

Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 06
Lecture - 02

So, since we have understood why link prediction you know is a difficult problem, we also have formally defined what is link prediction. Let us try to you know let us try to understand how to evaluate a link prediction method ok.

(Refer Slide Time: 00:39)

Evaluating Link Prediction Methods: Train-Test Split

Case I (task of inferring missing links)

- Only a single snapshot $G_{t_1}(V, E)$ of the network at timestamp $t = t_1$
- Split E into disjoint sets E_{train} and E_{test}
- To obtain test set, delete edges from E and add them to E_{test}
- Deletion strategies:
 - Uniformly at random
 - Based on the degrees of their endpoints
 - Based on the degrees of their endpoints

Case II (task of predicting future links)

- At least two snapshots of the network: $G_{t_1}(V, E)$ at time $t = t_1$ and $G_{t_2}(V, E')$ at time $t = t_2 (> t_1)$
- Set $E_{train} = E$ and $E_{test} = E' \setminus E$

The slide includes a diagram of a network graph with red edges being removed and added to a test set. It also shows the mathematical representation of the train and test sets for Case II: $E' - E$.

So, generally we will not talk about metric here, we will talk about the evaluation setting setup. So, in case of you know inferring missing links right, missing link prediction not future link prediction; missing link prediction what we do, we are given a static network ok this one. And what we do we split we try to split the edge set E into training set E_{train} and test set E_{test} ok.

So, out of say 100 edges, we divide it into two parts. We assume that ok 80 edges are the training sets and remaining 20 edges form the test set. And so, essentially what you need to do, you need to remove this 20 edges from the static graph right and your task could be to predict this 20 edges. And, this 20 edges are your E_{test} ok. Now, how do you choose this 80, 20 samples right?

Now, how do you delete, I mean how do you choose this E test samples right? Of course, you can randomly choose edges and delete them and your algorithm will try to predict those edges, random deletion is possible. Other possibility would be based on some degree, some notion.

Either in degree notion or whatever degree notion or homophile motion, some sort of motion. So, why it is important? Think, about the network like this ok and assume that ok. So, this is a network and let us say you delete this edge right and you want to predict this edge. This edge is very easy to be predicted. This is this edge is very easy to predict because, these two nodes this node and this node.

These two nodes are part of a cluster, a triangle. So, due to the transitivity property, it is highly likely that these two nodes will also be connected ok. This edge is easy to predict. What about this edge? If I remove this edge right, then the graph would become like this ok. So, how do we predict that this node and this node will be connected? It is very difficult because, these two nodes are far from each other ok and this kind of edge, these kind of edges are bridge right.


We discussed in the previous chapters right. So, when we remove bridge, a bridge edges kind of connect components which are far apart from each other. And, upon deletion of this bridge sometimes the graph the graph gets disconnected or the shortest part distance between pairs of nodes increase significantly ok. So, now, the question is how do you choose this E test right? So, often times I have seen people oftentimes cheat right.

So, what people do, people selectively choose the you know remove those edges which are easy to predict and then you know show that look our algorithm produces 90 percent accuracy, 95 percent accuracy and so on. But, the difficulty is to predict those edges which actually connect nodes which are far apart from each other otherwise ok.

So, how to you know divide E train and E test, this is this is important to consider. The second problem if it is future link prediction problem, then it is very straightforward. You do not need to delete anything, because you have two snapshots now at t_i and at t_j right. You have same set of vertices, but different set of edges. So, given this graph, can we predict this $E - E$ edges? Ok.

So, this is the task. Given this graph, can we predict which edges are going to come to form this network and this is future link prediction problem ok.


(Refer Slide Time: 05:09)



Evaluating Link Prediction Methods: Positive-Negative Samples

- Initial Network: $G_t(V, E)$ of the network at timestamp $t = t_t$
- Set of all possible edges in the network: $U = 2^E$
- Obtain E_{train} and E_{test} following either of the cases mentioned in earlier
- Set of edges not formed till timestamp $t = t_t$: $L = U \setminus E_{train}$
- Convert the problem of link prediction into a binary classification problem
 - Edges in E_{test} form the positive samples
 - Edges in set $L \setminus E_{test}$ form the negative samples
- Positive samples are expected to have higher probability than negative samples

Handwritten notes: $|U| = \binom{V}{2} = \frac{V(V-1)}{2}$



So, I have already discussed you know the issues of positive and negative samples, but let me again reiterate. So, when we have V number of nodes, you have $\binom{V}{2}$ number of pairs ok. And, and say out of this, out of this only you have $|E|$ number of edges which are already there. So, $\binom{V}{2} - |E|$ number of node pairs are connected ok.

These many number of node pairs are not connected ok. So, your task would be to predict those nodes, those pairs of nodes which are going to be connected in the future. So, this among these many number of pairs, you need to predict how many of them will be connected. And, I mentioned in the last lecture that this is really small, say 0.5 percent of the node pairs will be connected. You know sometimes 1 or 2 right.

So, in terms of classification problem, if you think of link prediction problem as a class binary classification problem, you have positive samples and negative samples. So, what are the positive samples? Positive samples are those node pairs which are already connected through this $|E|$ number of edges ok. And, negative samples are those point, those pairs which are not connected at current time stamp ok.

So, you have huge negative samples. Its a massive imbalance classification problem ok. So, how do we deal with this problem? So, you can actually do some sort of under sampling, you

sample equal number of you sample equal number of I mean you basically sample from your negative samples right, under sample the negative sample population.

So, that the number of positive edges and number of number of positive samples and number of negative samples will become equal ok; that can be one possibility. Other possibility is that since this is inherently an imbalance classification problem, let us make it imbalance ok. But, not this much of imbalance. Let us make it 1 is to 10 or 1 is to 5. So, 1 is to 5 positive is to negative samples and then we solve it ok.

(Refer Slide Time: 07:42)

Evaluating Link Prediction Methods: Confusion Matrix

		Actual →	
		Link formed	Link not formed
Predicted ↓	Link formed	True Positive (TP)	False Positive (FP)
	Link not formed	False Negative (FN)	True Negative (TN)

True Positive (TP): the count of how many times the model predicted a link to be formed, and it actually forms
 True Negative (TN): the count of how many times the model predicts that a link will not form, and it actually does not form
 False Positive (FP): the count of how many times the model predicts a link to be formed; however, it actually does not form
 False Negative (FN): the count of how many times the model predicts a link will not form; however, it actually forms (opposite case of FP)

So, since this is a classification problem as I mentioned right, the problem here is I mean not the problem, I mean how do we evaluate a classification problem. If you if you are familiar with a binary kind of classification problem, we generally draw this confusion matrix. Confusion matrix is a way to you know understand the false positive, true positive results right.

If you are not familiar with this, I will briefly explain here. So, in a confusion matrix right binary classification problem your rows indicate actual predictions and your columns indicate sorry the rows indicate actual levels and columns indicate the predicted levels ok.

Now, what do you mean by levels in case of link prediction? So, your one instance is a pair of nodes and your level would be either 1 or 0, 1 means this pair of nodes will be connected, 0

means this pair of nodes will be will not be connected ok. So, this is 1 0 actual; so, this is actual and this is predicted 1 or 0 ok. Now, look at this cell, this cell say this is 25 ok.

It means that there are 25 such instances which are actually positive and your prediction algorithm also says that they are positive. So, these are true positive. Your algorithm says that this is positive and that is true. So, these samples are called true positive ok. Now, let us look at this quartile ok. So, this means so, say there are 10 such instances. So, it means there are 10 instances which are predicted as 0, but their actual level levels are 1.

So, you are you so, you basically say that it is negative, it is 0. It is negative, but this is false ok. It is negative, but it is actually positive. So, you say that this is negative, but this is false. It is called false negative. Look at this part ok. So, this cells indicates that your prediction is 1, you predicted it as positive, but your actual level is negative.

So, this is so, you predicted it as positive, but that is false. So, it is called false positive right. Now, what is this one? You predicted it as negative and it is actually negative. So, this is true negative. You assigned it as negative and is actually true ok. So, you may wonder that ok you mentioned; so, I mentioned this is a true positive, false positive, false negative, false positive and true negative.

But, in this slide it is true positive, this is false positive right. Why it is so? Because, in this matrix rows indicate predicted outputs and columns indicate actual outputs. In our case, it is basically the other way around. Rows indicates actual output and columns indicate predicted output. Therefore, this mismatch ok. But, if you understand the nomenclature positive that is true, positive your so, this positive negative.

These are with respect to your prediction and you judge whether it is correct or not based on the ground truth right. And, then you decide whether it is a true prediction or false prediction ok.

(Refer Slide Time: 11:55)

Evaluating Link Prediction Methods: Confusion Matrix

NPTEL

✓ Accuracy (ACC): ratio of the total number of correct predictions to the total number of predictions

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

✓ Precision (P): out of all the links that are predicted by the model as positive, how many does actually belong to the positive samples

$$P = \frac{TP}{TP + FP}$$

✓ Recall (R): out of all the links that are actually positive, how many are predicted as positive by the model

$$R = \frac{TP}{TP + FN}$$

The slide includes handwritten red annotations: a checkmark next to the Accuracy definition, a checkmark next to the Precision definition, and a checkmark next to the Recall definition. The formulas for Precision and Recall have their numerators (TP) circled in red. A small diagram of a triangle is also present.

So, you have this 4 quantities now. So, based on these 4 quantities, one can come up with multiple such metrics to evaluate a system. So, first metric is called accuracy. So, accuracy is very simple. Accuracy basically says that looks at all correct predictions, correct predictions and you divide it by the total samples right. So, how many correct predictions are there?

True positives are correct predictions and true negatives are also correct predictions. So, these diagonal elements are correct predictions. So, accuracy is this true positive plus true negative divided by total; true positive false positive, true negative false negative. And, total basically a TP FP T and FN, if you sum them up this is basically the number of instances N right.

This is accuracy. Why accuracy is not a good measure, I will discuss later. The other metric is called precision which is basically saying that out of all the predictions right, how many of them are positive, how out of all the predictions how many of them are how many how many of them are correct? Out of all the positive predictions, let me put it in this way, out of all the positive predictions how many of them are correct?

So, how many positive predictions are there by your model? Number of true positives and number of false positives because false positive instances are those which have been predicted as positive, but they are false. So, how many positive samples are predicted? TP and FP and out of them how many are correct? TP. So, it is TP by TP plus FP as a precision. What is recalled?

Out of all the positive instances irrespective of your models prediction, out of actual positive instances how many of them have been predicted correct, correctly? How many positive instances are there? True positive plus false negative, because false negative instances are those which have been predicted as negative, but that prediction is wrong; means they are actually positive.

So, TP plus FN are total number of positive samples, actual positive samples right. And, how many of them are correct? TP. So, this is your recall, precision recall numerator is same, but the denominators are different ok.

(Refer Slide Time: 14:34)

Evaluating Link Prediction Methods: Confusion Matrix

NPTEL

True Negative Rate (TNR)/specificity: Out of all the links that are actually negative, how many are predicted by the model to be negative

$$TNR = \frac{TN}{TN + FP}$$

False Positive Rate (FPR)/false alarm ratio/fallout rate: Out of all the negative samples, how many are wrongly predicted to belong to positive class instead

$$FPR = \frac{FP}{FP + TN}$$

Handwritten note: $FPR = 1 - TNR$

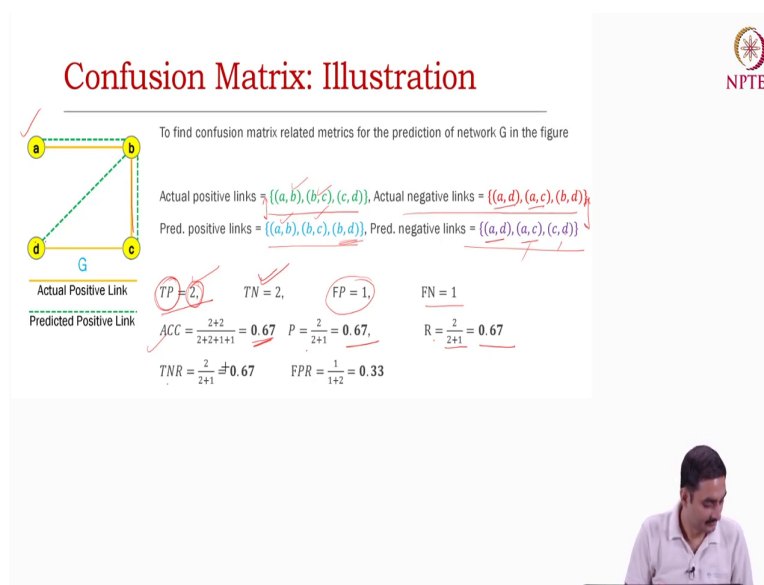
Now, let us try to understand a bit difficult matrix now. True negative rate TN TNR, this is also called specificity, specificity. I will use this term later. So, true negative rate is basically out of all the links, out of all the samples that are actually negative, how many are predicted by the models to be negative, true negative rate?

So, how many negative instances are there actually? True negative plus false positive, false positive instance are those which have been predicted as positive, but that is false. They are actually negative. So, TN plus FP and out of them, how many of them are actually predicted as negative? TN, this is true negative rate. This is something called false positive rate ok or false alarm rate or fall out rate ok; remember this terminologies.

So, false positive rate is out of all the negative samples right, how many are wrongly predicted? Out of all the negative samples right, number of negative samples; we have already discussed TN plus FP right. How many of them are wrongly predicted to belong to the positive class? FP. So, this FP number of instances have been predicted as positive, but that is wrong.

So, FP by FP plus TN. This is called false positive rate. You see here false positive rate is 1 minus true negative rate, is not it? Ok. You can easily see this.

(Refer Slide Time: 16:26)



Now let us take an example. Say this is your network ok and the yellow lines are actual edges and dotted lines are edges which are predicted ok. So, how many actual edges are there? Actual edges are a, b b, c and c, d right. So, these are positive samples, actual positive samples. How many actual negative samples are there? So, those pairs which are not connected a, d b, d a, d b, d and a, c.

These are negative samples. Now, let us look at the prediction. Our prediction says that the positive links are this dotted lines a, b b, d and b, c. These are predicted positive links and predicted negative links are b, c a, d and a, c sorry d, c not b, c d, c d, c a, c and a, d ok. So, what is true, what is true positive? If you look at here carefully true positive is; so, let us compare these two sets ok. So, true positive would be which one? a, b a, b and b, c b, c. So, true positive is 2.

How many of them are false positive? Only 1, this one. How many of them are true negative? Let us compare these two sets right, a, d a, d a, c a, c right 2. So, 2 of them are true negative. And, how many of them are false negative? 1. So, we got the confusion matrix now. So, the accuracy would be this plus true negative this by total.

Precision would be true positive by true positive plus false positive 0.67. Recall would be true positive by true positive plus false negative 0.67. True negative rate you can measure and false positive rate also you can measure ok.

(Refer Slide Time: 19:12)

The slide features the NPTEL logo in the top right corner. The main title is "Evaluating Link Prediction Methods: Mere Accuracy Is Not Enough". Below the title, there are several bullet points:

- A typical example network that consists of
 - 9 actual negative edges
 - 1 actual positive edges
- A typical prediction model:
 - the model predicted 10 negative edges in the network
- How Good is the Prediction Model???

 A hand-drawn calculation in red ink shows:

$$\frac{9N + 1P}{9 + 0} = \frac{10N}{10}$$
 Below the calculation, the slide lists:

- $TP = 0, TN = 9, FN = 1, FP = 0$
- $ACC = 90\%$
- However, the model failed at its desired task of predicting a rare positive edge in the network!!!

 A small video feed of a presenter is visible in the bottom right corner of the slide area.

So, now let us try to understand why accuracy is not you know not a good measure. So, what is accuracy? Accuracy says true positive plus true negative by all right. Now, let us think of an imbalance classification problem. Because, link prediction is ultimately a link classification problem right. So, you have 9 actual negative edges and 1 actual positive edges in the ground truth.

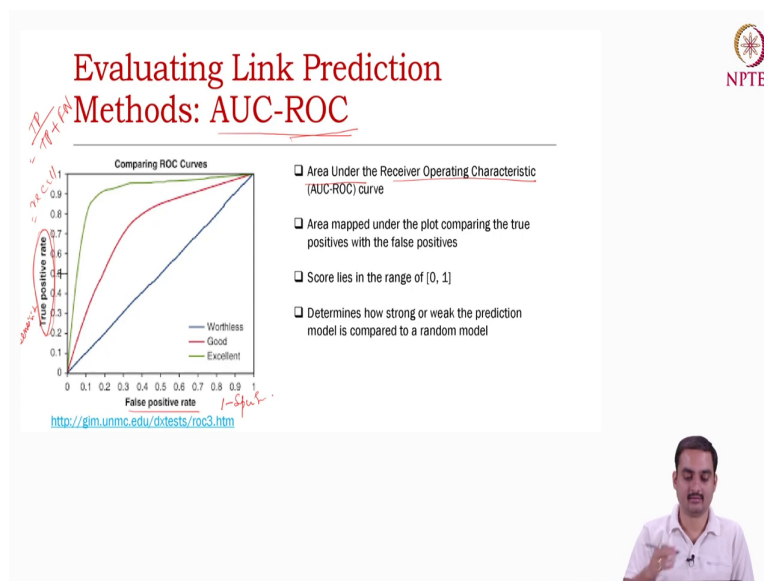
So, if you blindly say that look all the pairs are negative edges, all the pairs are negative right, your algorithm says that all the pairs are negative. There are 10 pairs and all the pairs are negative right. So, then what would the; what would be the accuracy? So, there are 9 negative edge and 1 positive edge and your method says that all of them are negative.

So, how many true how many true negative cases are there? There are 9 true negative cases, because you know method says all 10s are negative, but out of them 9 are actually negative.

So, true negative is 9 ok, but true positive is 0 ok and total is 10. So, the accuracy is 9 by 10, 0.9, 90 percent.

Think about it. You can blindly say that everything is negative and still you get 90 percent accuracy right. You are unable to predict the you know the smaller class right. You are unable to predict it, but you still got 90 percent accuracy, that is the problem of the accuracy measure ok.

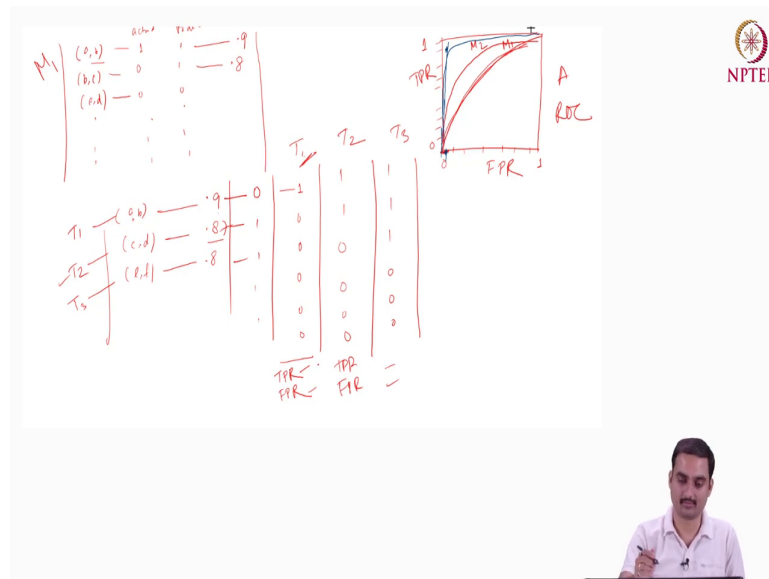
(Refer Slide Time: 21:11)



So, how do we get rid of this problem? There are many ways. For example, you can do class wise precision, say precision for positive class precision for negative class, recall class wise recall and so on and so forth. And, use your target would be to look at the rare class which is in our case this is a positive class ok. We look at the rear class and with respect to the rare class, you need to predict the you need to measure the precision recall. But, there are even more sophisticated measures right.

So, this measure is often used in information retrieval (Refer Time: 21:52) classification as well. This is called AUC-ROC. So, ROC the full form of ROC is Receiver Operating Characteristic curve, receiver operating characteristic curve. Why the why is the name? I do not know ok and area and then what we will do in this. So, we plot this ROC curve and we look at the area under this ROC curve ok. And, the area would give you the measurement, the accuracy. Now, let me discuss what is this AUC-ROC metric ok.

(Refer Slide Time: 22:42)



So, let me you know let us take an example. So, let us say link prediction problem again and you have different pairs say a, b b, c right c, d ok and you know their levels, actual levels. Say this is 1 0 0 whatever right and you also know their prediction. So, this is actual right and your prediction right. Your prediction is say 1 1 0 something like this ok.

So, whenever you predict, whenever you run any classifier, the classifier also outputs a probability right, a confidence; that ok the model says that with probability 0.9, the model says that this is a positive instance. With probability say point say what about 0.8, you say that this is a negative instance and so on and so forth.

So, we will discuss later in some link prediction algorithm that this probabilities are also writtended ok. So, given this thing how do we how do we draw the AUC-ROC curve? Ok. So, what you do first? We first you know rearrange all these instances, sort all the instances based on this probability right. Say a, b will come first, its probably is 0.9, then say c, d would 0.87 right, e, f 0.8 and so on and so forth.

We short it based on the probability obtained from the classifier ok. And, then what you do and we also know their actual labels right, say the actual label is 0 1 1 and so on and so forth ok. Now, let us assume that you know this sample is T 1 or say whatever T 1, this is T 2 this is T 3 and so on ok. So, let us only look at T 1. So, this is T 1 and confidence is 0.9.

So, I will assign all the instances as positive whose confidence value, whose corresponding confidence value is greater than equals to 0.9 right. So, I will I will say that ok this is positive because, my threshold is 0.9. So, whatever comes above 0.9, on and above 0.9, everything will be classified assigned as 1. So, this is 1, all the others are 0.

So, with respect to the first instance T 1 we get a prediction. So, with respect to the confidence level of the first instance T 1, we get a prediction of all the other instances right. And, we can if we get this, we can draw this confusion matrix right because we know the actual levels. We will get and we can measure true positive rate and false positive rate right. Remember, the AUC-ROC curve you see here in the previous is basically a curve, where x axis is a false positive rate.

False positive rate is something that we discussed already right. False positive rate right and y axis is true positive rate. What is true positive rate? True positive rate is same as recall right. It is basically you know true positive by true positive plus false negative ok. This is also called this is also called sensitivity. And, what is false positive rate? We have already seen false positive rate is 1 minus true negative rate and true negative rate is called specificity.

So, false positive rate is 1 minus specificity ok. So, we basically draw a curve between curve a curve where x axis is ones 1 minus specificity and y axis is sensitivity. These are just names ok. Recall, y axis and x axis is false positive rate. Now, for the first instance, you can measure t m true positive rate and false positive rate, you get some value. Let us say you get some value ok. Now, you take T 2, this one and you know that the same thing. So, the threshold is 0.87.

So, on and above the threshold, everything would be 1, others would be 0. So, true positive rate and false positive rate you can measure, T 3 similarly you can measure ok. So, these are all pairs and you have false positive rate and true positive rate. You can plot it right. Let us say the plot looks like this and remember this always ranges between 0 to 1, this also ranges between 0 to 1 ok.

So, if you look at this that the square, the total area is 1 ok. So, for a particular model; so, this is for a particular model M1, you will get a curve like this. Now, think about the curve right. This will never decrease. Why this will; why this will never decrease? For with the increase of false positive rate right, you keep adding false positives. If you keep adding false positives right, let us again go back you will understand.

Go back to the formula, you keep adding false positives, your recall will never change right. So, with the increase of false positive; say for example, you have drawn this curve and you are; and you are planning to draw the remaining shape of the curve. You are at this this position, you have certain false positive value right. This is a false positive. You increase false positive right, the first positive will increase, but this will never decrease.

This will either remain same or increase ok. Recall, will never decrease. Then, what is recall? Whatever you have whatever positive samples you have return so far divided by out of all the positive samples ok. So, the curve will never go down ok. This will (Refer Time: 29:49) increase. And, what is the best possibility? Now, say for M1 for a model M1 you get this curve, for model M2 say you get another curve like this.

So, this is ROC curve ok and AUC is the area under this ROC curve. So, for M1, I actually look at I actually measure this area and we know how to measure this area. For M2, we measure this area and the idea is that higher this area better the model. Why? Think about it. What would be the best case? With little false positive, with little false positive, if I get the maximum true positive right; say I get.

So, with this false positive, I get a true positive like this; that is the best case ok. So, the more the curve actually you know move towards this boundary, beta would be the model. So, this is AUC-ROC and this is important particularly in case of imbalance classification problem ok.

(Refer Slide Time: 31:15)

Evaluating Link Prediction Methods: Precision-Recall Curve

- A plot where
 - the precision is along y-axis,
 - the recall is along x-axis
- Based on the positive samples only
 - more stable evaluation metric than a ROC curve
- Misclassification in the PR curve is limited to only the total predicted samples
- Random predictor (aka baseline curve) is shown as a horizontal line along the x-axis ($y = c$)

<https://towardsdatascience.com/gaining-an-influential-understanding-of-precision-and-recall-368e37304a7>


So, let us look at the last metric, that we use. This is called precision recall curve. Now, precision recall curve gives you the flexibility to choose a particular recall value and identify the corresponding precision right. Sometimes what happens is that you are told that look I want to understand the precision value at recall 0.3, I want to understand the precision value at recall 0.4 right.

So, this curve gives you the inter spectrum of it right. So, what is this curve? x axis is the recall and y axis is the precision ok. So, you return an item right, you return an item and you predict it to a you predict it whether it is positive or negative. So, let us say you have these items right; item 1 2 3 4. In our case items are pairs of nodes and you predict it as 1 0 1 0 and so on.

So, till this part you have a precision recall value, because you already know the actual value, actual level of this it is 0. Till this part you know the precision recall, till this part you have a precision recall, this part precision recall and so on and so forth right. So, as you increase your recall. So, eventually you will return all the positive samples.


So, eventually your recall will be 1 right, but precision will decrease significantly. So, this is precision recall curve. You can say that ok at 0.4, I know the precision, 0.6 I know the precision. Similarly, with point 0.7 precision right, 0.7 precision I know the recall ok.

(Refer Slide Time: 33:13)



Evaluating Link Prediction: Some Unique Problems

- Temporal dynamics of the network
 - temporal changes hard to obtain from the real-world data
 - difficult to fit in a binary-classification scenario
 - a complicated mixture of addition and deletion of nodes and edges; complicates the comparison of the network instances
- Directionality of the edges
 - confusion matrix and AUC scores to be calculated accordingly
- Sign and weight of links
 - All the edges are not equally important
- Class imbalance
 - Size of negative samples is often much larger than that of the positive samples



So, that is about the evaluation matrix. The unique problems of link prediction, we have already discussed. The first problem is that you know it is very difficult to collect temporal information right. So, this temporal change is hard to collect. Oftentimes, we look at a mixture of addition and deletion, but we generally look at when we talk about link prediction, we only focus on addition of edges.

We do not consider deletion of edges, that is also a problem. We generally do not explicitly consider the sign of an edge whether it is a positive or negative. We do not consider the weight of an edge, the direction of an edge right. Link direction prediction problem again is a separate problem, separate problem, separate problem statement right. And of course, the class imbalance problem itself is a major you know major setback here ok.

So, we stop here. The remaining part of this chapter, we will discuss more of an algorithms for link prediction ok.

Thank you.