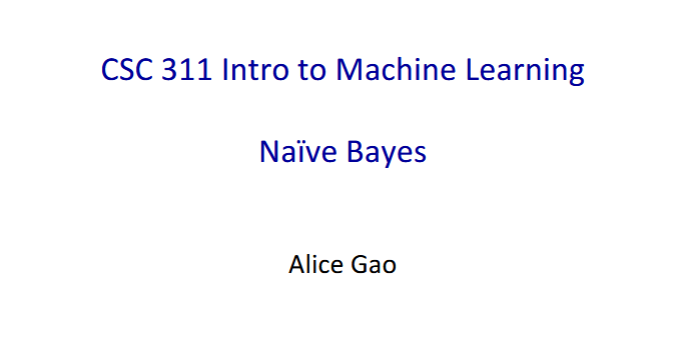
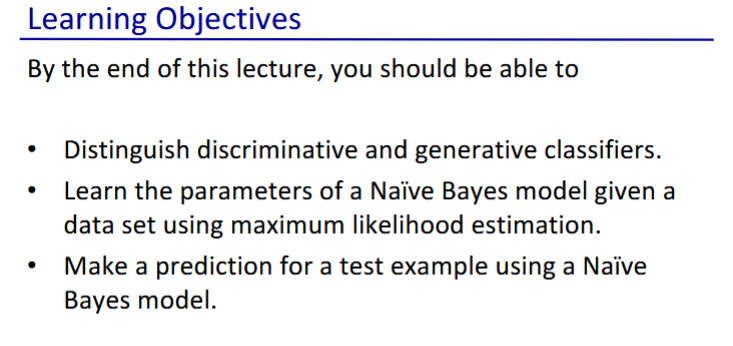
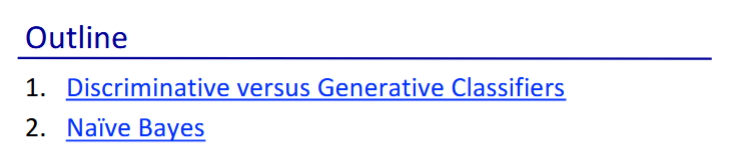
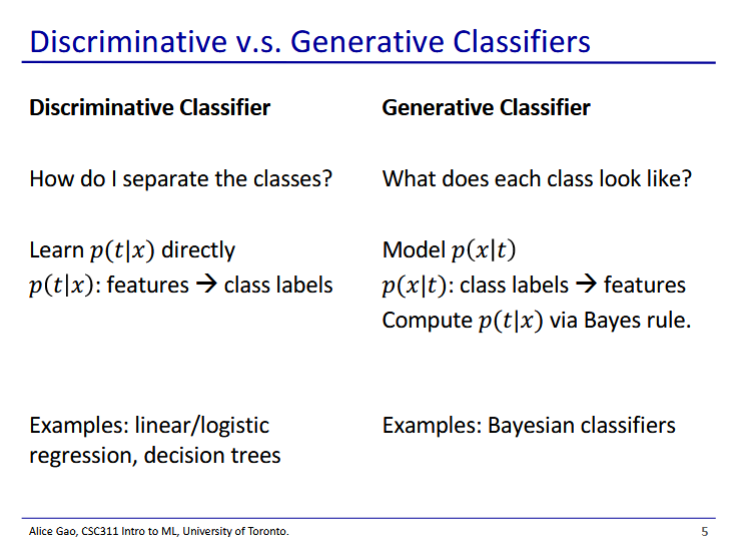
|  |
| --- |
| **Discriminative vs generative classifiers**   * Discriminative classifiers   + Directly learns the class based on its features (learns)   + Examples are all the models we have learned previously (linear regression, logistic regression, softmax regression, etc. ) * Generative classifiers   + Tries to learn what each class looks like, then uses Bayes’ rule to make predictions     - Learns first, then uses Bayes' rule to calculate   + Examples are Bayesian classifiers, generative AI models   **Naive Bayes**   * Called naive because the model makes a simplifying assumption   + Assumes that features are conditionally independent given the class   + Assumption may not be true to reality, however it greatly simplifies calculations     - Allows to be decomposed into * Breakdown of log-likelihood is between slides 15 and 16 * Predict class by choosing the class with the highest conditional probability   + Example prediction on slide 25   **Text sentiment analysis with Naive Bayes**   * Naive Bayes can be used for sentiment analysis (sorting text into positive/negative categories) * First we assign features to each sentence   + We have a feature for each word that appears in the dataset   + The feature is set to 1 if the word appears in the sentence, 0 otherwise     - Order of words does not matter * Then we learn the prior () * Then we learn the conditional probability for each feature () |





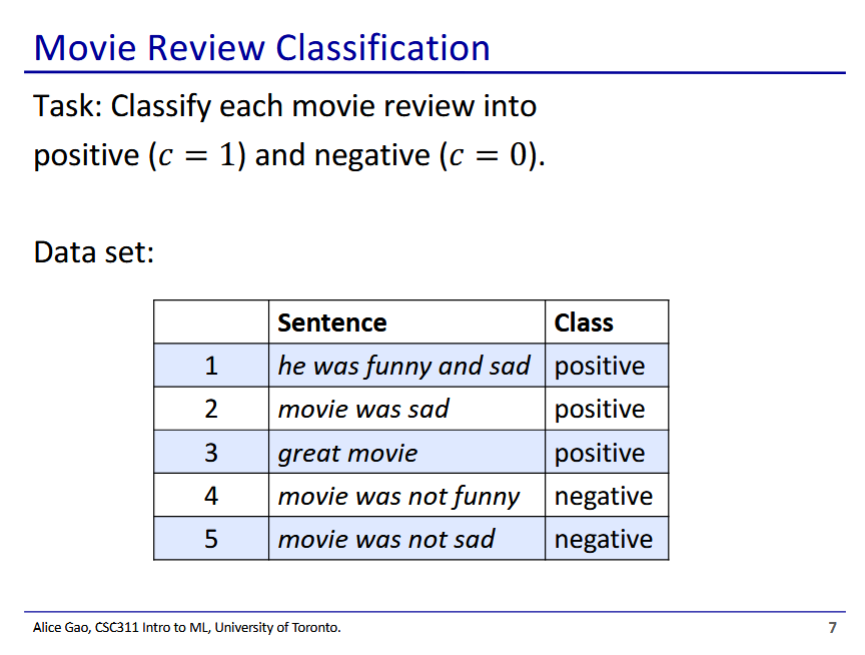




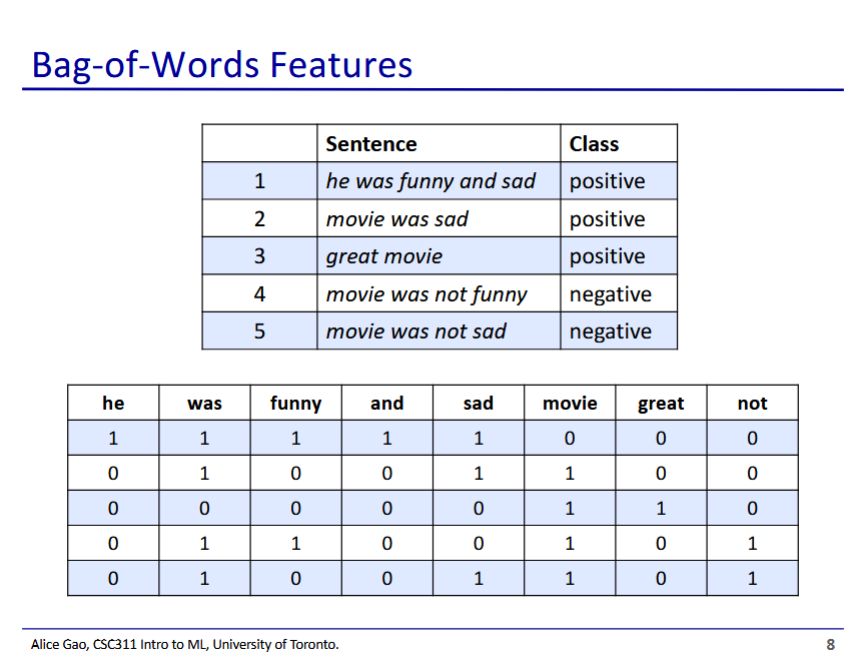


* Discriminative classifier
  + All the models we have learned so far are discriminative classifiers
  + Given an input with specific features, what class does it fall into?
    - Learns
* Generative classifier
  + Bayes is an example of generative classifiers
    - Another example are generative AI models (ChatGPT)
  + Tries to learn what each class looks like, then uses Bayes’ rule to make predictions
    - Learns
    - Thus tries to look at the underlying way that the data was generated
  + Generative classifiers seem to do more work than necessary
    - Needs to first learn , then use Bayes’ rule to calculate

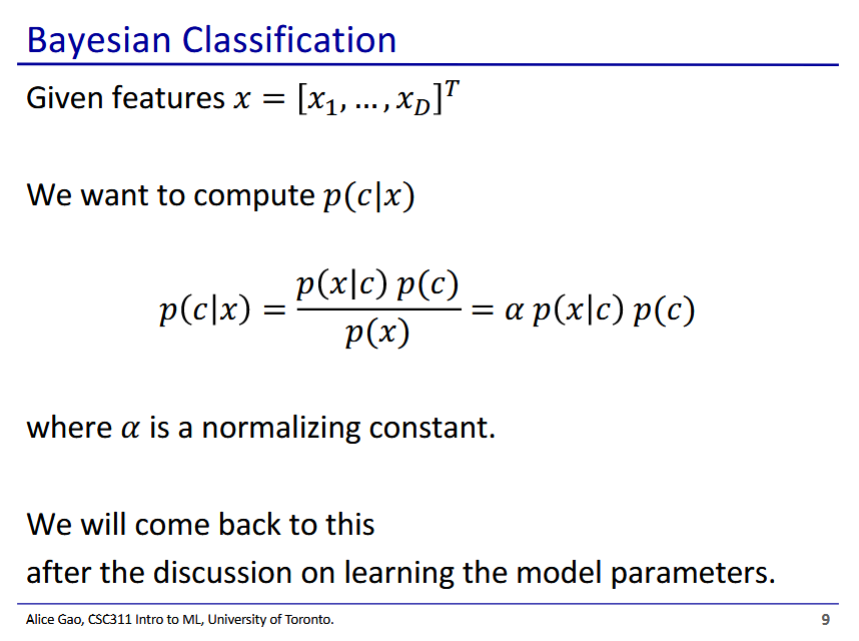




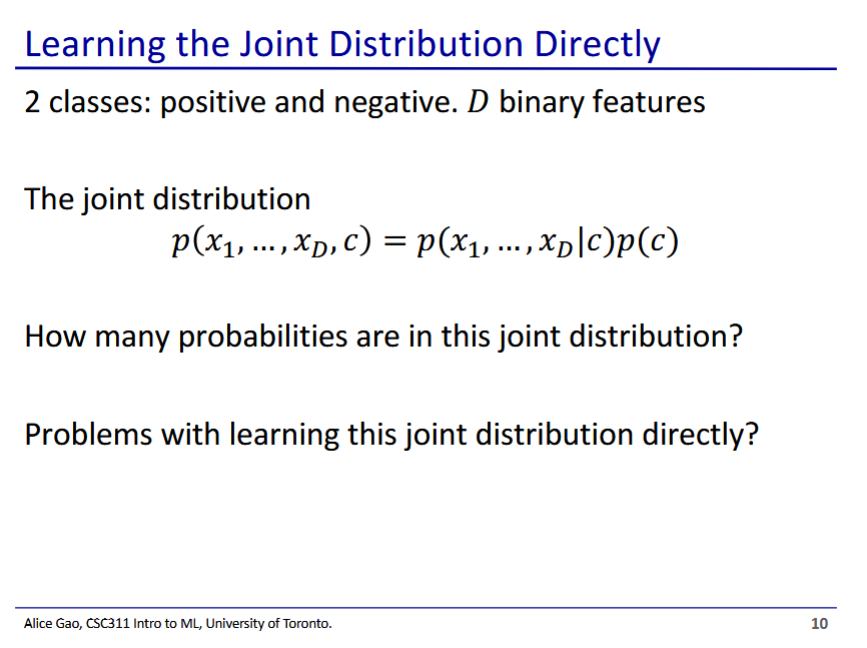
* Naive bayes is often used for text sentiment analysis



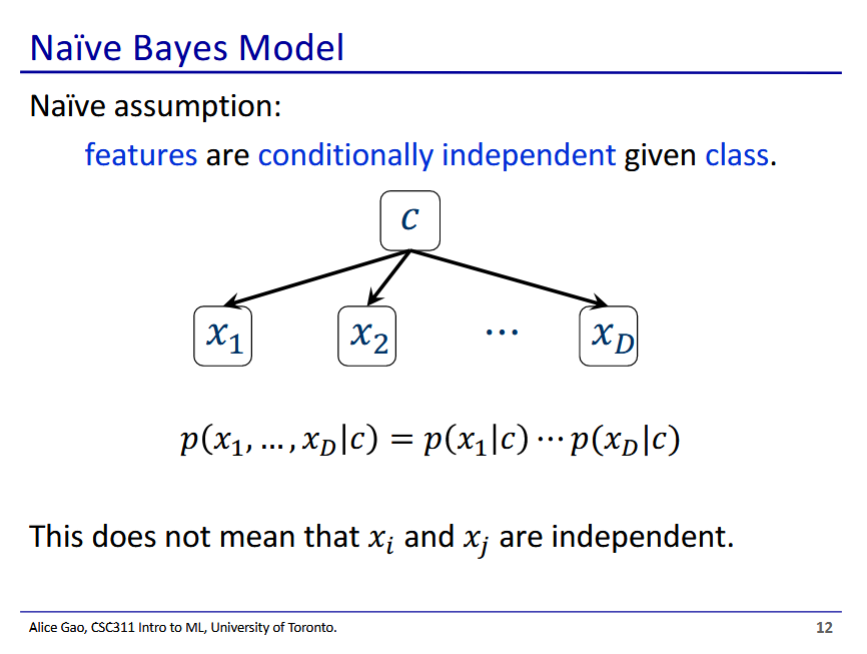
* We make each word a feature
  + Feature is 1 if the word appears in the sentence, and 0 if it does not appear



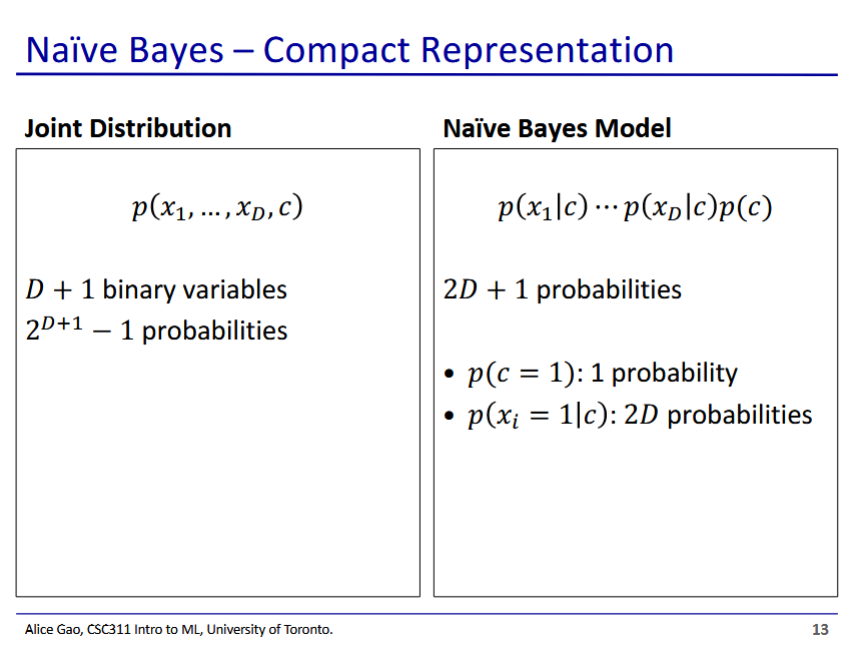
* We use Bayes’ rule to calculate the probability that an input belongs to a class given its features
* Bayesian Classif



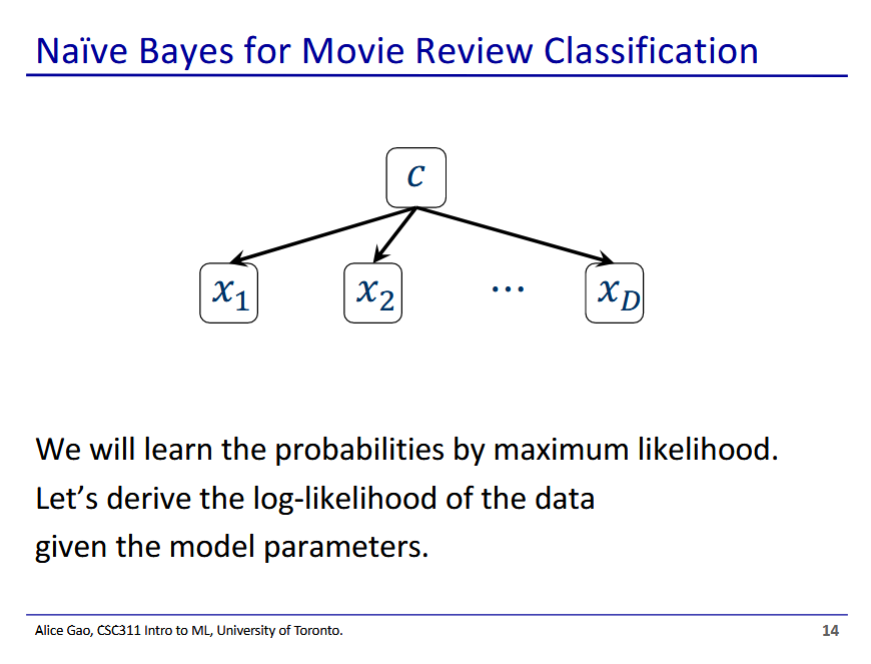
* This slide shows trying to learn the distribution directly
  + How many probabilities are in this joint distribution?
    - We have D+1 random variables, thus we have probabilities
  + Problems with learning this joint distribution directly?
    - We need to learn an exponential number of probabilities
    - Very computationally expensive
    - We need a very large number of data points to prevent overfitting
* Why do we use naive bayes?
  + Naive bayes gives us a more compact representation

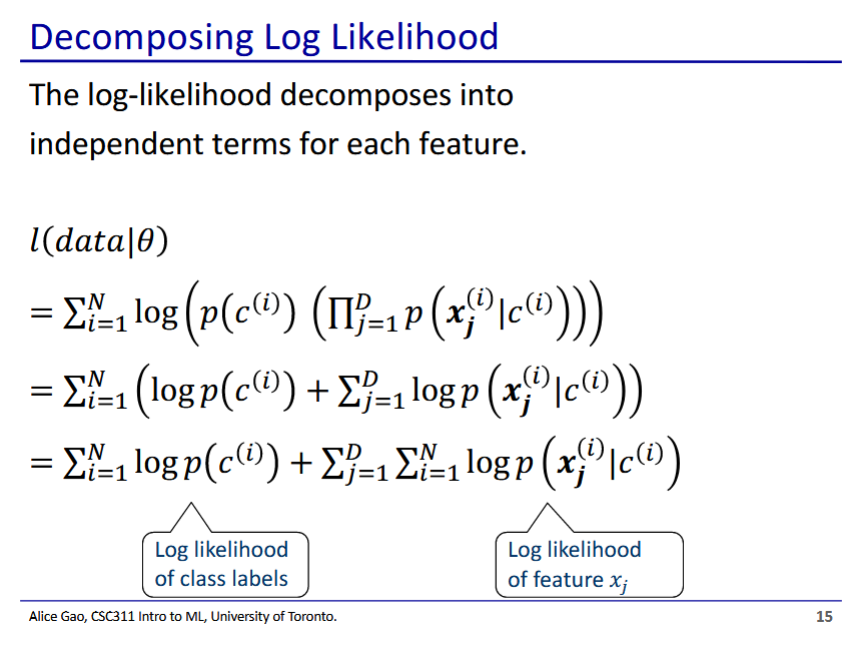


* Naive bayes
  + Makes the naive assumption that all the features are **conditionally independent** given the class
    - In a given class c, the probability that a word appears is independent to the probability that a different word appears
    - This may not be reflective of the actual probability
* The naive assumption simplifies computation
  + Allows us to decompose the term
* Note: conditionally independent is different than independent

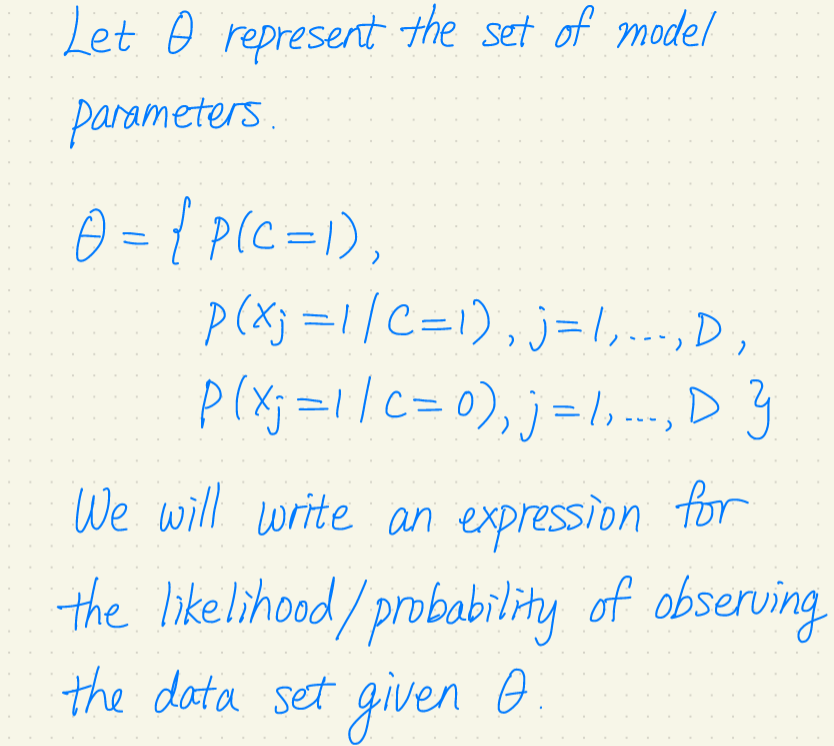


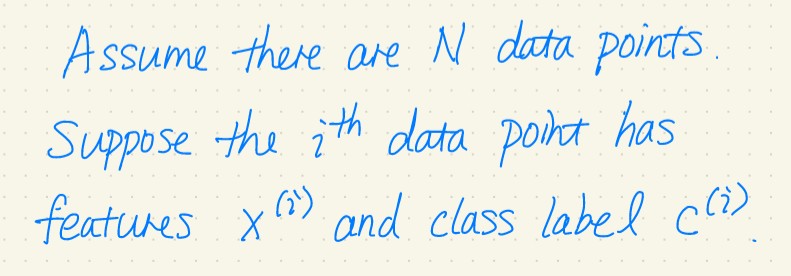
* We have gone from an exponential number of probabilities to linear using naive bayes
  + This is much more efficient

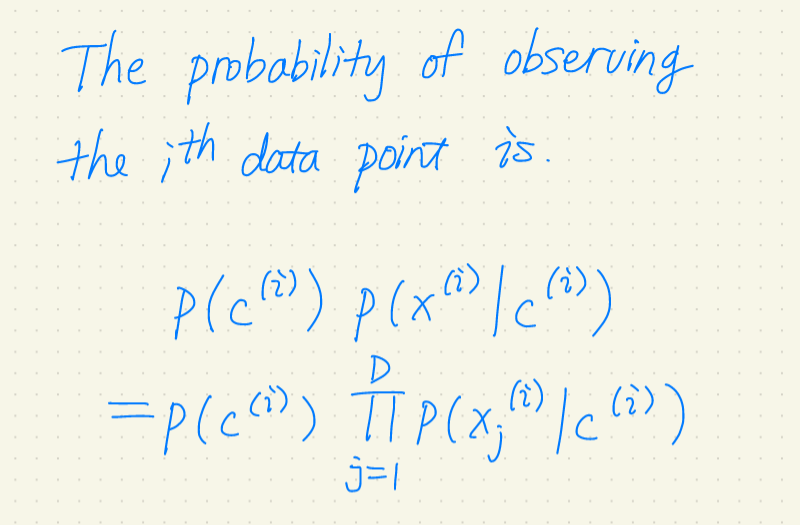




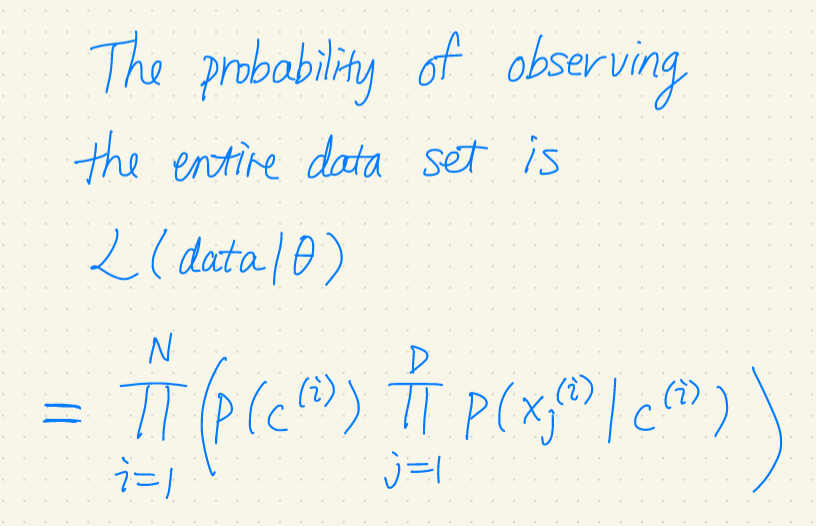
* Our log-likelihood function decomposes into the log-likelihood of the class labels and the log-likelihood of feature xj
* In this case theta represents the probabilities for each feature
  + This is the decomposed term from before, possible due to naive assumption
* This is over the whole dataset (we are summing over N)



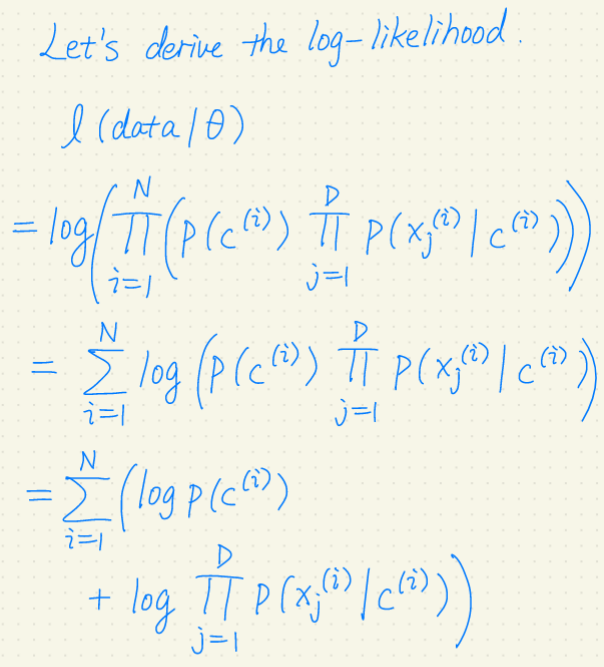


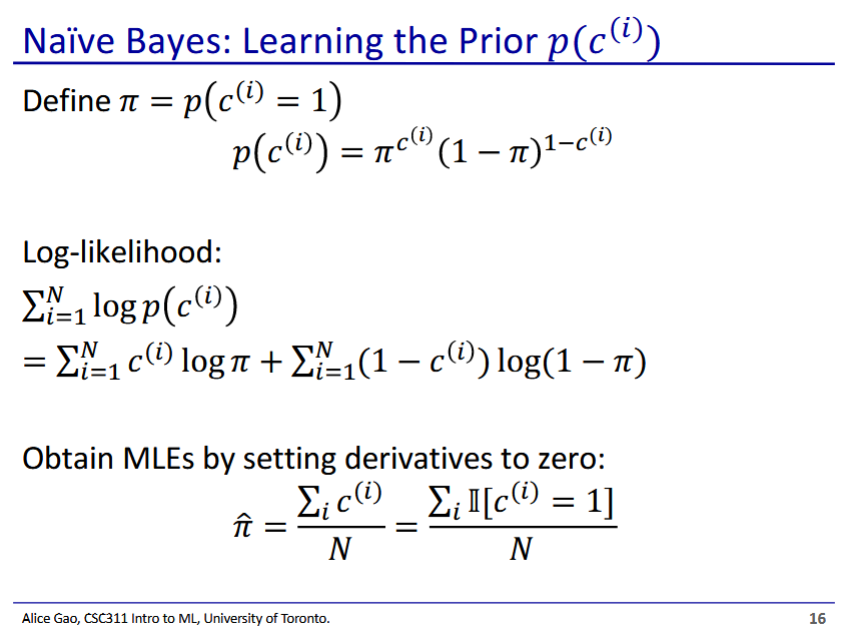


* This decomposition is only possible because of the assumption of naive bayes

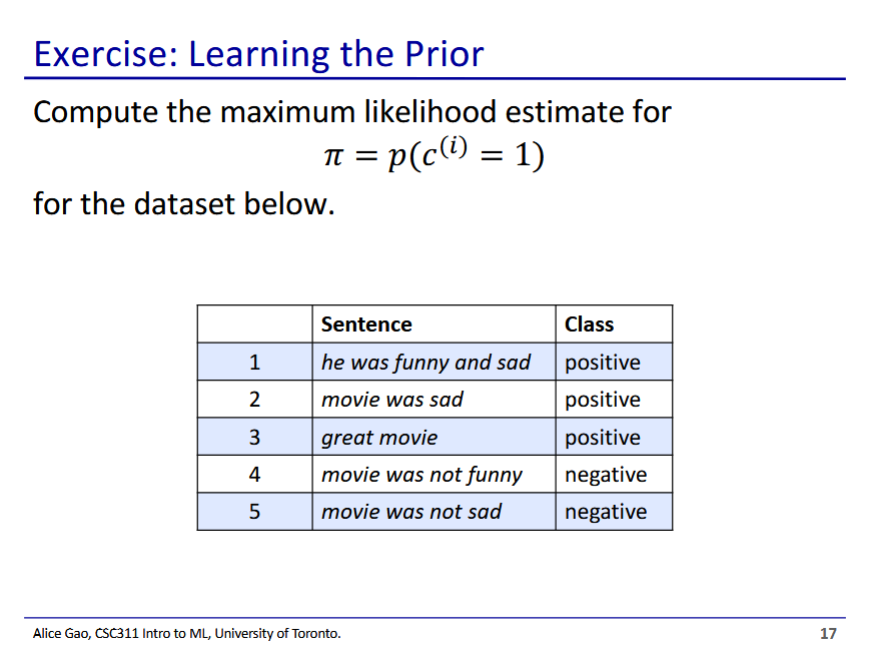


* This is the likelihood function

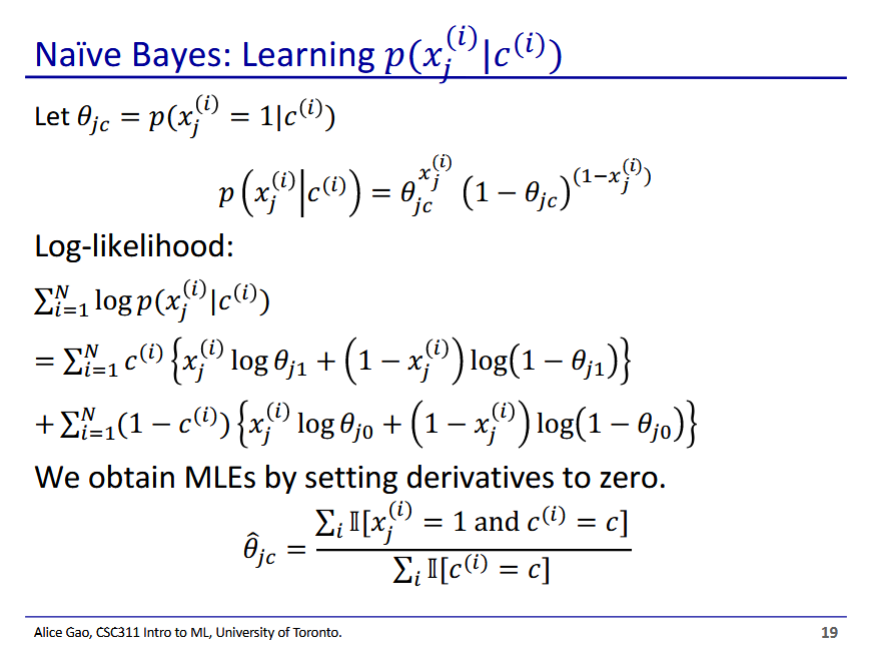




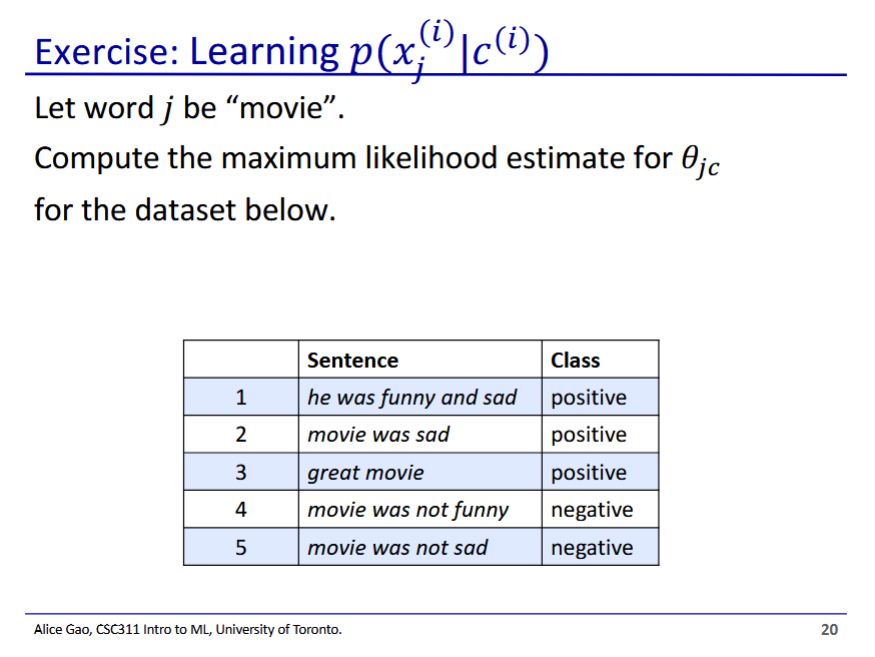
* We use MLE to learn
  + This is the probability that takes on the value that it does
  + This is the prior, and will be used in the Bayes’ theorem to classify the input
* is the probability that data point i is classified as 1
  + - Note that is only 0 or 1
* In the bottom expression, we are calculating the fraction of positive reviews in all of our data
  + The I looking thing means indicator variable



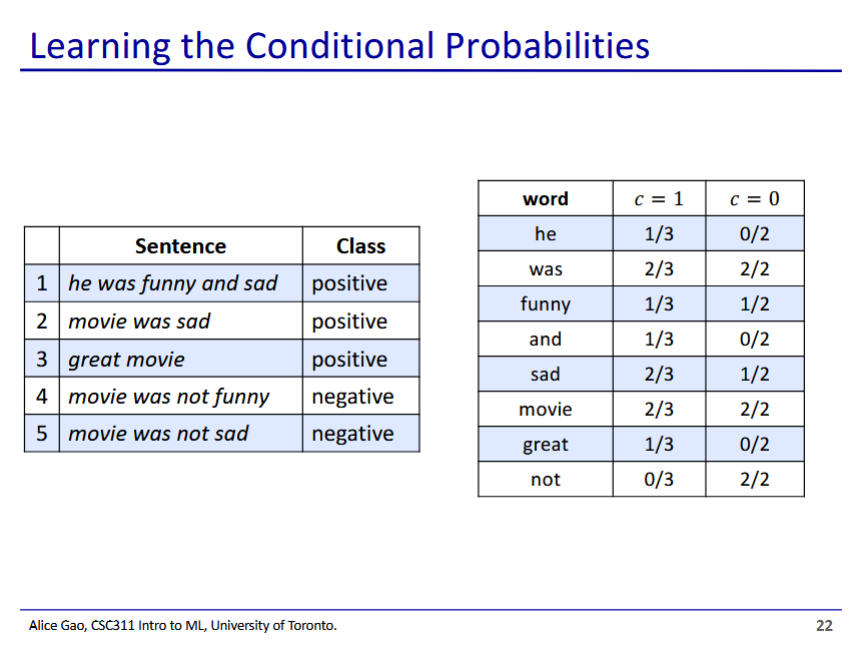
* in this case is 3/5
  + We have 3 positive reviews over our 5 total reviews



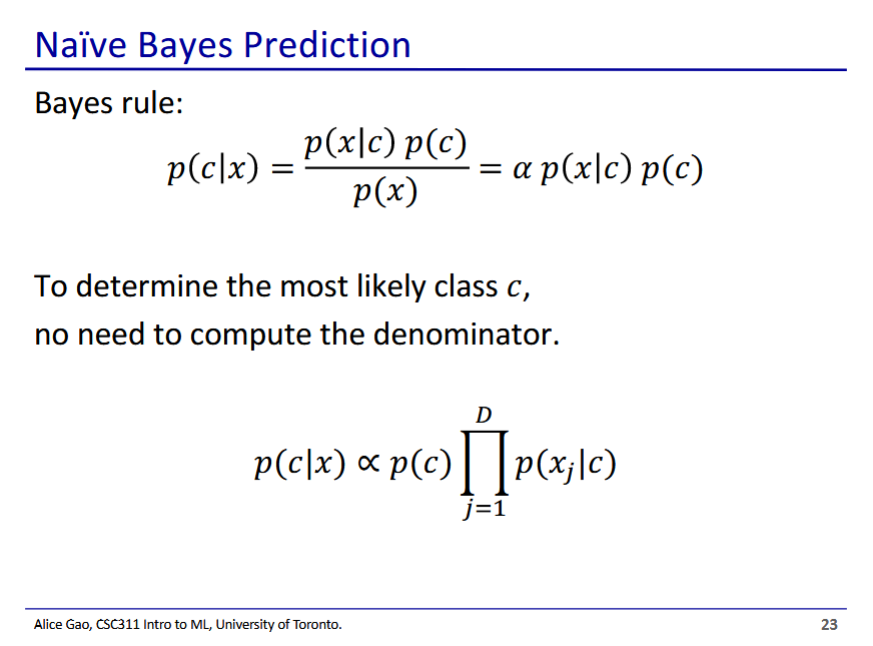
* We now learn the conditional probabilities for each component using MLE
* We have 2 split sums since can either be 1 or 0, thus needing either or to be used
  + One sum handles the case, the other handles the case
  + Otherwise is the exact same as the log-likelihood function



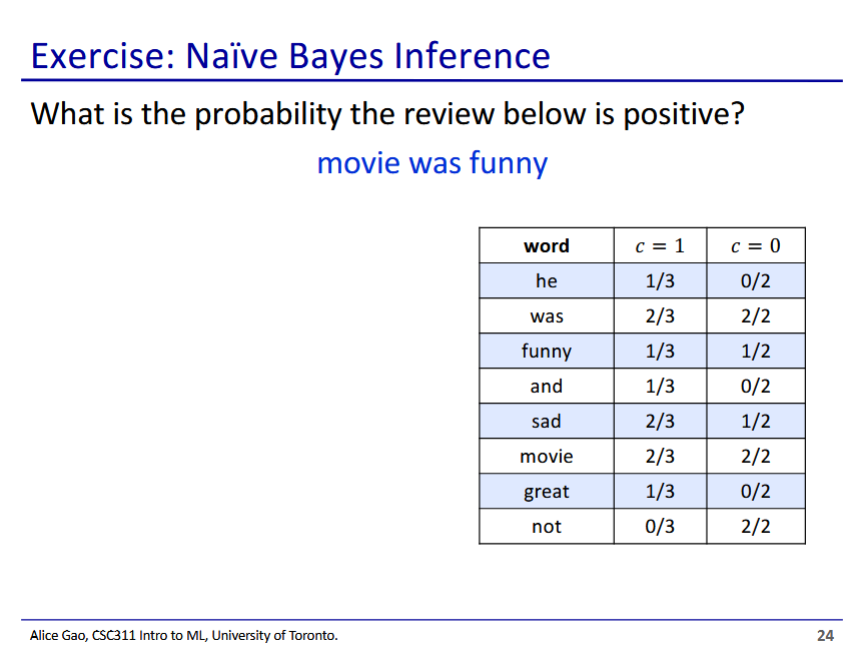
* + ⅔ positive reviews have the word movie in it
  + All negative reviews have the word movie in it



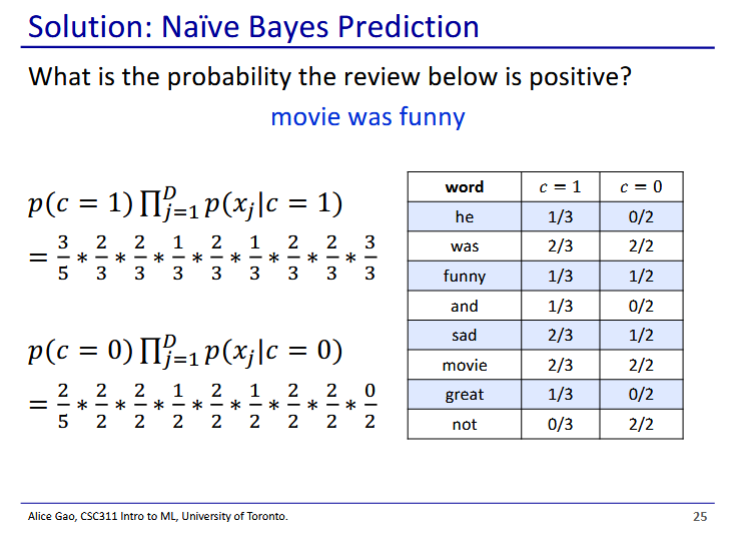
* These are the probabilities for each feature
* Now we are ready to make predictions



* If all we want to do is to determine the most likely class, we don’t need to calculate the denominator
  + The probability is proportional to the numerator, so we can just pick the largest numerator
  + is kind of a normalising constant for each of the terms



* We calculate probability for c=1 and c=0, then take which probability is larger



* Need to include all features, not just words in the sentence
  + If the word does not appear, use the inverse probability
* In this case has a larger probability