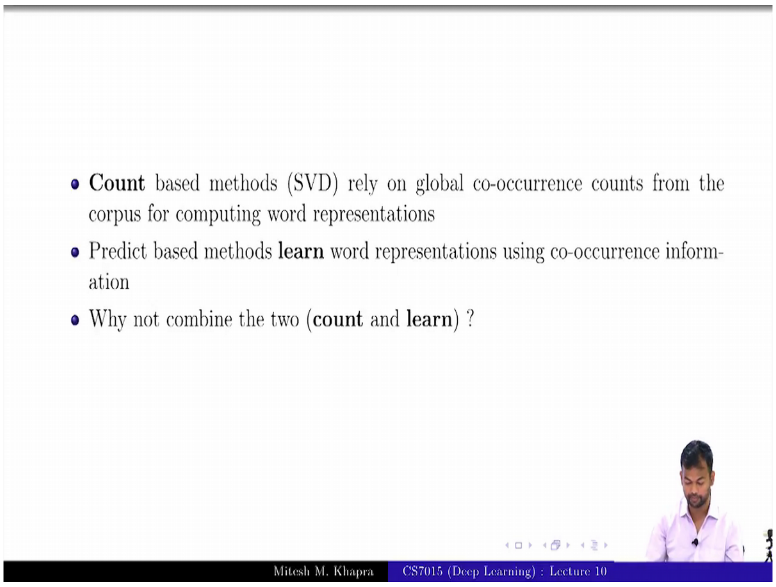
Deep Learning

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

Module – 10.8
Lecture – 10
GloVe representations

Ok. So, now, from here we will move on to yet another way of learning word representations, which is known as the GloVe representations.

(Refer Slide Time: 00:20)



- **Count** based methods (SVD) rely on global co-occurrence counts from the corpus for computing word representations
- Predict based methods **learn** word representations using co-occurrence information
- Why not combine the two (**count** and **learn**) ?

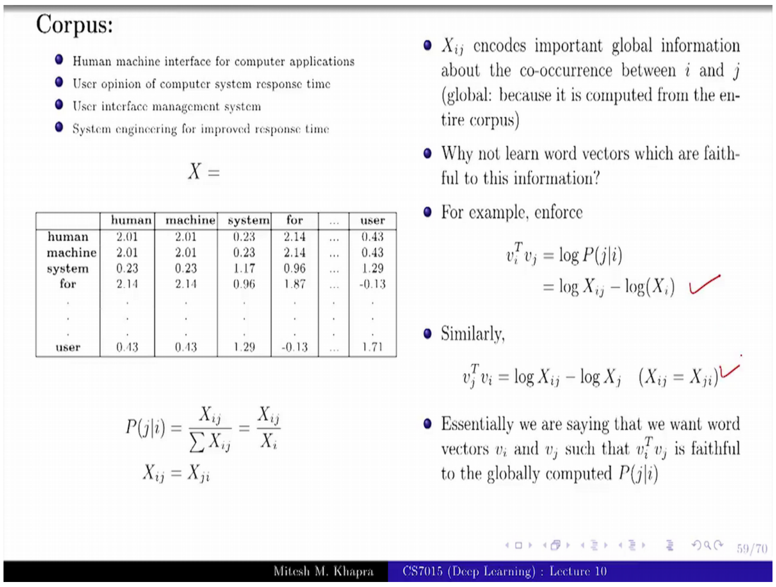
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

So, the count based methods rely on global co occurrence counts from the corpus for computing word representations that is what we saw in SVD. They look at these co occurrence counts and from there they build the word representations.

The predict based models, set up a learning problem where you have this feed forward net network and it tries to predict certain things from the given words and then you learn the parameters of that network, and you set up the task in such a way that the parameters actually correspond to word representations. So, this was the difference between count and predict based methods. Now what is the obvious next thing to do? Like hear the answer from a few of you, but I want to hear it from everyone.

What is the obvious next thing to do, you have count based methods you have predict based methods combine the two right. So, come up with some kind of a hybrid. So, that is exactly what GloVe does which is known as global vectors.

(Refer Slide Time: 01:10)



Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$X =$

	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

$$P(j|i) = \frac{X_{ij}}{\sum X_{ij}} = \frac{X_{ij}}{X_i}$$

$$X_{ij} = X_{ji}$$

- X_{ij} encodes important global information about the co-occurrence between i and j (global: because it is computed from the entire corpus)
- Why not learn word vectors which are faithful to this information?
- For example, enforce

$$v_i^T v_j = \log P(j|i)$$
$$= \log X_{ij} - \log(X_i) \checkmark$$

- Similarly,

$$v_j^T v_i = \log X_{ij} - \log X_j \quad (X_{ij} = X_{ji}) \checkmark$$

- Essentially we are saying that we want word vectors v_i and v_j such that $v_i^T v_j$ is faithful to the globally computed $P(j|i)$

59/70

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

So, I will go back to the co occurrence matrix. So, remember X ij, encodes the important global information about the word i and j and whether you replace it by pmi or ppmi or just keep the counts, it just gives you some information about how many times these two words actually appeared together.

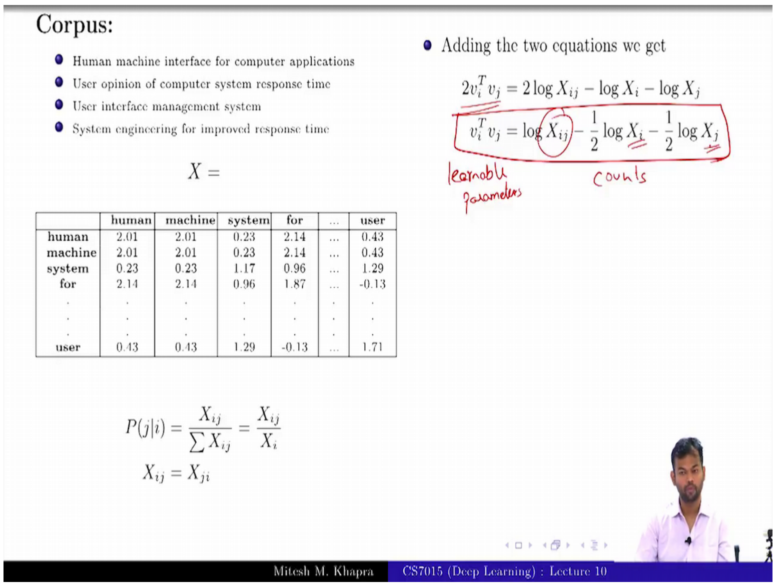
So, X ij encodes this global information and I call it global, because it is computed from the entire corpus fine. Why not learn word vectors which are faithful to this information. So, what do I mean by that? Suppose v i is the representation of the i th word and v j as a representative the j th word, which I want to learn I do not have these representations I wanted to learn. Now, this gives me the dot product between them, which gives me the similarity between them. Why not I set up my task in such a way that this similarity is actually proportional to this probability?

So, what does a similarly tell us, how well these two go together, what does p of j given I tell us, how likely j is given i right. So, does that make sense to have this analogy that, the dot product tells me the similarity the other notion of similarity is that how likely j is to appear in context of i which is given by p of j given i. So, why not set up my task such that whatever vector as I learn are actually faithful to this global similarity that I have computed from the entire corpus how many if you get this intuition? Okay.

How many if you see the difference between this and the predict based models? In the prediction based models you are operating at one word pair at a time; here you are looking at these global counts. We are trying to directly learn vectors which are faithful to your global similarity as given by your co occurrence counts. You get the merger between the two methods? You should not get it yet because we still have to do something or at least you get the intuition. Now what is p of j given i? It is actually this.

So, I can write it as this and similarly I can write the other guy v j transpose v i and that is going to be different because that is going to have p i given j instead of p j given i. So, I will have log X ij is fine, but instead of X i I will have X j here, how many of you get this? Fine.

(Refer Slide Time: 03:34)



Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$X =$

	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

$$P(j|i) = \frac{X_{ij}}{\sum X_{ij}} = \frac{X_{ij}}{X_i}$$

$$X_{ij} = X_{ji}$$

- Adding the two equations we get

$$\underline{2v_i^T v_j} = 2\log X_{ij} - \log X_i - \log X_j$$

$$\boxed{v_i^T v_j = \log \bigcirc\!\!\!\!X_{ij} - \frac{1}{2}\log \underline{\underline{X_i}} - \frac{1}{2}\log \underline{\underline{X_j}}}$$

learnable parameters

counts

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

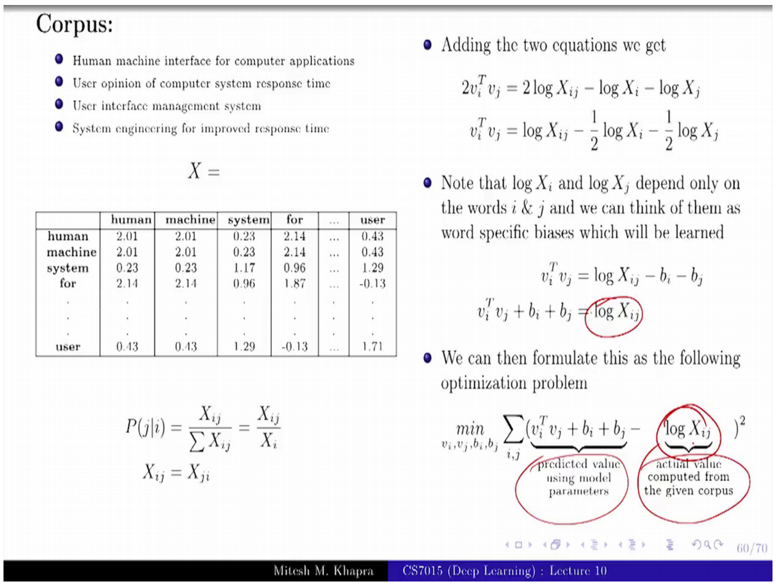
Now, if I add these two equations. So, I am going to add this equation and this equation. So, the left hand side I just get 2 times v i transpose v j, because v i transpose v j is the same as v j transpose v i and on the right hand side I get certain quantities. So, this is what I would actually want my word vectors to look at look like. I would want my word vectors to be such that, when I take their dot product they give me the quantity on the right hand side and this quantity has come based on counts learned from the corpus.

So, I have counts on the right hand side, and I have learnable parameters on the left hand side. So, you see how we are merging these two, but how do you learn this problem. Now it is ok to say, what I like what I have said now is that this is what I desire. I desire that my word vectors should be learned in such a way that they are faithful to the global counts through the following equation. This is what I desire. Desiring something is one thing, but now how do I set this up as a learning problem.

So, when I ask you what is the learning problem what do you need to think about? Objective function good that is a good start. So, what is the objective function for this, what are the parameters of the optimization you are optimizing with respect to what? The v i is and the v j's right all the word representations how many of those do you have? V each of size k. So, those are your parameters for optimization, now what is the loss function? If I give you the loss function it will look very very obvious, but I do not want to do that.

So, just continue thinking about that, while I will make some more simplifications to what we have here now, what is this count? This is the co occurrence count how many times these two occur together, what is this count? The number of times the word i appear. So, this depends only on i what is this count? The number of times the word g appears. So, to make the model more flexible; that means, give it some more freedom what I am going to do is instead of log x i and log x j I am going to introduce parameters b i and b j.

(Refer Slide Time: 05:32)



Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$X =$

	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

$$P(j|i) = \frac{X_{ij}}{\sum X_{ij}} = \frac{X_{ij}}{X_i}$$

$$X_{ij} = X_{ji}$$

- Adding the two equations we get

$$2v_i^T v_j = 2\log X_{ij} - \log X_i - \log X_j$$

$$v_i^T v_j = \log X_{ij} - \frac{1}{2}\log X_i - \frac{1}{2}\log X_j$$

- Note that $\log X_i$ and $\log X_j$ depend only on the words i & j and we can think of them as word specific biases which will be learned

$$v_i^T v_j = \log X_{ij} - b_i - b_j$$

$$v_i^T v_j + b_i + b_j = \log X_{ij}$$

- We can then formulate this as the following optimization problem

$$\min_{v_i, v_j, b_i, b_j} \sum_{i,j} (\underbrace{v_i^T v_j + b_i + b_j}_{\substack{\text{predicted value} \\ \text{using model} \\ \text{parameters}}} - \underbrace{\log X_{ij}}_{\substack{\text{actual value} \\ \text{computed from} \\ \text{the given corpus}}})^2$$

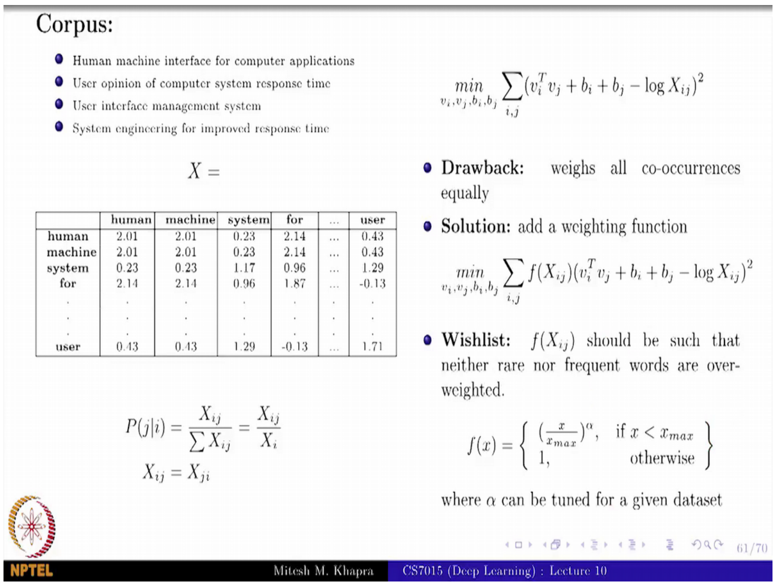
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 10

60/70

I am saying that these parameters can also be learned. So, effectively using all these three I should be able to get this, this is what I desire now set up the loss function. Using these two things come on that should not be so hard, what is this? This is what you are trying to predict; what is this? This is what you know is true because you have computed from the corpus; now can you come say the loss function the difference between these two right. So, you could have this as the loss function, this is the predicted value using models parameters, this is the actual value computed from the corpus.

So, think of this that you are trying to learn the parameters in such a way, that you end up predicting this and if you predicted this, you know you have done the right thing and this you know already because you have computed it from the corpus. So, this is the true value and this is the predicted value. So, as in any loss function predicted minus true the whole square, does that make sense how many if you are fine with this?

(Refer Slide Time: 06:44)



Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$X =$

	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

$$P(j|i) = \frac{X_{ij}}{\sum X_{ij}} = \frac{X_{ij}}{X_i}$$

$$X_{ij} = X_{ji}$$

$$\min_{v_i, v_j, b_i, b_j} \sum_{i,j} (v_i^T v_j + b_i + b_j - \log X_{ij})^2$$

- **Drawback:** weighs all co-occurrences equally
- **Solution:** add a weighting function

$$\min_{v_i, v_j, b_i, b_j} \sum_{i,j} f(X_{ij})(v_i^T v_j + b_i + b_j - \log X_{ij})^2$$

- **Wishlist:** $f(X_{ij})$ should be such that neither rare nor frequent words are over-weighted.

$$f(x) = \left\{ \begin{array}{ll} (\frac{x}{x_{max}})^{\alpha}, & \text{if } x < x_{max} \\ 1, & \text{otherwise} \end{array} \right\}$$

where α can be tuned for a given dataset

61/70

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

So, now how will you train in this network? Gradient right. So, I will use gradient set and you will get these parameters.

So, there is a bit more on this which I will not cover actually. So, I will just skip this slide, you can go back and take a look at. It is a some slight modifications to this yes. So, again the same idea that cat will go close to all the. So, here again you will have the v i and the uc s right. So, you will have cat will come close to all the words that it co occurs with, feline will also come close to the same words. So, maybe I have not used to right notation here if you need to change it again. So, we should have v i s and u j's right. So, again you have one word matrix, word representations and the other is the context representation then it is fine right that is the problem here how many if you get that right? Again we have to have these two things let us change that a bit.