Week1

* 1. Natural Language Content Analysis

Lexical analysis(词汇)

Syntactic analysis(语法)

Semantic analysis(语义)

Programtic analysis

Prepositional(介词) phrase attachment ambiguity

Anaphora(照应，省略)

Presupposition(前提)

Entity extraction

Relation recognition

Bag of words representation： ignore orders, sufficient for most search tasks

Query : environment

Feedback: additional word

Pull: search engine, user, ad hoc information

* Querying: knows keywords
* Browsing: navigate, explore

Push: recommender systems, system, stable

TR vs Database Retrieval

TR: empirically defined problem, rely on empirical evaluation

Database Retrieval: not strictly

Document selection

Document ranking: prioritization is needed, users help decide cutoff

Two assumptions:

Document independent=> not really: same document, one is read, the other is useless, or collective relevant(three docs are a whole)

Browse sequentially

1.4 Overview of Text Retrieval Methods

TF: term frequency

Week2

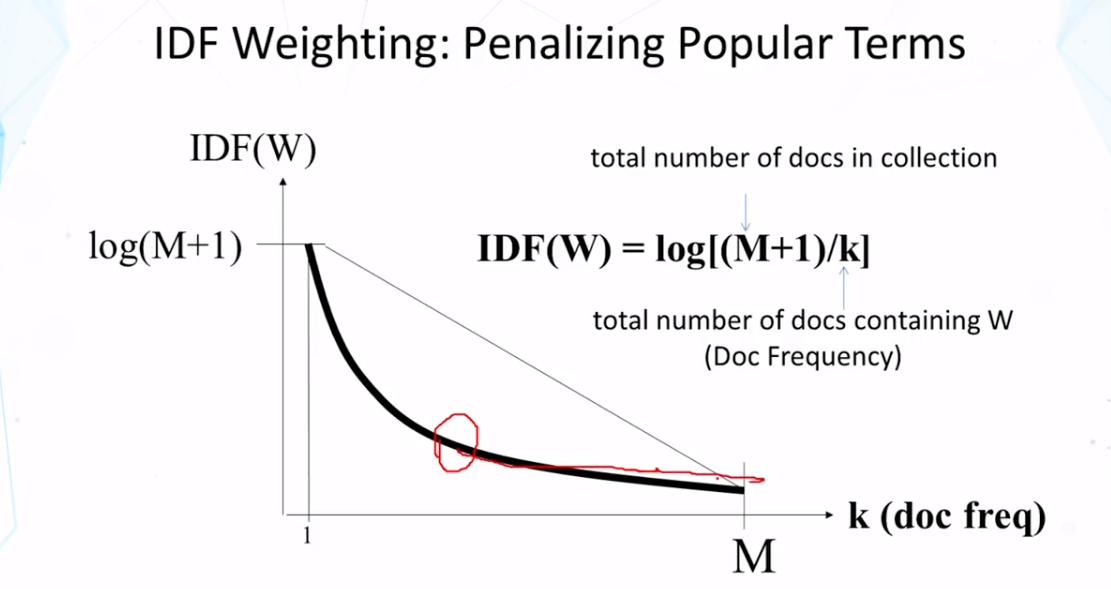
IDF: inverse Document Frequency

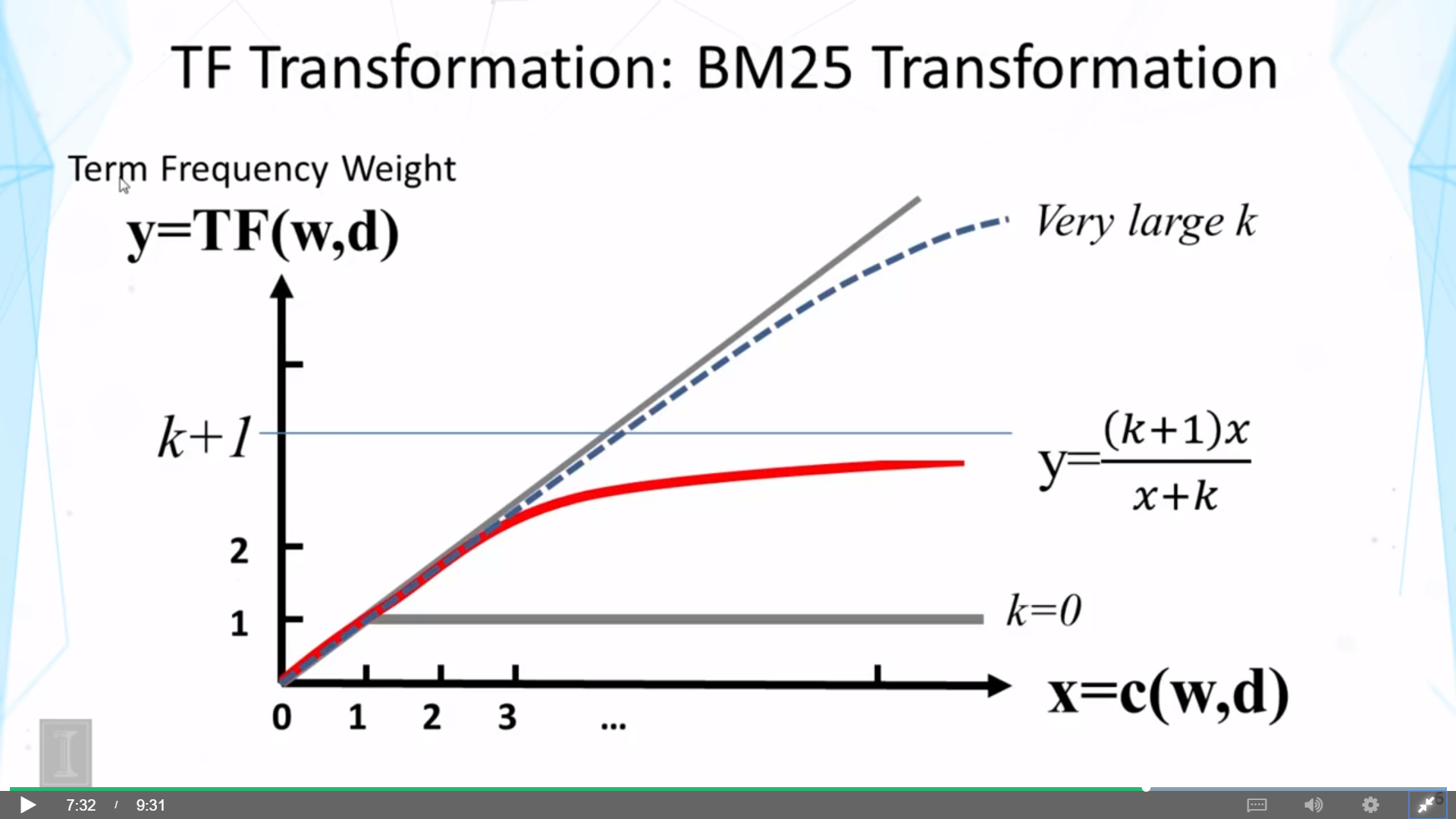
Reward a word that doesn’t occur in many documents

Panelize common words which have a lower IDF and reward rare words which have a higher IDF

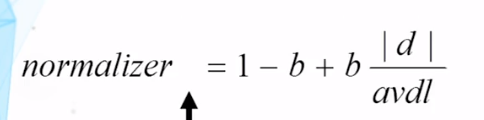
IDF = log((M + 1)/k)

Log is better than linear, since there is a turning point, lower than that is considered at useless words





M = 5



b > 0,

long, length > av, |d|/avdl > 1, -b + b \* |d|/avdl > 0

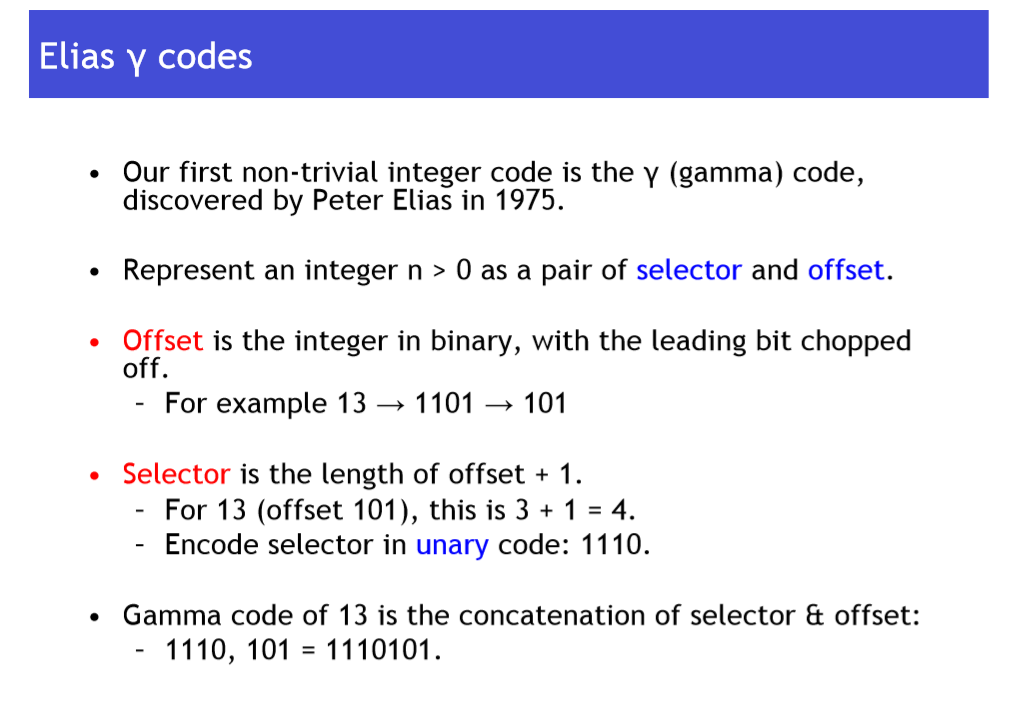
short, length <av, |d|/avdl < 1, -b + b \* |d|/avdl < 0

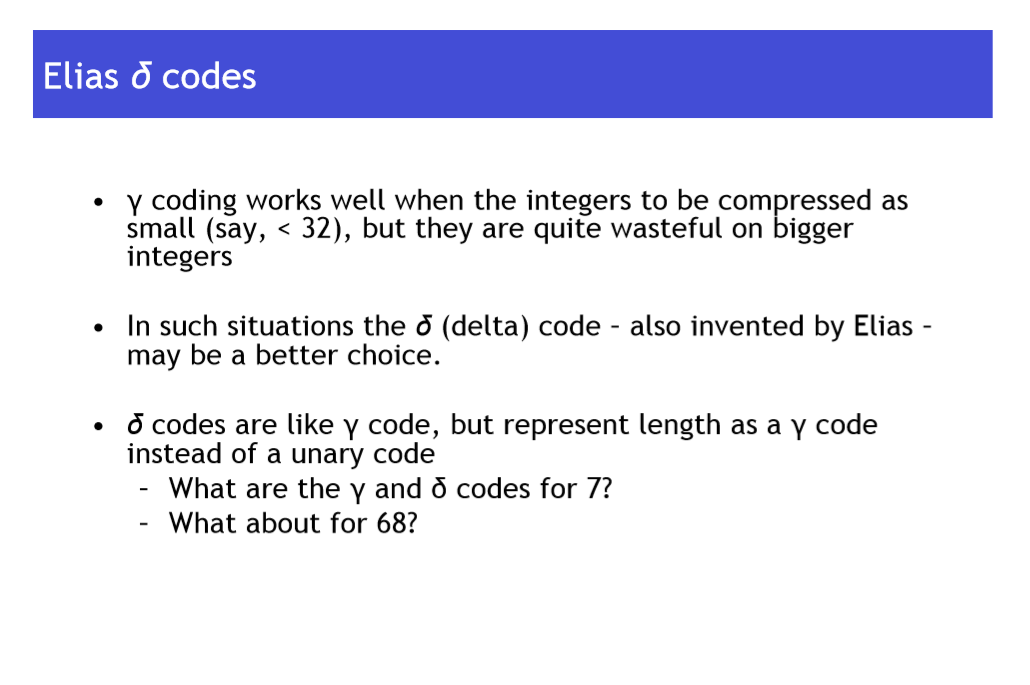
A & B: conjunctive => intersection

A or B: disjunctive => union

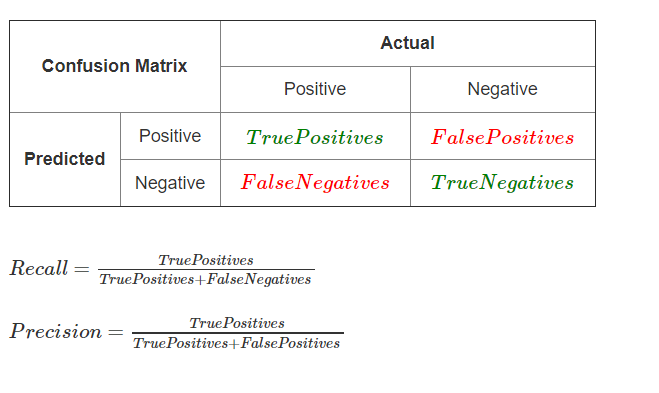
Inverted Index: why it is called inverted?

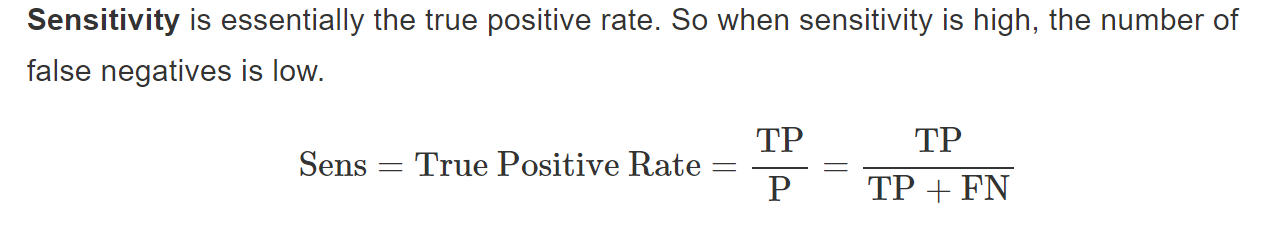
Word = > doc instead doc => word





Week3

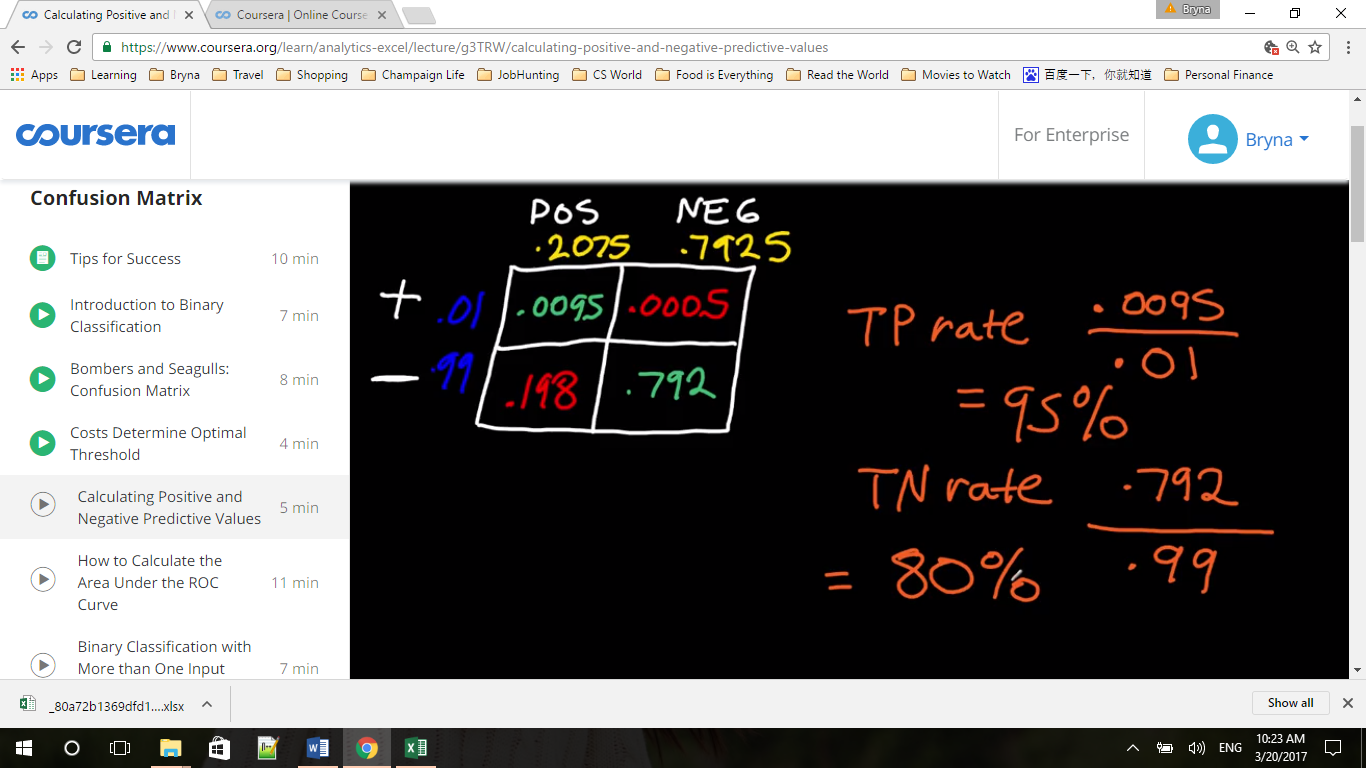




Recall = TPR = sensitivity



Precision = Positive Predictive Values



A true positive rate of 95% is conditional probability of having a having a positive test if I have cancer. And true negative rate is the p of having a negative test if I do not cancer.

What we want to know here is the latter: the conditional probability that we have the cancer if I have a positive test, or I don’t have cancer if I have a negative test?

P(POS TEST / + ) VS P( + / POS TEST)

P(NEG TEST/ - ) VS P( - / NEG TEST)

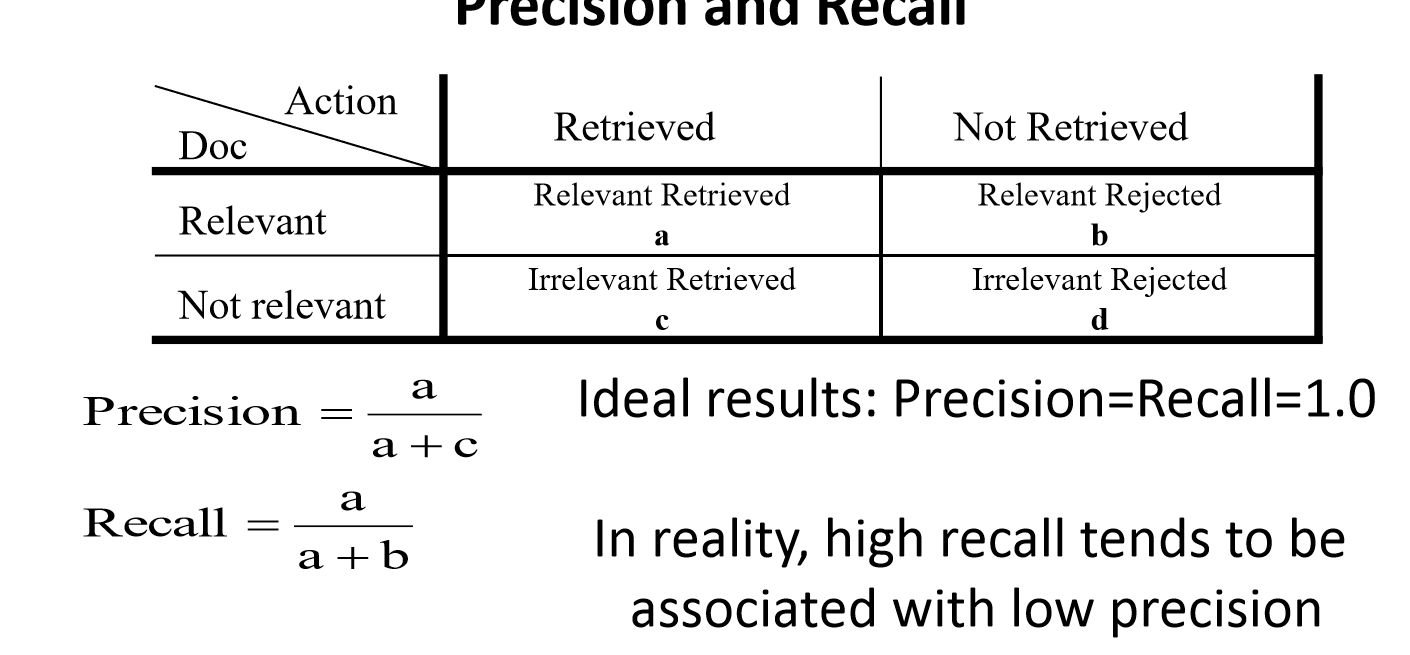
前者是说，我有病能测出来的准确率，后者是我测出来有病而我实际有病的准确率，不一样

计算方法：前者TP除以所有实际的True（上面两格）,后者TP除以测出来的Pos(左侧两格)

The latter is called positive predictive value (green yellow )

= 0.095/0.2075 = 4.58%, if I received a positive test, I have a 4.58% chance of having cancer

negative predictive value = 0.792/0.7925 = 99.937%, means that if the test turns out negative, I have 99.937% chance of not having cancer, 0.063% chance of having cancer

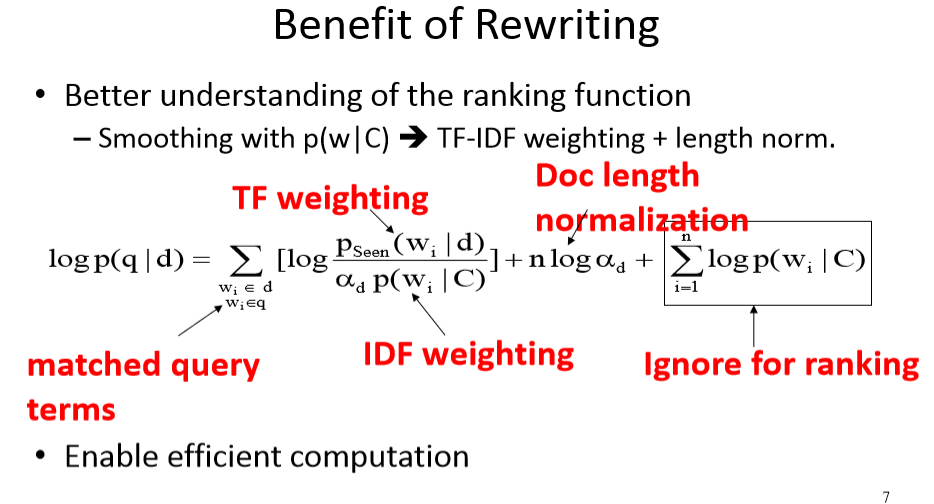


Compare MAP and gMAP

MAP = arithmetic mean of average precision over a set of queries, dominated by large values(where query is easy and average precision is high), this method is preferred when you want to improve overall performance

gMAP = geometric mean of average precision over a set of queries, affected more by low values(where query is not well performed and average precision is low), this method is preferred when you focus complicated queries

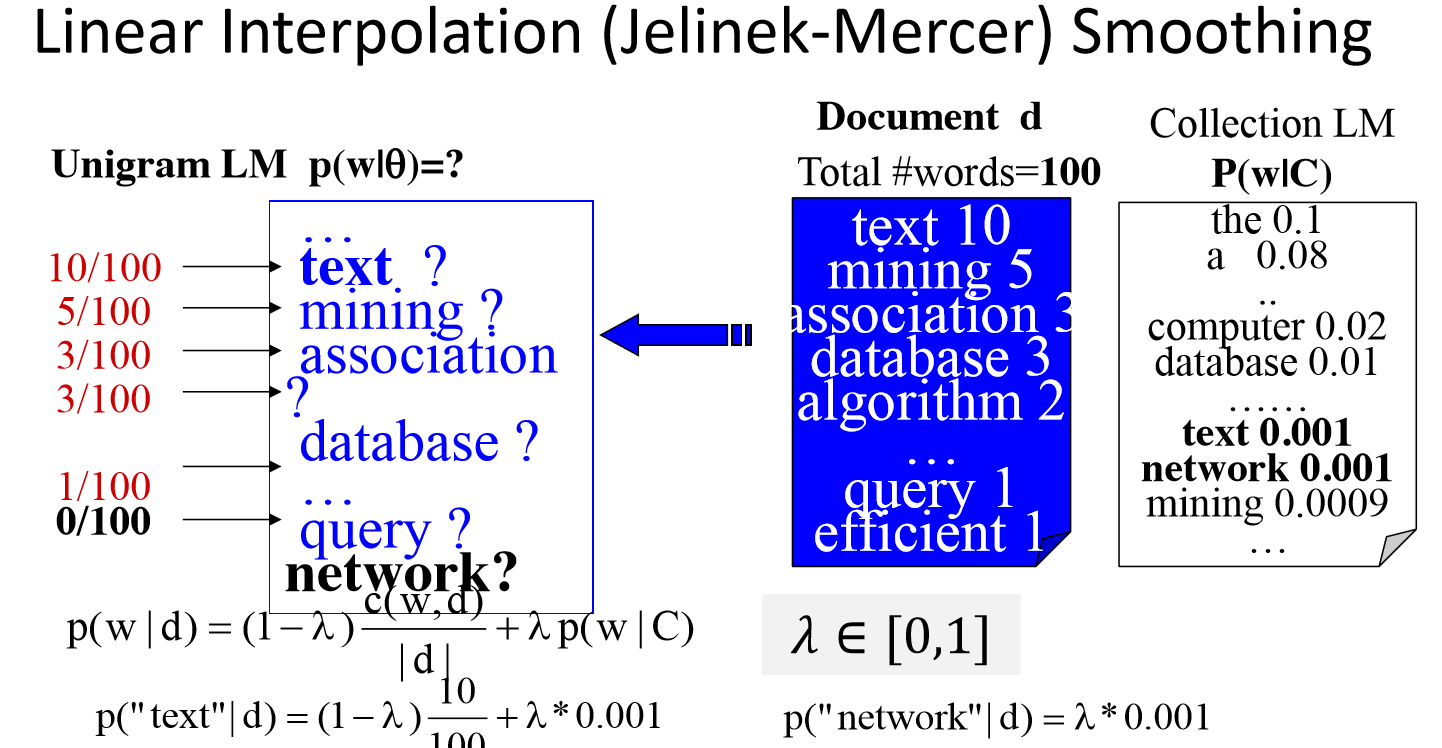
Week4



Alpha d: if document is long, we assume more words could been observed and need to do less smoothing, alpha d is small;

If document is short, more smoothing, alpha d is large

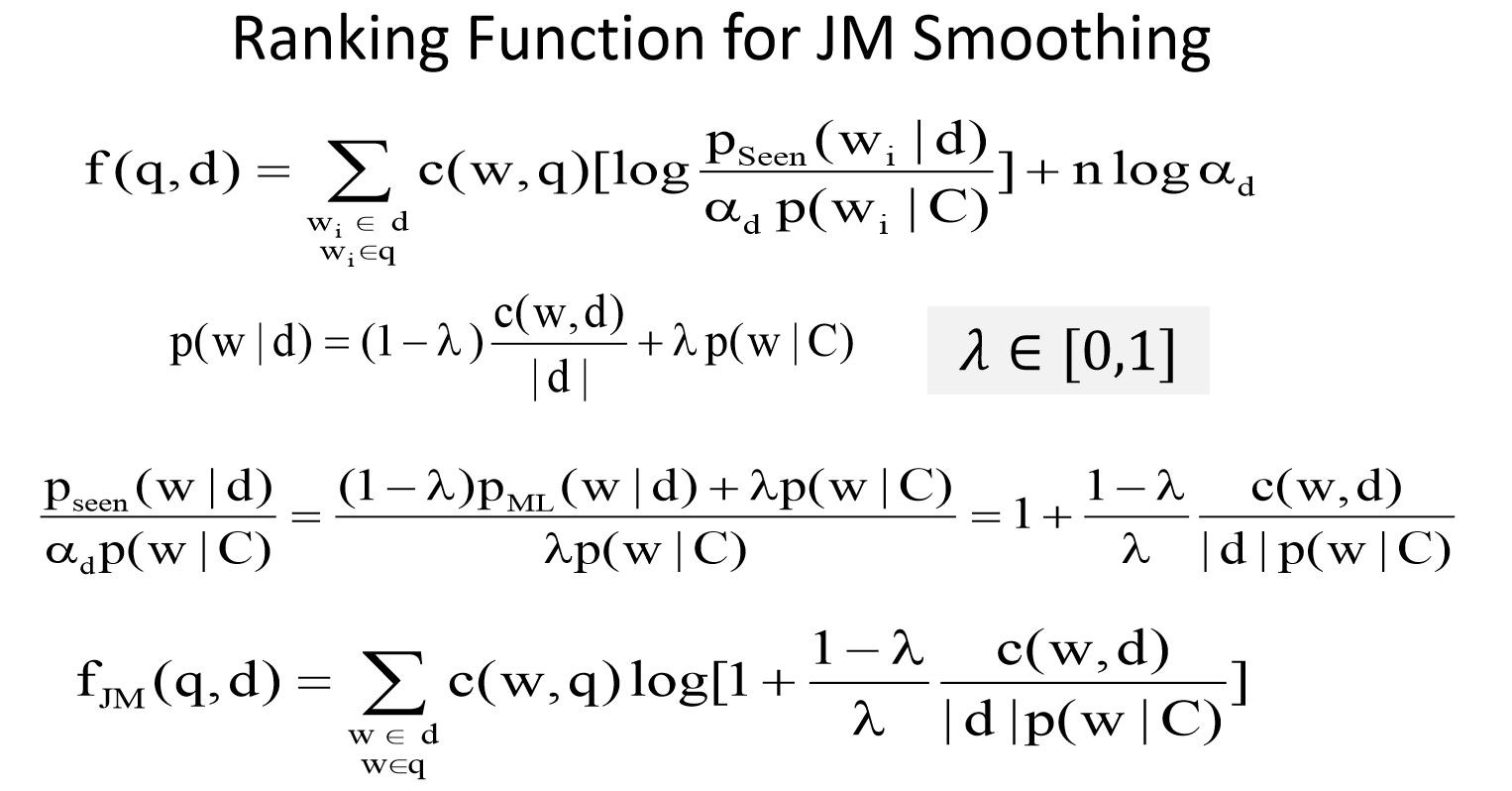
So alpha d penalizes long documents



Lamba = alpha d



Mu/100+mu => alpha d

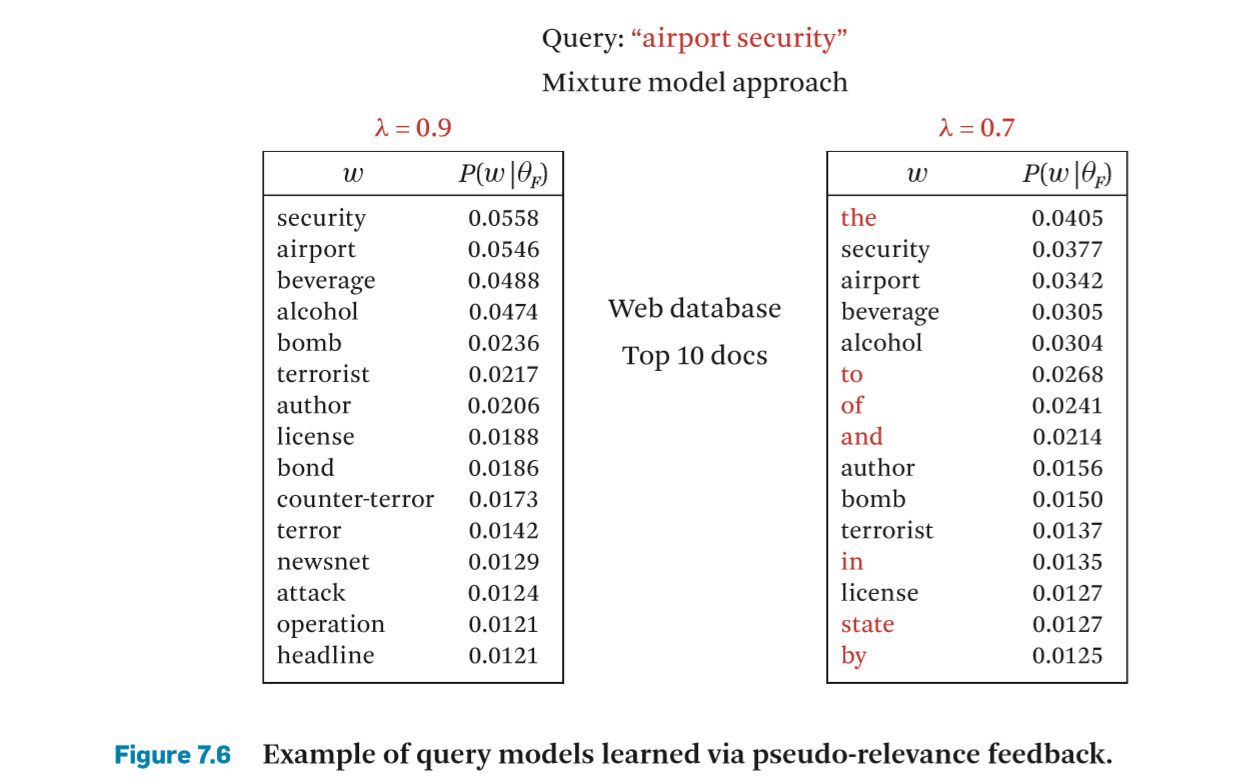
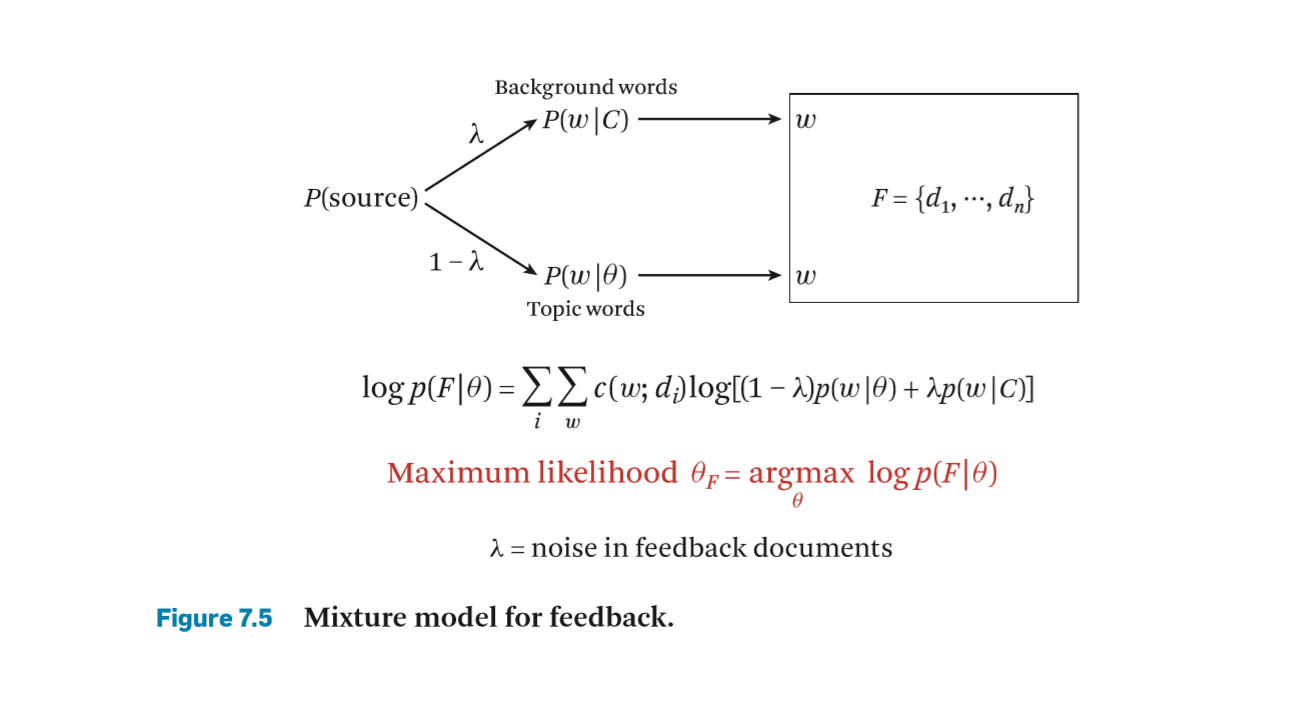


Word number d \* a word from Collection =>

If we write a document of d length, how many choices do I have the only one choice with these word

Week5

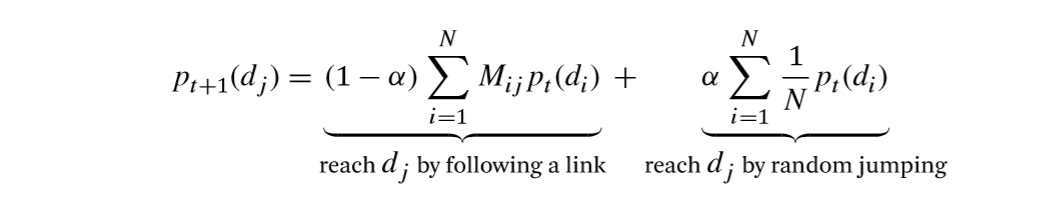
Lambda in Mixture model for feedback

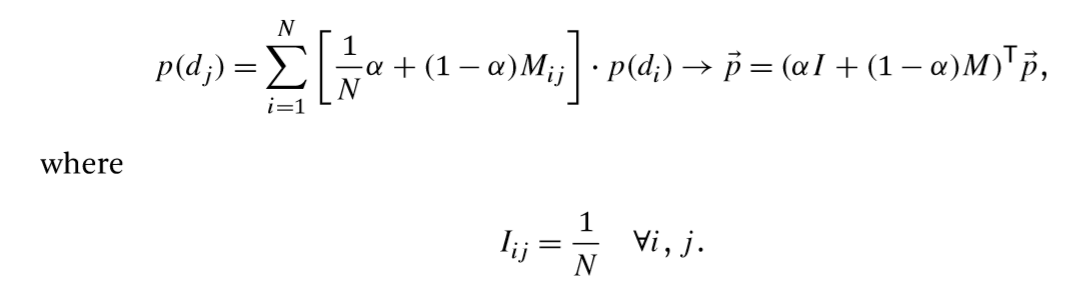


Lambda controls the probability of using background model

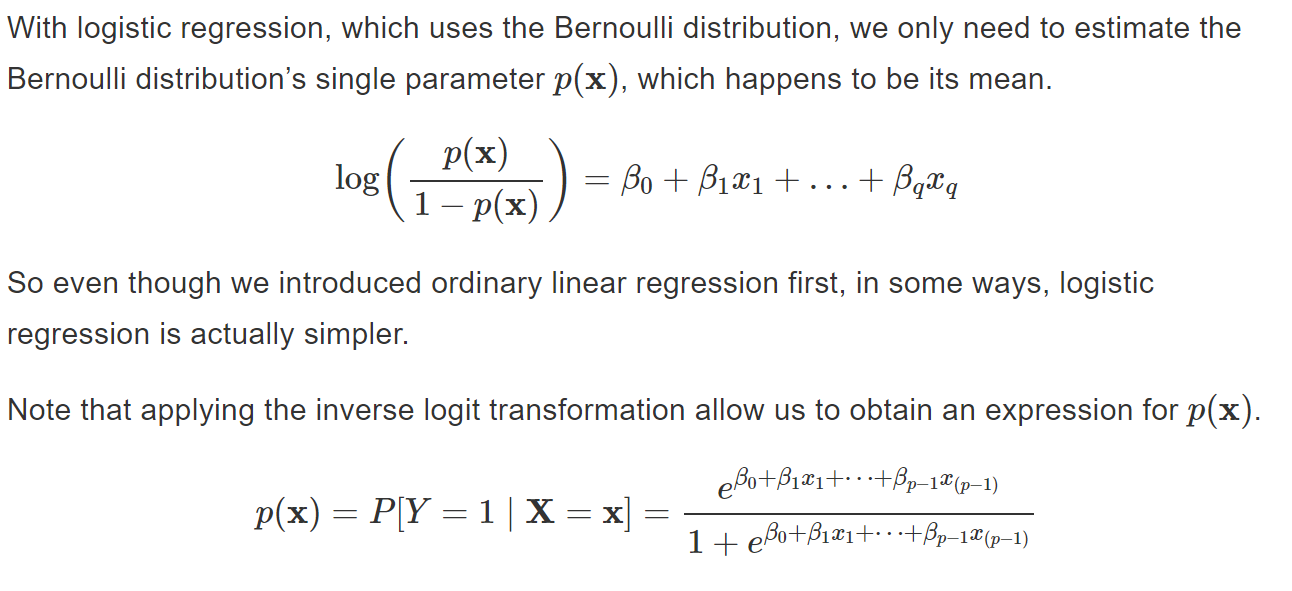
mixing parameter λ is explained as: If λ is high, it’s going to prefer the background distribution. Conversely, if λ is very small, we’re going to use only our topic words. However Figure 7.6 compares two models when λ = 0.9 and λ = 0.7, you can see from the picture below, the 0.9 model includes more topic words while the 0.7 model seems to prefer background distribution.

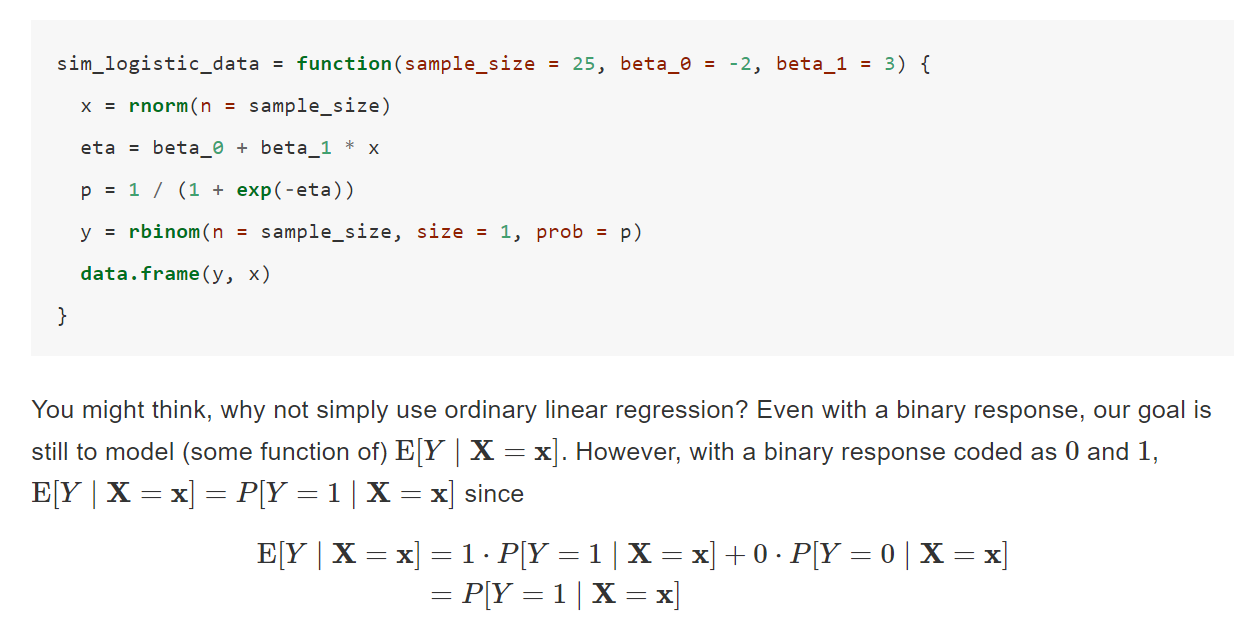
when λ = 0.9, background distribution is preferred => the topic model is very discriminative—it contains all the relevant words without common words





Week 6



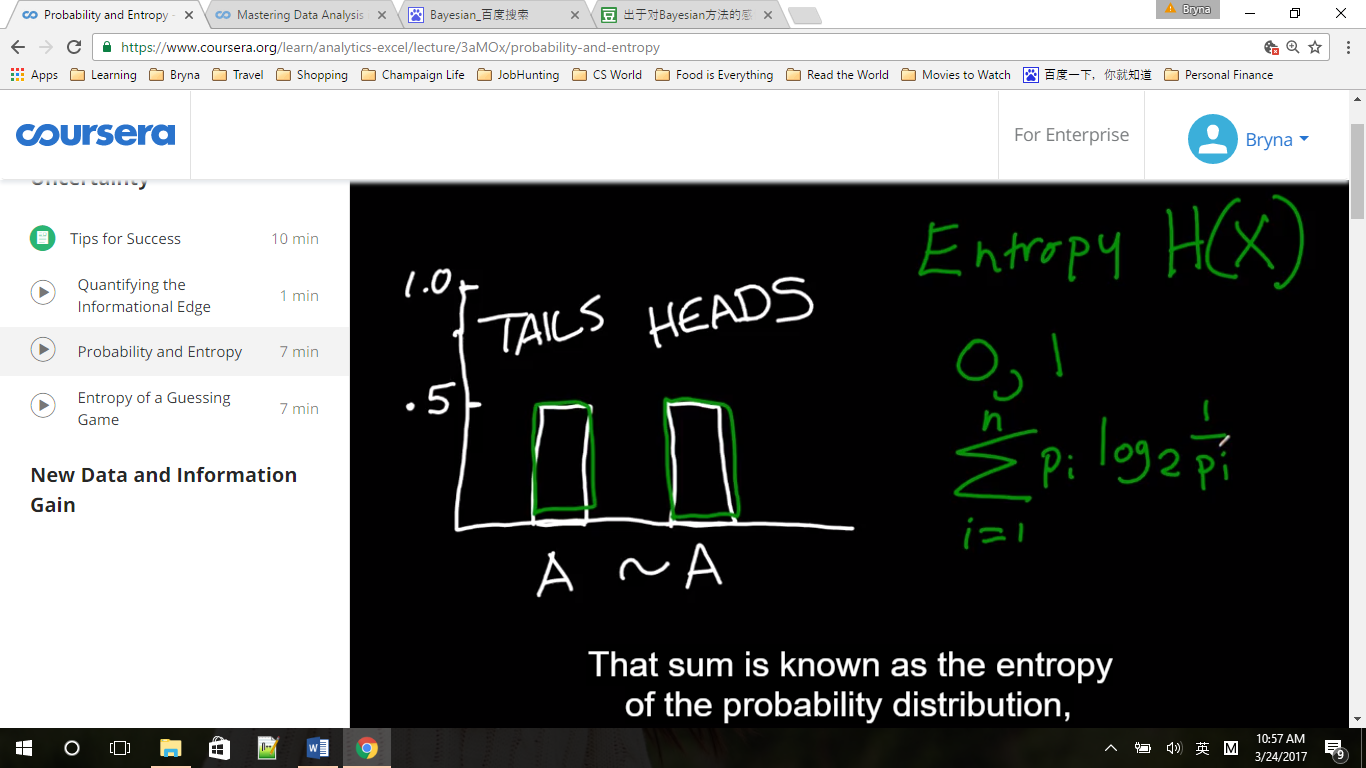


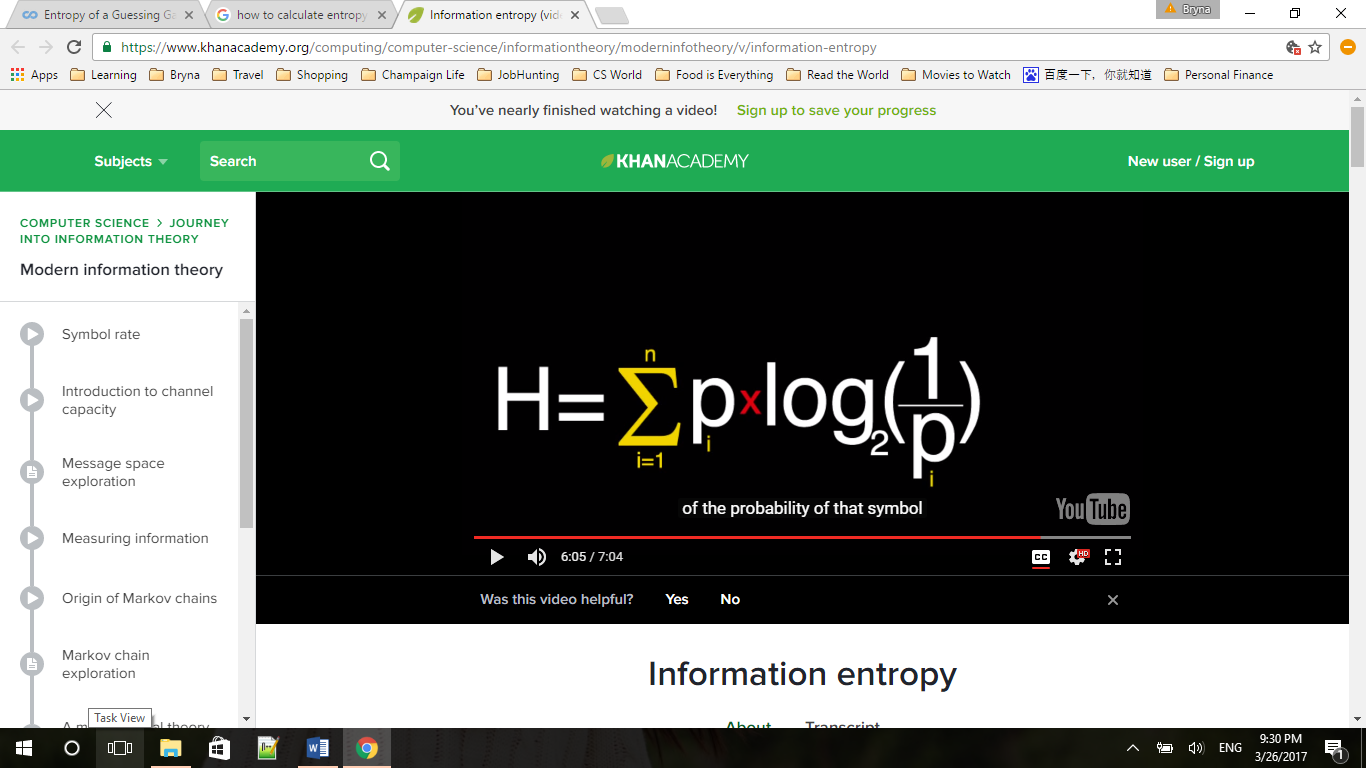
Week7

Paradigmatic 范例的（变形的）

Syntagmatic 组合的

Week 8



 from Khan Academy

Entropy is the sum of each probability times the log to the base 2 of one over that probability

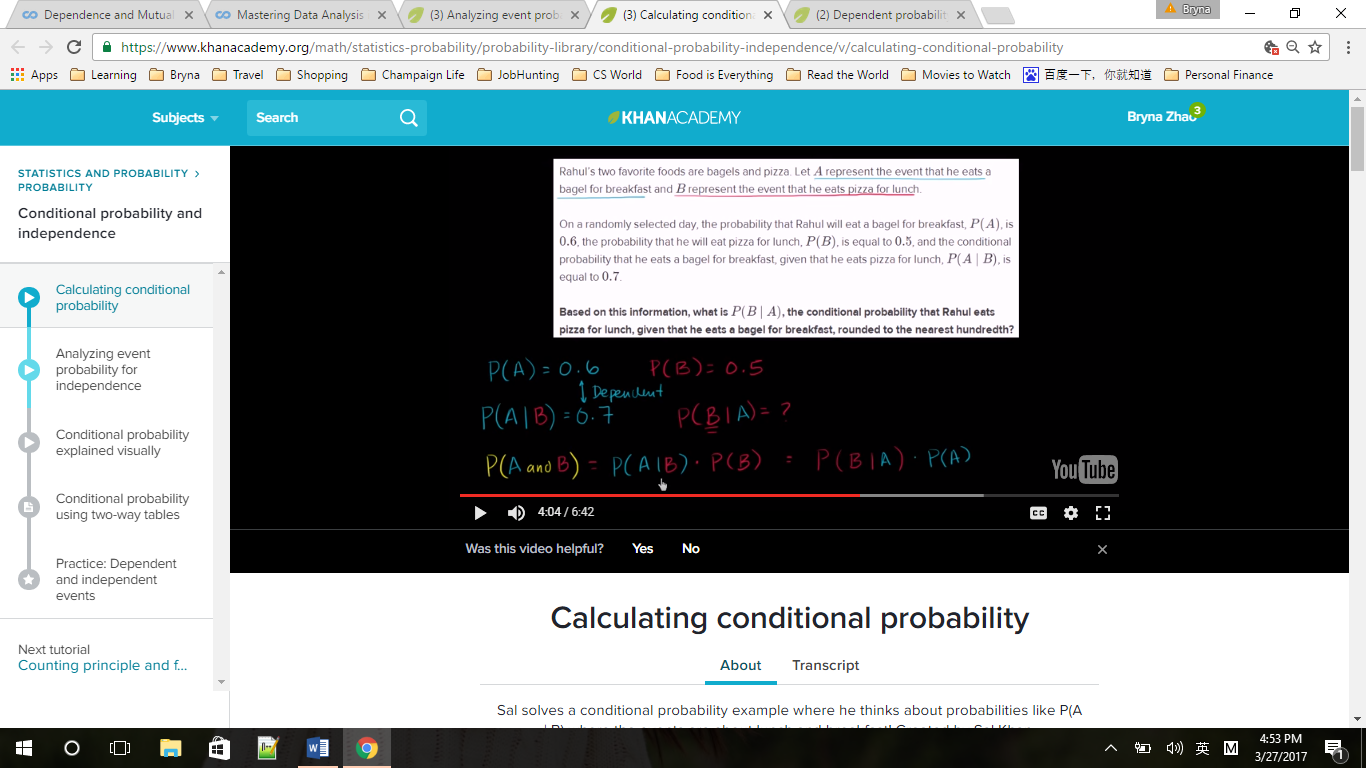
If we have a uniform distribution (均匀分布) with n possible outcomes, H(X) = log2n, also the maximum of entropy, other distribution will have a smaller H than a discrete distribution

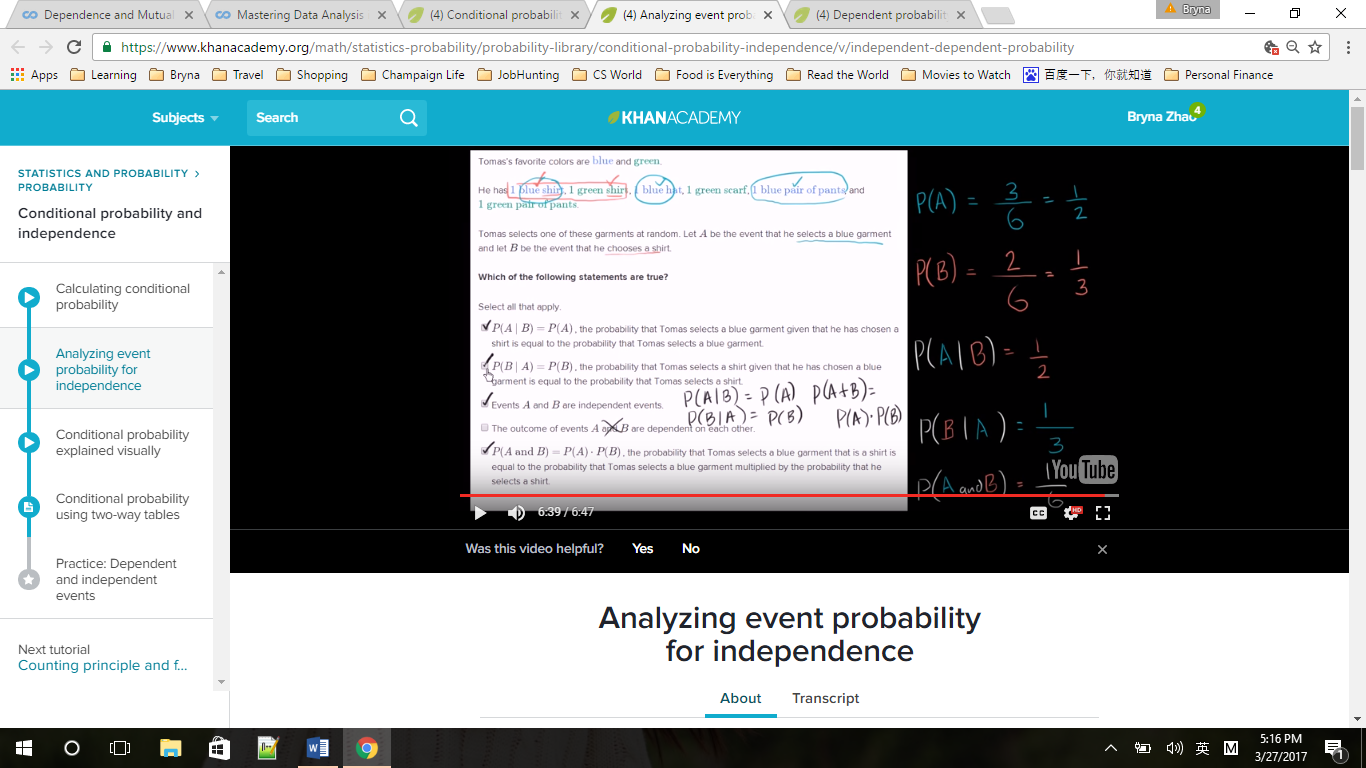
(for example, tail or head, p =1/2, H(X) = 1/2\*log2(1/(1/2)) +1/2\*log2(1/(1/2)) = log22 = 1

Or p=1/6, H(X) = 6\*( 1/6\*log2(1/(1/6))) = log26

After we’ve thrown the coin, P(Tails) = 1, H(X) = 0

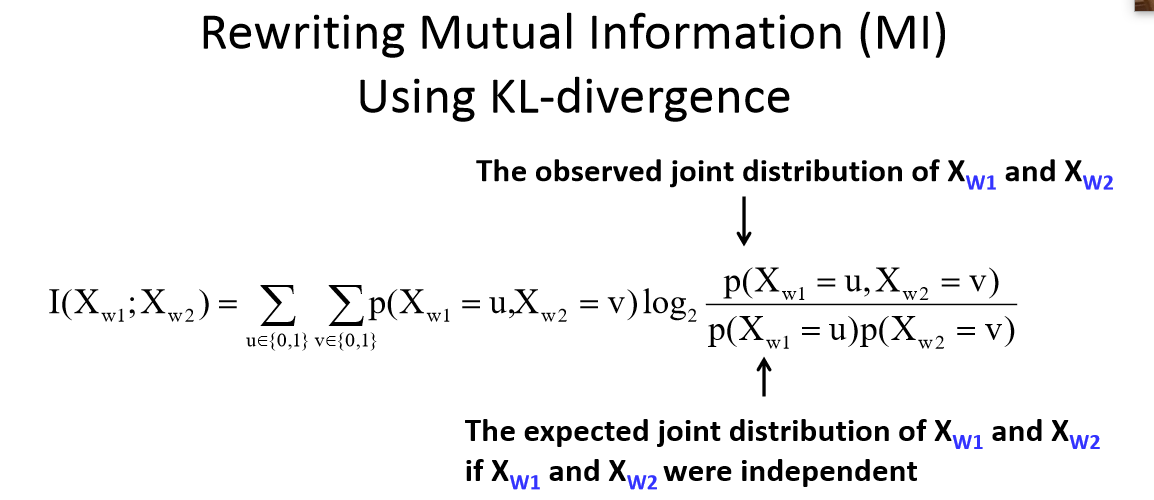
The more uncertainty we have, the bigger it gets, minimum at zero



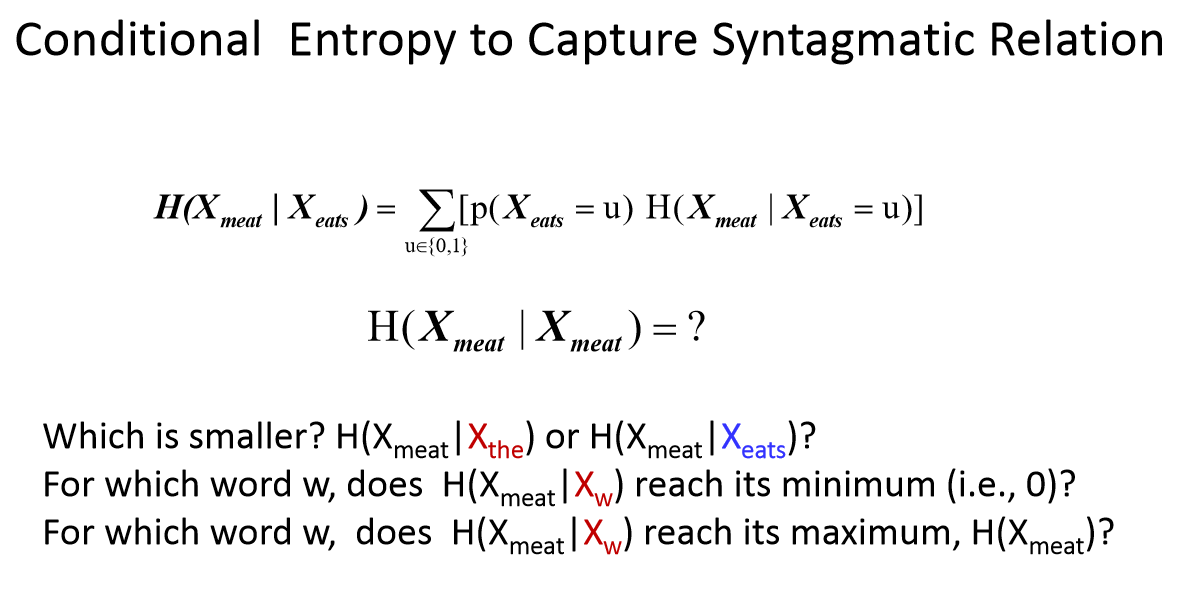


Mode: The **mode** of a set of data values is the value that appears most often.[[1]](https://en.wikipedia.org/wiki/Mode_(statistics)#cite_note-1) It is the value *x* at which its [probability mass function](https://en.wikipedia.org/wiki/Probability_mass_function)takes its maximum value. In other words, it is the value that is most likely to be sampled.

<https://en.wikipedia.org/wiki/Mode_(statistics)>



The larger divergence, the higher mutual information



H(Xmeat|Xthe) > H(Xmeat|Xeats)

H is the uncertainty, so when we know the, we know little about meat, so H is larger

week 9



P(w|ϴ) is the word distribution of ϴ, is randomly assigned,

P(z=0|w) is the probability of the word coming from the topic distribution(by-product)

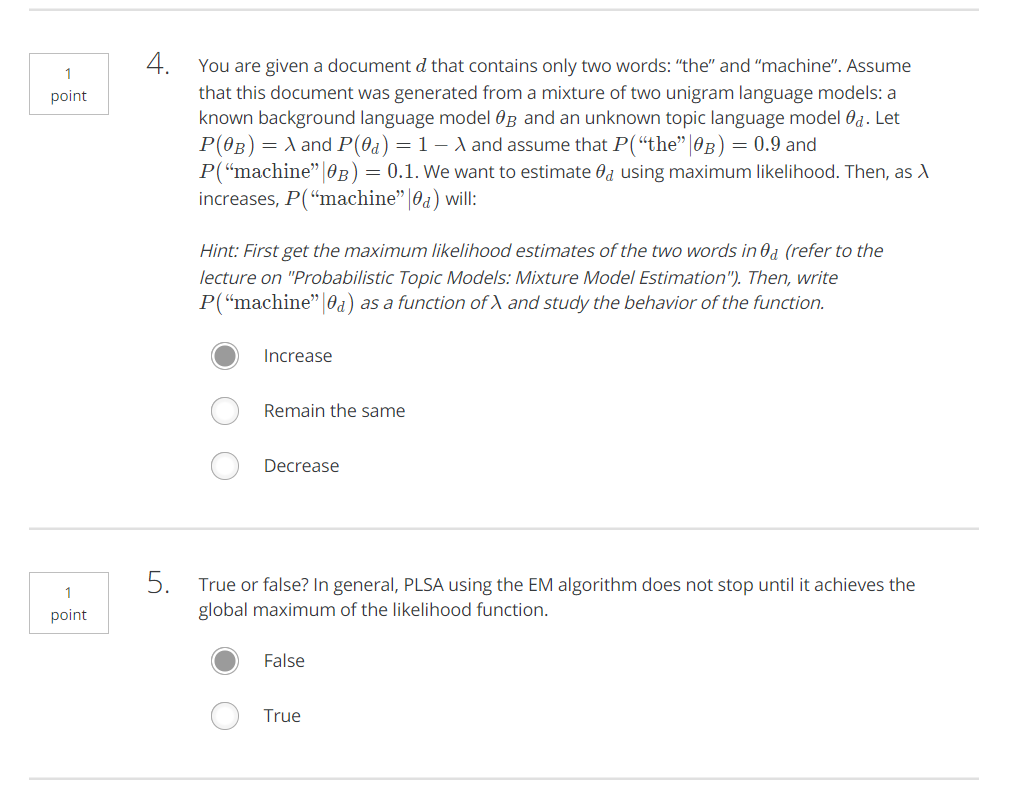
Iteration 1:

P(w|ϴ) is randomly assigned,

P(z=0|w) changed because p(w| ϴB) are different

Iteration 2:

P(w|ϴ) changed, because word account has play roles



Maximum likelihood = p(“machine”)\* p(“the”)

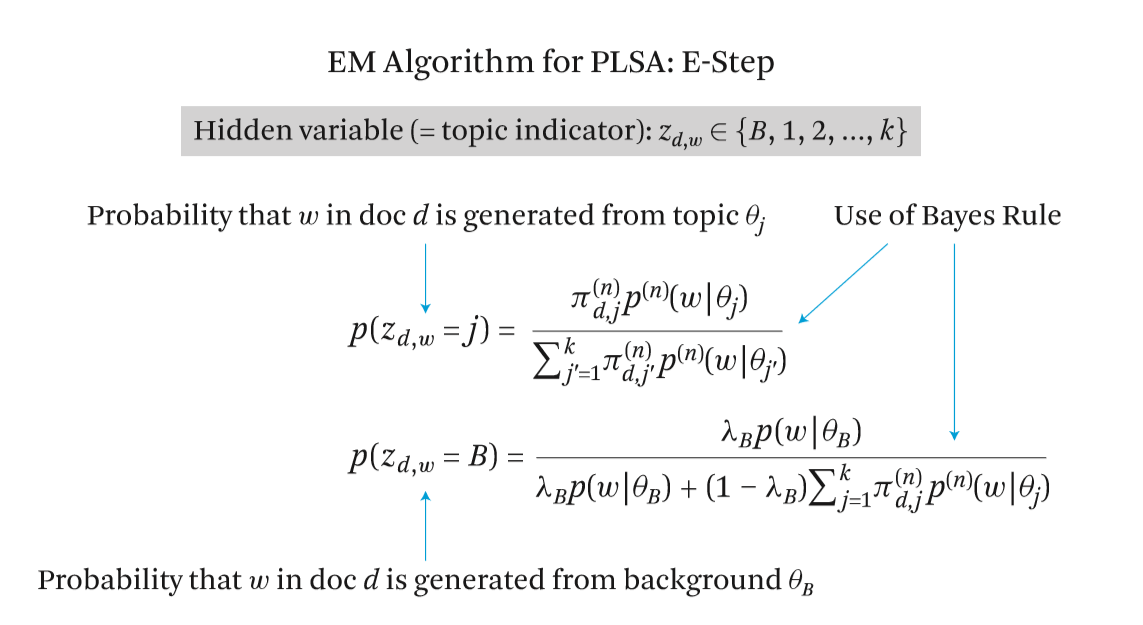
p(“machine”) + p(“the”) = 1,两个相等的时候ML 最大

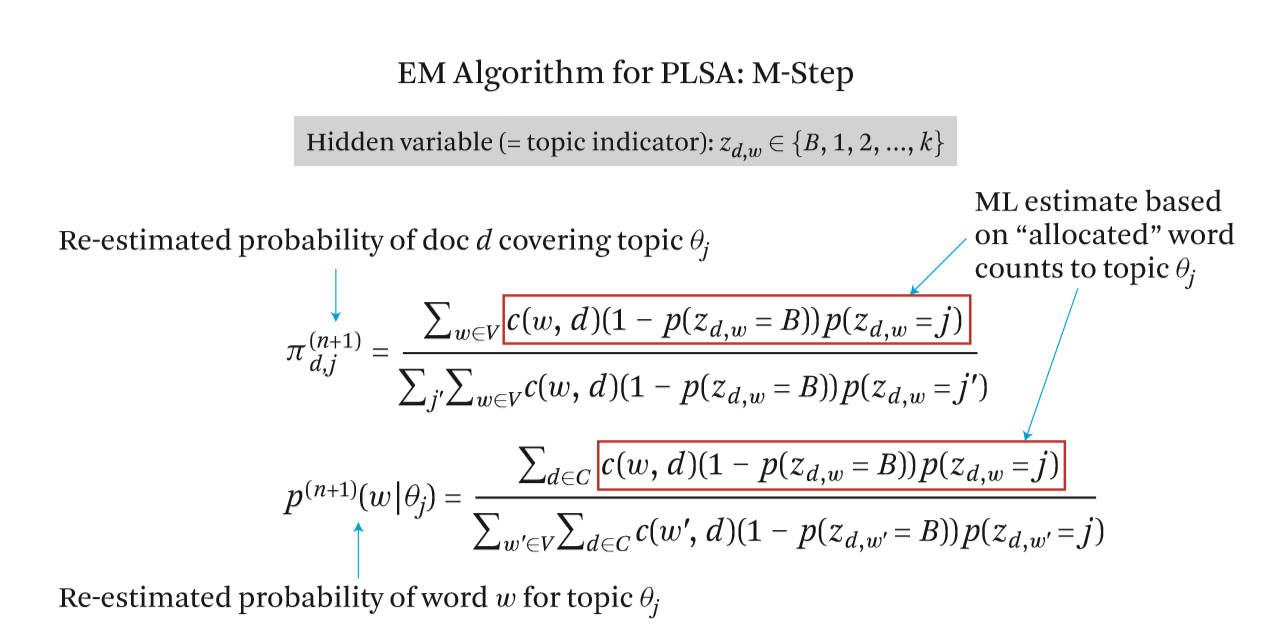
于是就有了

(1-λ)\* P(“machine” | ϴd) + λ \* P(“machine”|ϴB) = ( 1- λ)\* P(“the” | ϴd) + λ \* P(“the”|ϴB)

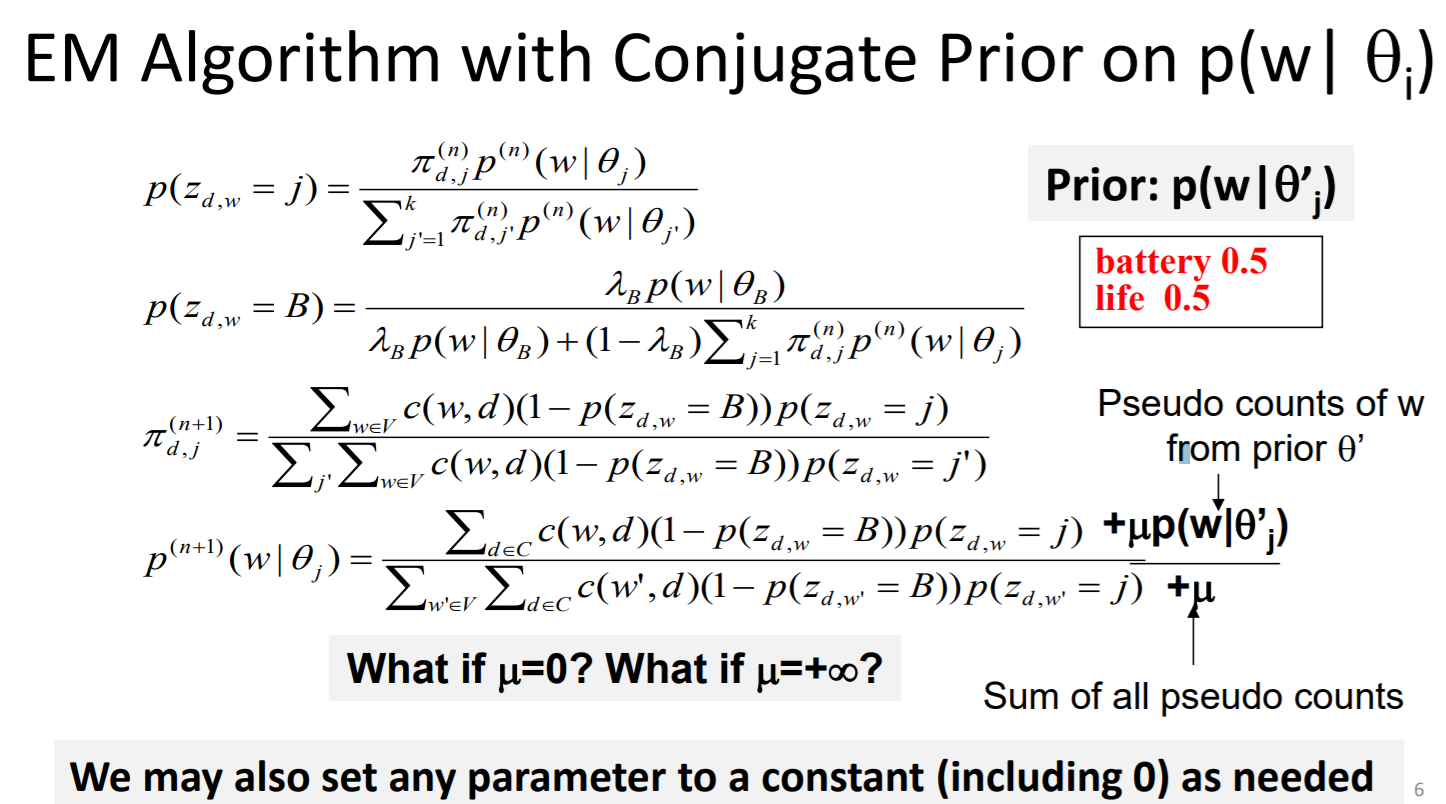
(1 - λ)\* P(“machine” | ϴd) + 0.1 λ = ( 1- λ)\* (1 - P(“machine” | ϴd)) + 0.9 λ

2P - 1 = 0.8 λ / (1-λ) , λ越大，P越大



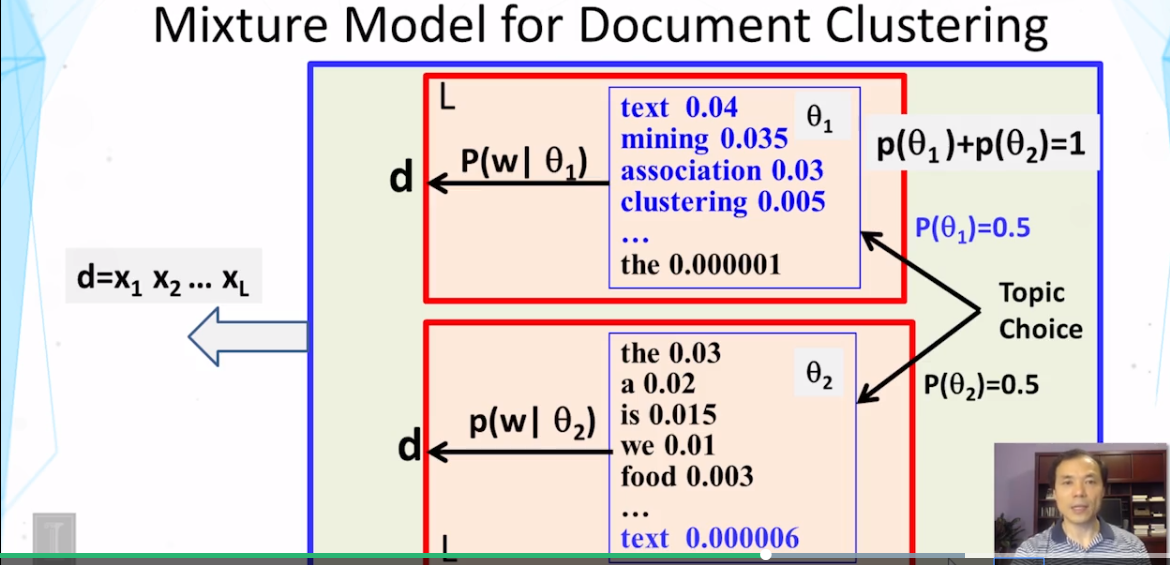


<https://blog.csdn.net/buggiant/article/details/12260555?utm_source=blogxgwz2>



mu = 0, no prior, mu = infinity, no listen to data, just prior

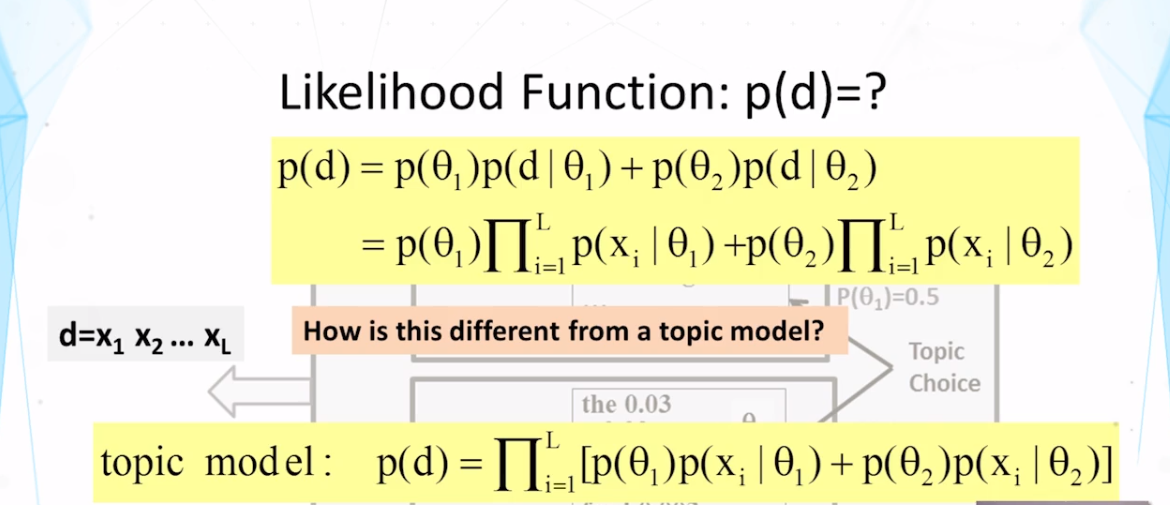
week 10



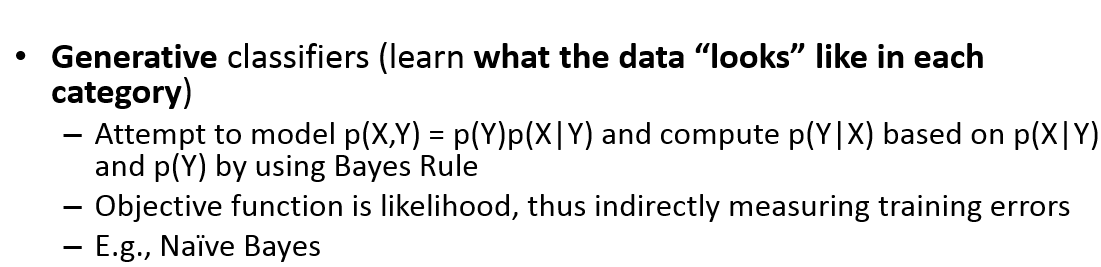
The decision of using a particular distribution is made just once for this document, in the case of document clustering.

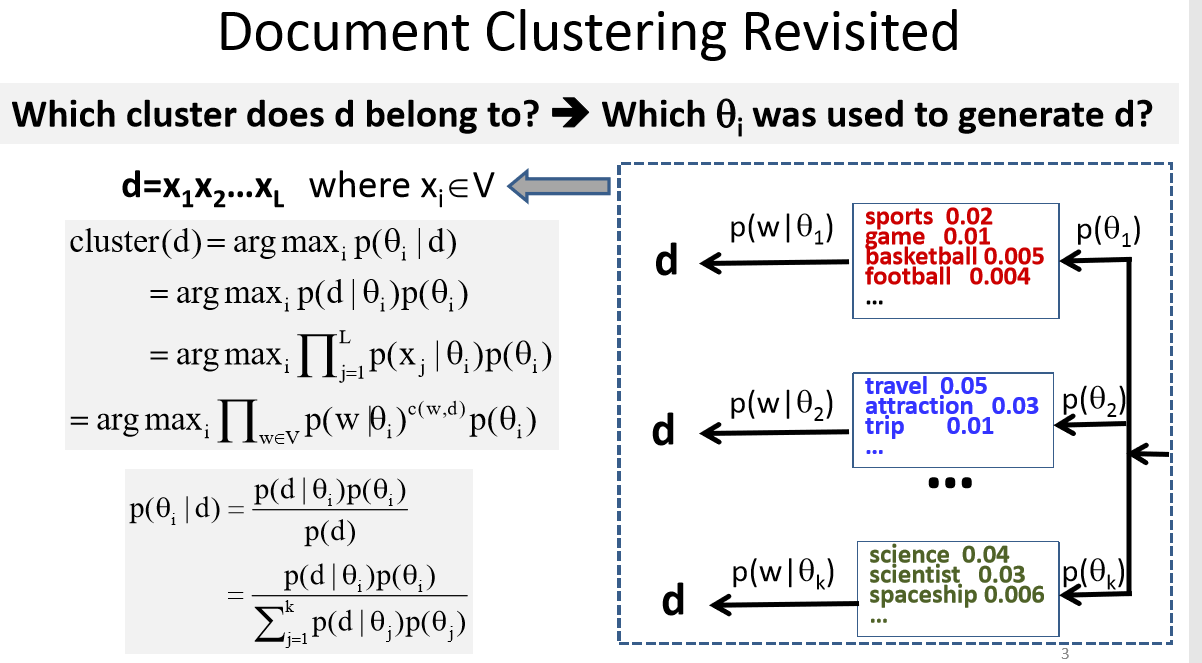
But in the case of topic model, we have to make as many decisions as the number of words in the document. Because for each word, we can make a potentially different decision.

And that's the key difference between the two models.



Week 10





One is P(ϴ), the prior

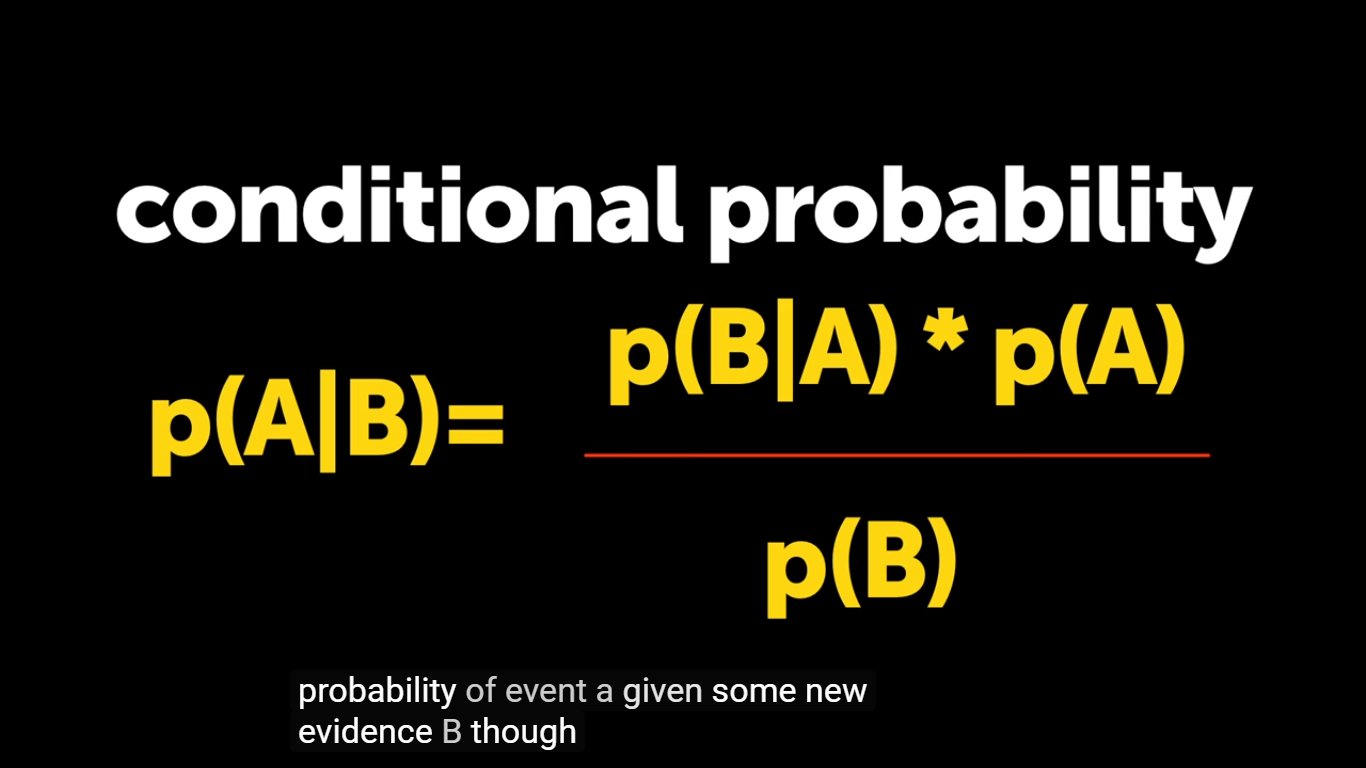
we need to consider if a topic or cluster has a higher prior then it's more likely that the document has been from this cluster.

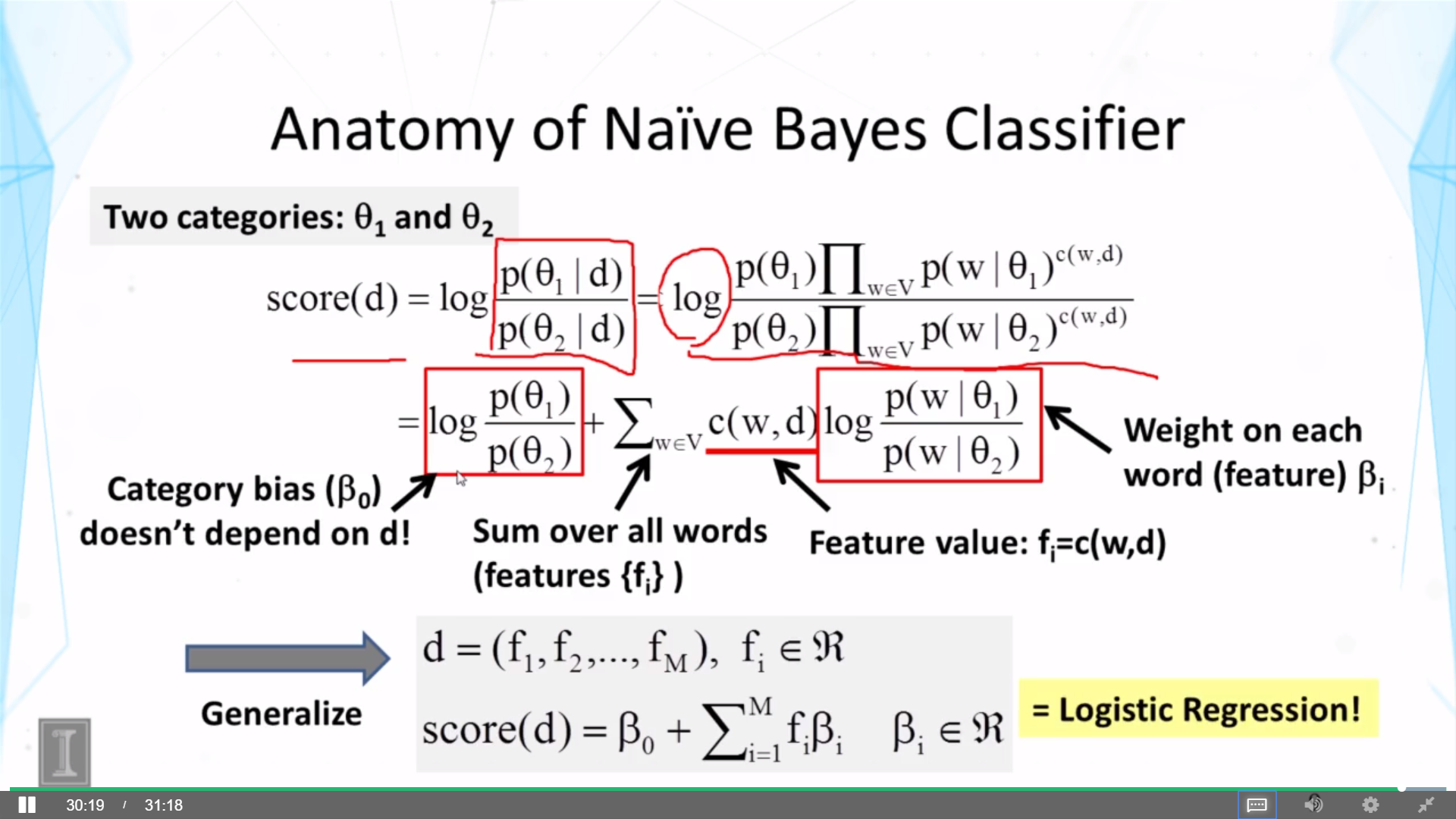
The other is a likelihood part, whether the topic word of distribution can explain the content of this document well.

And we want to pick a topic that's high by both values.

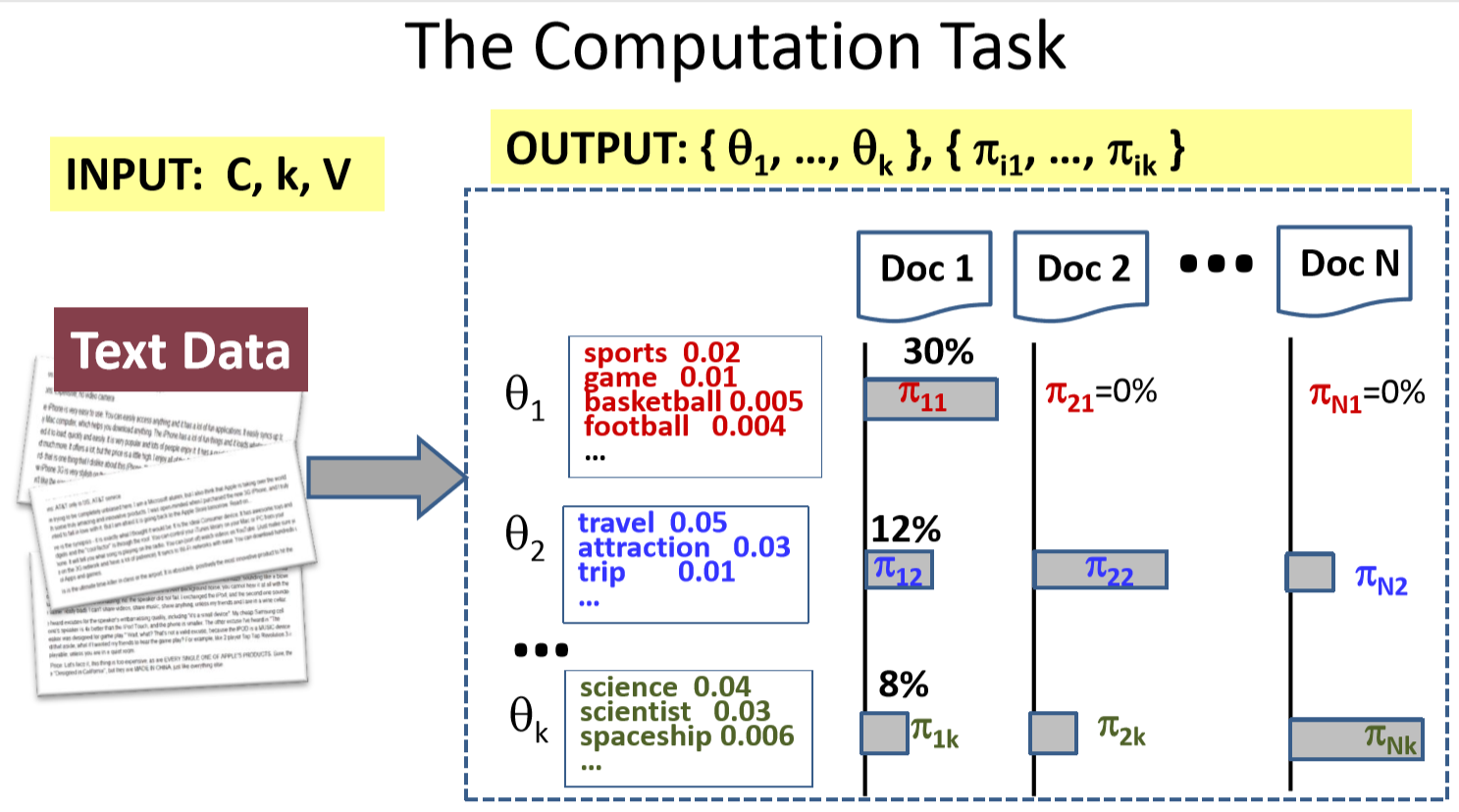
We want data given label p(Y|X)

p(Y|X) p(X) = p(X|Y) p(Y), data given label p(X|Y) and label p(Y) are known

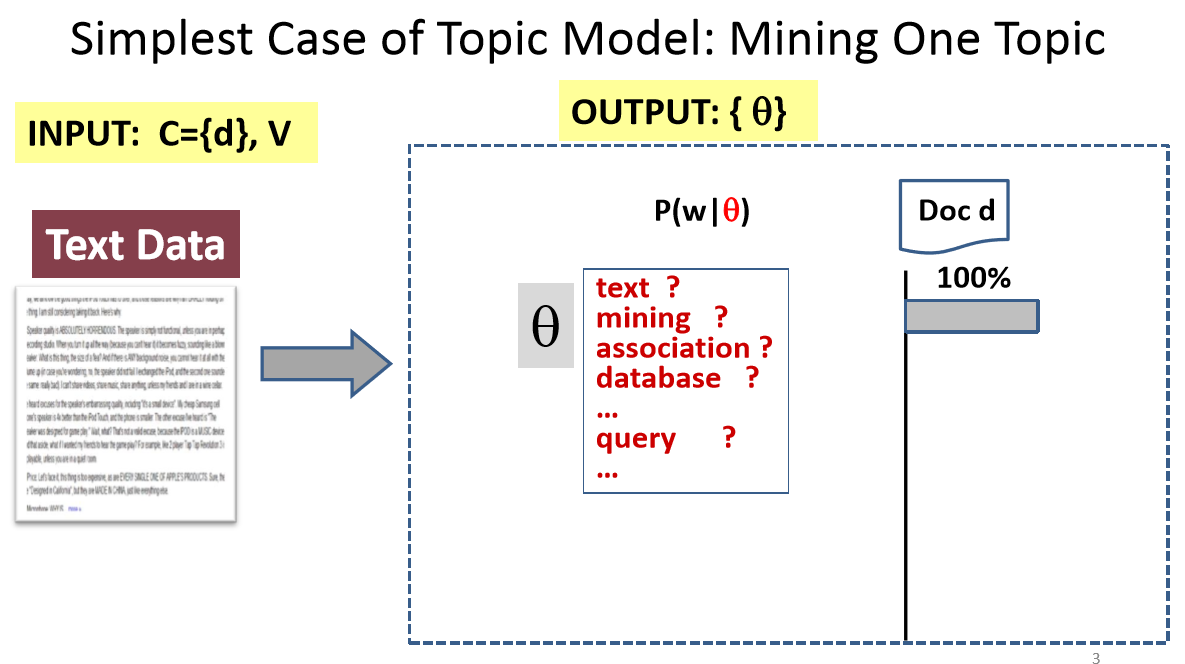


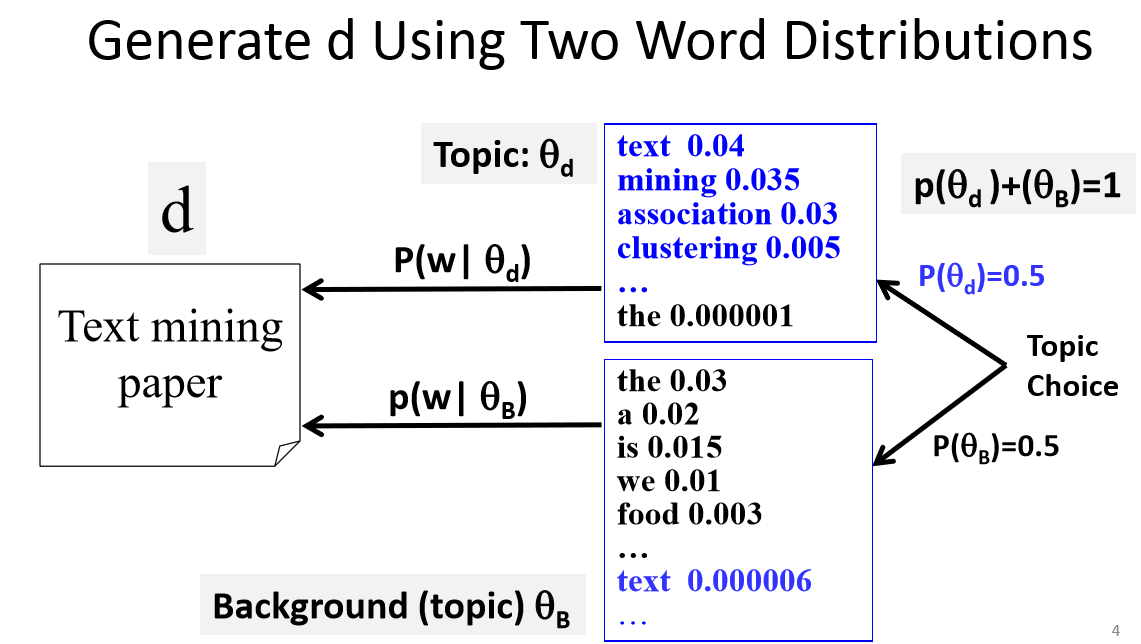


Compare with 8.7

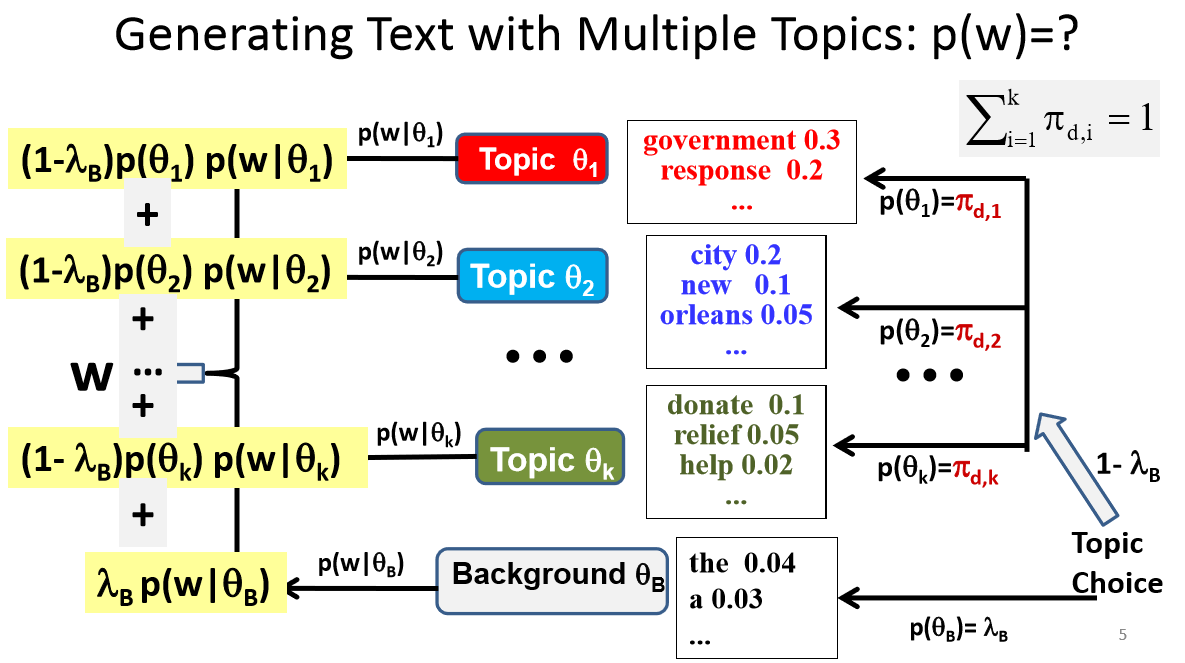


8.10





9.7



The difference between Bayesian estimation vs maximum likelihood is basically if you have prior knowledge or not.

Think of maximum likelihood a special case of Bayesian estimation. The Baysian rule states:

p(θ|D)=p(D|θ)p(θ)p(D)

When the prior knowledge is uninformative,The MLE treats p(θ)p(D) as constant, and MLE is only proportional to:

p(D|θ)

That's why in the basic PLSA (lecture 9.7) when we talk about the constrained optimization, we used

∧∗=argmax∧P(C|∧)

Whereas in the PLSA extension, PLSA with prior knowledge (lecture 9.9) the objective function now becomes

[submit](https://piazza.com/class/jky3wzh9mon231?cid=702)

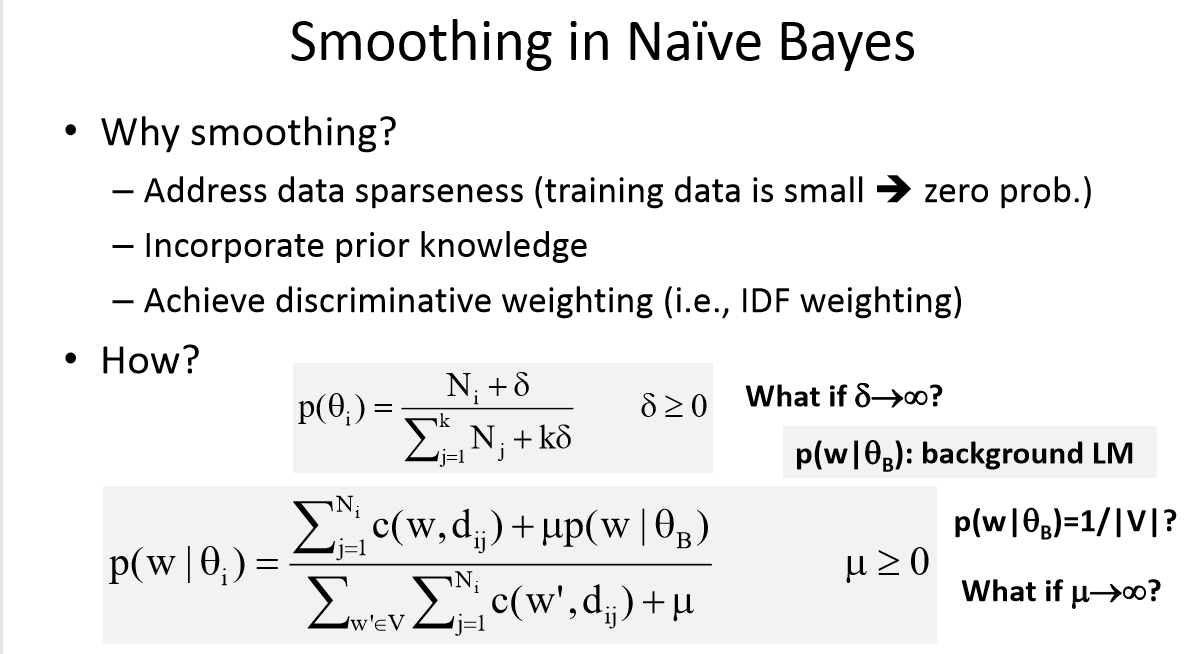
∧∗=argmax∧P(∧)P(C|∧)

So coming back to your question. It really depends on your use case. Are you referring to the basic form of PLSA? Then it's using MLE to estimate the parameters. If you are referring to the PLSA with prior knowledge, then the objective function becomes Bayesian estimation.

By the way these two forums discussed Bayesian estimation vs maximum likelihood at great length, it helped me a lot:

<https://stats.stackexchange.com/questions/74082/what-is-the-difference-in-bayesian-estimate-and-maximum-likelihood-estimate>

<https://stats.stackexchange.com/questions/58564/help-me-understand-bayesian-prior-and-posterior-distributions/58792#58792>



Larger δ=> more smoothing => more rely on pseudocounts => δ -> infinity, every category infinite documents => no distinct between then, uniform; if δ -> 0, rely training data

µp(w|B) => dynamic smoothing, Common words get more pseudocount,

µ-> infinity => p(w|ϴ) -> background model

p(w| ϴB) = 1/|V|, constant pseudocount