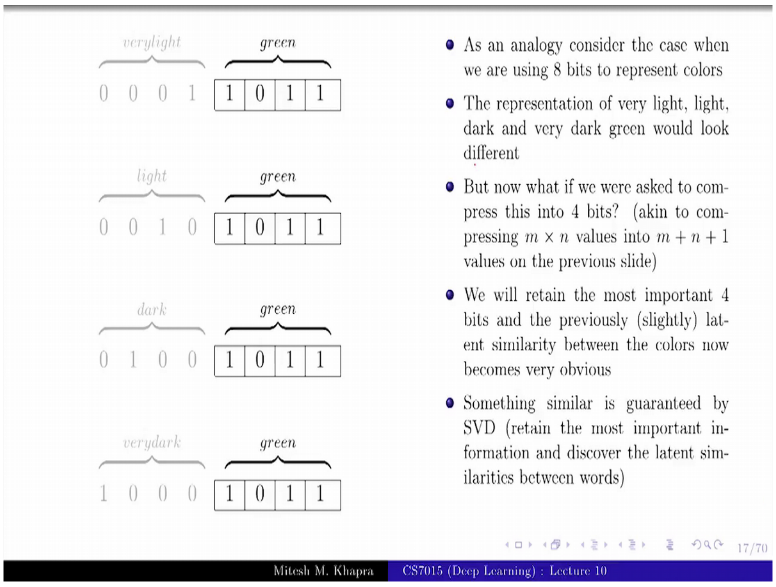
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

16/70

Now, what do we mean by approximation here? Actually and I mean to say approximation always in this course at least try to think in terms of compression. How many elements are there in the original matrix? m cross n, that is how many elements you need to describe the matrix completely. If you do a rank one approximation how many elements are you using? m plus n plus 1 right; So, the original matrix has m cross n entries, entries and when you do a rank 1 approximation you have m plus n plus 1 entry. So, that that is the approximation right. So, you are trying to really compress the original data using only these many variables you get that.

And if we do a rank 2 twice this right; So, as many rank I mean as deeper as you go in the sum you will have that many elements to do the approximation, but what is important is that the SVD theorem tells us that this is not just any random approximation. But this is the best approximation that you could have done; that means, if you wanted to use only these many elements these are the best elements to use, right? Everyone gets that.

(Refer Slide Time: 04:16)


verylight
green
0 0 0 1 | 1 | 0 | 1 | 1
light
green
0 0 1 0 | 1 | 0 | 1 | 1
dark
green
0 1 0 0 | 1 | 0 | 1 | 1
verydark
green
1 0 0 0 | 1 | 0 | 1 | 1

- As an analogy consider the case when we are using 8 bits to represent colors
- The representation of very light, light, dark and very dark green would look different
- But now what if we were asked to compress this into 4 bits? (akin to compressing $m \times n$ values into $m + n + 1$ values on the previous slide)
- We will retain the most important 4 bits and the previously (slightly) latent similarity between the colors now becomes very obvious
- Something similar is guaranteed by SVD (retain the most important information and discover the latent similarities between words)

17/70

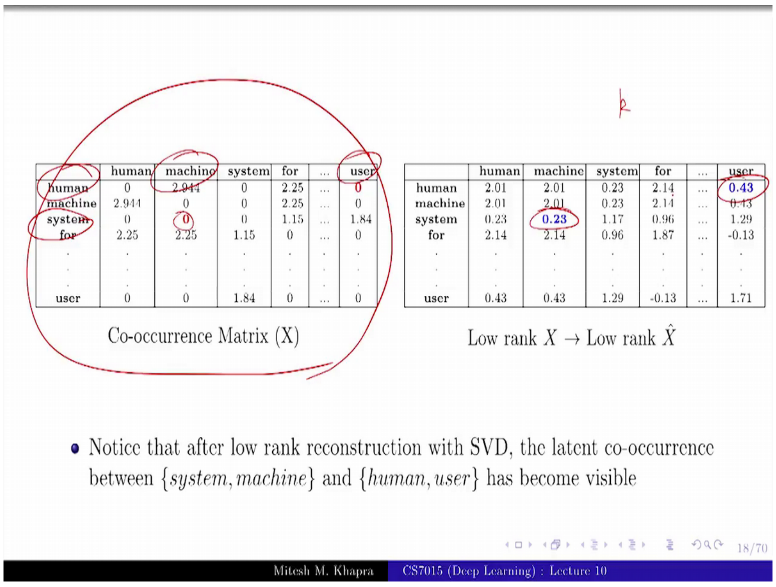
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, as an analogy consider this right suppose you are given 8 bits to represent colors. And this is how you represent very light green, light green, dark green and very dark green this is what your representation is.

In this original 8-bit representation, there is some similarity between the colors, but it is still a bit latent, but now if I were to ask you to use only 4 bits to represent these colors, what would you do? The lowest significant bits; if you use the first 4 no then use only get very light that is not the essence of that color right, you need the color to be there. So, if you compress what would happen is. So, that is what happens in when you go from 256 bit colors to higher or lower, right? The distinctions between the colors go off.

So, all of them would be compressed to green well that is the most important, important information in terms of the color right, because you need to be able to distinguish between green and red, as suppose to very dark and very light that is the more important information that is there right. So, when you compress it the most important information in that entity should be retained. And that is exactly what SVD does when it does a compression, it retains the most important information in the corresponding entries is that clear is the Intel intuition clear fine.

(Refer Slide Time: 05:32)



	human	machine	system	for	...	user
human	0	2.944	0	2.25	...	0
machine	2.944	0	0	2.25	...	0
system	0	0	0	1.15	...	1.84
for	2.25	2.25	1.15	0	...	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0	0	1.84	0	...	0

Co-occurrence Matrix (X)

	human	machine	system	for	...	user
human	2.01	2.01	0.23	2.14	...	0.43
machine	2.01	2.01	0.23	2.14	...	0.43
system	0.23	0.23	1.17	0.96	...	1.29
for	2.14	2.14	0.96	1.87	...	-0.13
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0.43	0.43	1.29	-0.13	...	1.71

Low rank $X \rightarrow$ Low rank $\hat{X}$

k

- Notice that after low rank reconstruction with SVD, the latent co-occurrence between $\{system, machine\}$ and $\{human, user\}$ has become visible

18/70

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

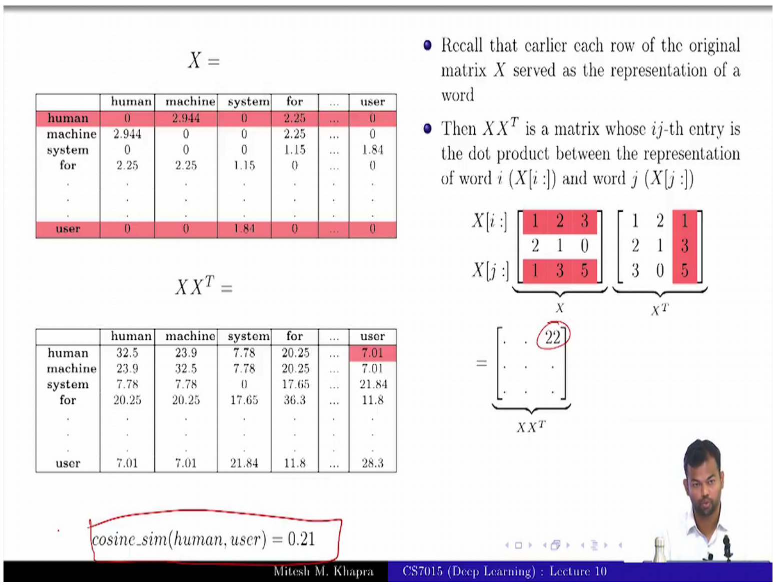
So, let us actually do this. So, this is my original co-occurrence matrix X, and I just repeat when I say, X I mean X PPMI. And now I have done SVD and I have done a low rank approximation of it. I do not know what was the value of k I selected, but some value of k it was definitely greater than 1 or 2. So now, you see a low rank approximation of X, what is the first obvious thing that you notice? It is dense now it is the longest sparse.

Now, can you tell me something about the colored entries, what was happening in the original matrix X? The word system and machine was never co-occurring because of which their value was 0. Same for human and user, but remember there is some important information in this matrix, which also tells you what are the words with user appears with and what are the words with human appears with, and that actually gives you intuition that these two words are actually related right same for system and machine

System and machine both would appear in the context of words like interface, install, run and so on. So, you know they are similar it just happens that these two words never appeared together. So, this similarity between them was latent or hidden in the original co-occurrence matrix. Now once I have done the SVD what has happened because I have forced it to compress the data, it has retained the most important in information and under that information these two words have actually come closer to each other right. So, you see that now you have a non-zero entry for the similarity between those two word pairs do you get the intuition and can you imagine that this would happen with SVD

And what is wrong in imagining you can, but I guess right that is what is happening with this. So, you think about PCA you think about SVD you think about auto encoders all the intuitions that we had build there the same is being applied here, right? All if you get this fine yeah. After SVD you could have right that is not necessary that it should be positive in the original matrix you do not have negative entries.

(Refer Slide Time: 07:31)



$X =$

	human	machine	system	for	...	user
human	0	2.944	0	2.25	...	0
machine	2.944	0	0	2.25	...	0
system	0	0	0	1.15	...	1.84
for	2.25	2.25	1.15	0	...	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0	0	1.84	0	...	0

$XX^T =$

	human	machine	system	for	...	user
human	32.5	23.9	7.78	20.25	...	7.01
machine	23.9	32.5	7.78	20.25	...	7.01
system	7.78	7.78	0	17.65	...	21.84
for	20.25	20.25	17.65	36.3	...	11.8
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	7.01	7.01	21.84	11.8	...	28.3

$\boxed{cosine\_sim(human, user) = 0.21}$

- Recall that earlier each row of the original matrix X served as the representation of a word
- Then XX^T is a matrix whose ij-th entry is the dot product between the representation of word i ($X[i:]$) and word j ($X[j:]$)

$$\begin{array}{c} X[i:] \\ \\ X[j:] \end{array} \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 1 & 3 & 5 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 3 & 0 & 5 \end{bmatrix}}_{X^T}$$

$$= \underbrace{\begin{bmatrix} . & . & \boxed{22} \\ . & . & . \\ . & . & . \end{bmatrix}}_{XX^T}$$

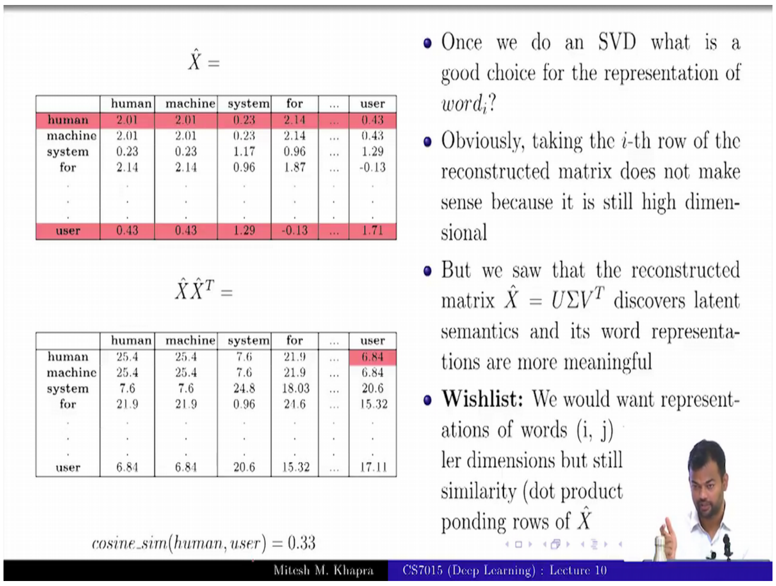
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

Now, here is a question right recall that earlier each row of the original matrix X served as the representation of a word. This was my original X PPMI not the rank approximation now in that case what would X X transpose give me? What would the ij th entry of X X transpose be. So, let us look at this toy example you have this X matrix you have xi and xj now I take X transpose.

Now, this is Xi this is Xj just standing. Now what would be the ij th entry of X X transpose, it will just be the dot product between these two right; is that fine? So, this is just the dot product between them and we know that dot product is more or less the same as cosine similarity module over the normalization right, you just need to normalize it by the norms of X and Xi and Xj in this case right?

So, I will just assume that this is a substitute for the cosine similarity. So, every entry at every ij th cell in X X transpose is the cosine similarity between the representations of the ith word and the g th word, is that clear to everyone? Ok fine. And in the original case which was the XPPMI the cosine similarity between human and user was 0.21.

(Refer Slide Time: 08:51)



$\hat{X} =$

	human	machine	system	for	...	user
human	2.01	2.01	0.23	2.14	...	0.43
machine	2.01	2.01	0.23	2.14	...	0.43
system	0.23	0.23	1.17	0.96	...	1.29
for	2.14	2.14	0.96	1.87	...	-0.13
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0.43	0.43	1.29	-0.13	...	1.71

$\hat{X}\hat{X}^T =$

	human	machine	system	for	...	user
human	25.4	25.4	7.6	21.9	...	6.84
machine	25.4	25.4	7.6	21.9	...	6.84
system	7.6	7.6	24.8	18.03	...	20.6
for	21.9	21.9	0.96	21.6	...	15.32
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	6.84	6.84	20.6	15.32	...	17.11

$cosine\_sim(human, user) = 0.33$

- Once we do an SVD what is a good choice for the representation of $word_i$?
- Obviously, taking the i-th row of the reconstructed matrix does not make sense because it is still high dimensional
- But we saw that the reconstructed matrix $\hat{X} = U\Sigma V^T$ discovers latent semantics and its word representations are more meaningful
- **Wishlist:** We would want representations of words (i, j) ler dimensions but still similarity (dot product ponding rows of $\hat{X}$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

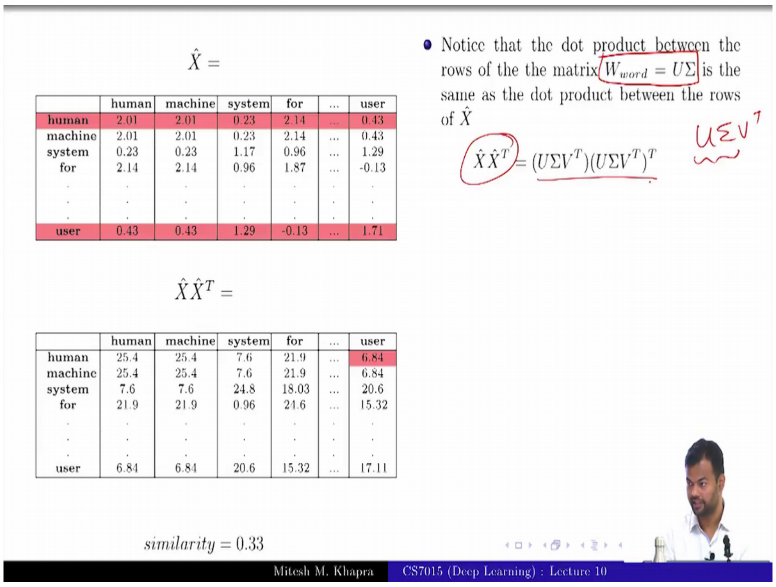
Now, once we do in SVD what is a good choice for the representation of the word i. After SVD what is the dimension of X hat? It is again n cross m because it is a sum of m cross n matrix. So, that the dimension of X hat is m cross n. Although it has been constructed using fewer information, but the dimension is m cross n right; that means, what is the size of the representation of every word? Still high dimensional still the same n or v, whatever everyone gets that is there any confusion with that.

Now you could say that, I will just take the ith row of the reconstructed matrix and use that as the representation. Because I know that now this representation is better some of those 0 entries have changed they have captured the latent semantics between the words. So, this is definitely better none is denying that that this compression has given us better representation because we are only keeping the most important information.

Now, if I do X hat X hat transpose, remember X hat is the reconstructed matrix. Then again by the same argument the ig th cell actually gives me the cosine similarity between the I th word and the g th word. And you can see that now the cosine similarity between human and user has actually increased. So, this is just for me to convince you that we have learned more meaningful representations. So now, what do we choose as the representation? I have still while computing this cosine similarity I have still used Xi which is high dimensional which has the entire vocabulary as the number of columns as a representation right.

So, there are 2 things coming out of here one is I really like this cosine similarity I see that it has improved; that means, the representations were computing something meaningful, but on the flip side I am still not happy because the representations are still high dimensional. So, can you construct a wish list for me based on this. I would want the same cosine similarity to be present as given by X hat, X hat transpose right, but I would like to represent it by fewer dimensions, that is exactly what my wish list is.

(Refer Slide Time: 11:01)



$\hat{X} =$

	human	machine	system	for	...	user
human	2.01	2.01	0.23	2.14	...	0.43
machine	2.01	2.01	0.23	2.14	...	0.43
system	0.23	0.23	1.17	0.96	...	1.29
for	2.14	2.14	0.96	1.87	...	-0.13
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0.43	0.43	1.29	-0.13	...	1.71

$\hat{X}\hat{X}^T =$

	human	machine	system	for	...	user
human	25.4	25.4	7.6	21.9	...	6.84
machine	25.4	25.4	7.6	21.9	...	6.84
system	7.6	7.6	24.8	18.03	...	20.6
for	21.9	21.9	0.96	21.6	...	15.32
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	6.84	6.84	20.6	15.32	...	17.11

$similarity = 0.33$

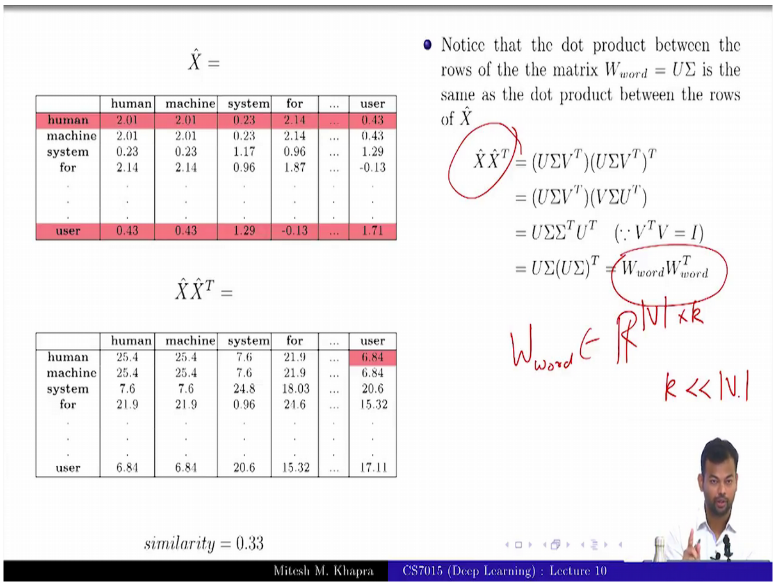
- Notice that the dot product between the rows of the the matrix $\boxed{W_{word} = U\Sigma}$ is the same as the dot product between the rows of $\hat{X}$

$$\hat{X}\hat{X}^T = \underline{(U\Sigma V^T)(U\Sigma V^T)^T} \qquad \underset{\sim}{U\Sigma} V^T$$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, let us see how do we do that now for no reason I am going to construct a matrix W word equal to U sigma what is u sigma it is the part of the SVD right the SVD told us it was U sigma V transpose. So, I am just considering this matrix I am going to call it W for no particular reason. Now, let me take X hat X hat transpose, I can write it as this is that fine, now what is the next step?

(Refer Slide Time: 11:32)



$\hat{X} =$

	human	machine	system	for	...	user
human	2.01	2.01	0.23	2.14	...	0.43
machine	2.01	2.01	0.23	2.14	...	0.43
system	0.23	0.23	1.17	0.96	...	1.29
for	2.14	2.14	0.96	1.87	...	-0.13
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0.43	0.43	1.29	-0.13	...	1.71

$\hat{X}\hat{X}^T =$

	human	machine	system	for	...	user
human	25.4	25.4	7.6	21.9	...	6.84
machine	25.4	25.4	7.6	21.9	...	6.84
system	7.6	7.6	24.8	18.03	...	20.6
for	21.9	21.9	0.96	21.6	...	15.32
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	6.84	6.84	20.6	15.32	...	17.11

$similarity = 0.33$

- Notice that the dot product between the rows of the the matrix $W_{word} = U\Sigma$ is the same as the dot product between the rows of $\hat{X}$

$$\hat{X}\hat{X}^T = (U\Sigma V^T)(U\Sigma V^T)^T$$
$$= (U\Sigma V^T)(V\Sigma U^T)$$
$$= U\Sigma\Sigma^T U^T \quad (\because V^T V = I)$$
$$= U\Sigma(U\Sigma)^T = W_{word}W_{word}^T$$

$$W_{word} \in \mathbb{R}^{|V| \times k}$$

$$k << |V|$$

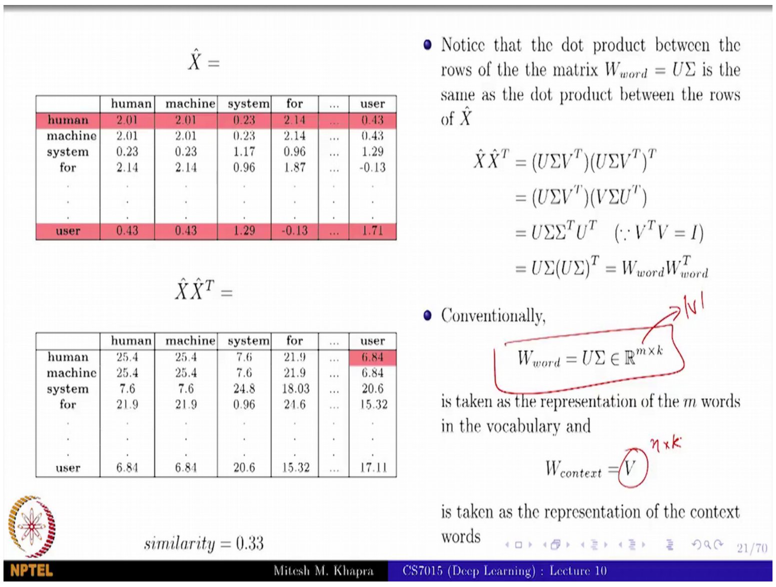
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

What does this mean? I want an answer right, this is that aha moment should be there or otherwise there is no point. What is how many rows are there in W? The same as the number of words in our vocabulary; what is the dimension of each row? K. So now, W word has low dimensional representations for the words in the vocabulary, but while doing this what have we not sacrificed the cosine similarity. The cosine similarity obtained by this is actually the same as this, do you get that; how many if you see this is very, very important? That if you have not understood, this everything is meaningless.

So, you see how from SVD we got a low rank or a low dimensional representation for the words right, W word is just to be clear k and k is very, very less than V right. So now, we have representations for words which are much smaller they are no longer V dimensional remember in practice this k would be of the order 100, 200, 300 and remember your vocabulary was of the order 50 k 1000 k and so on, right?

So, the huge reduction that you have got, and you have still been able to learn meaningful representations which give you better similarity between related words, right ?

(Refer Slide Time: 12:49)



$\hat{X} =$

	human	machine	system	for	...	user
human	2.01	2.01	0.23	2.14	...	0.43
machine	2.01	2.01	0.23	2.14	...	0.43
system	0.23	0.23	1.17	0.96	...	1.29
for	2.14	2.14	0.96	1.87	...	-0.13
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	0.43	0.43	1.29	-0.13	...	1.71

$\hat{X}\hat{X}^T =$

	human	machine	system	for	...	user
human	25.4	25.4	7.6	21.9	...	6.84
machine	25.4	25.4	7.6	21.9	...	6.84
system	7.6	7.6	24.8	18.03	...	20.6
for	21.9	21.9	0.96	21.6	...	15.32
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
user	6.84	6.84	20.6	15.32	...	17.11

$similarity = 0.33$

- Notice that the dot product between the rows of the the matrix $W_{word} = U\Sigma$ is the same as the dot product between the rows of $\hat{X}$

$$\begin{aligned} \hat{X}\hat{X}^T &= (U\Sigma V^T)(U\Sigma V^T)^T \\ &= (U\Sigma V^T)(V\Sigma U^T) \\ &= U\Sigma\Sigma^T U^T \quad (\because V^T V = I) \\ &= U\Sigma(U\Sigma)^T = W_{word}W_{word}^T \end{aligned}$$

- Conventionally,

$$\boxed{W_{word} = U\Sigma \in \mathbb{R}^{\overset{|V|}{m} \times k}}$$

is taken as the representation of the m words in the vocabulary and

$$W_{context} = \overset{n \times k}{\boxed{V}}$$

is taken as the representation of the context words

21/70

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, conventionally W word which is U sigma and belongs to m cross k. So, I am sorry for messing this up, but I have used m n and V are interchangeably. So, you would understand it from context that m is V. And the other matrix which is V is known as the W context matrix right what is the size of W context n cross k or k cross n, right?

That means it has the representations for all the context words and W word has a representation for all the target words right. So, we had these words on the rows and the context words on the column. So, W word has the representations for the rows and W context has the representation for the corpus.

So, this what we have seen so far, and this is where we learn today; is what a NLP was 6 years back right before the advent of deep learning. If you wanted to use word representations this is what you would do you would do con construct a co-occurrence matrix try these tricks of PMI PPMI positive negative 0 and all those things those heuristics. Then do a simple SVD retain the most important 100 200 dimensions and treat that as word representations and use it for whatever you want to do.

Now, what needs to be seen is, what happened with deep learning and how have this way of computing word representations changed over the past few years, right? So, that is what we are going to see in the next lecture right.