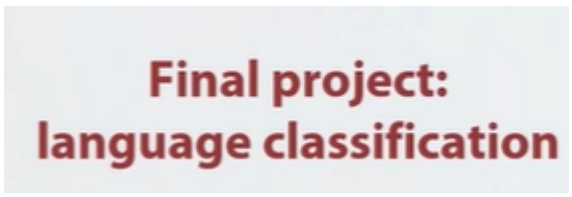
6 week Final Project lets build Bayesian classifier

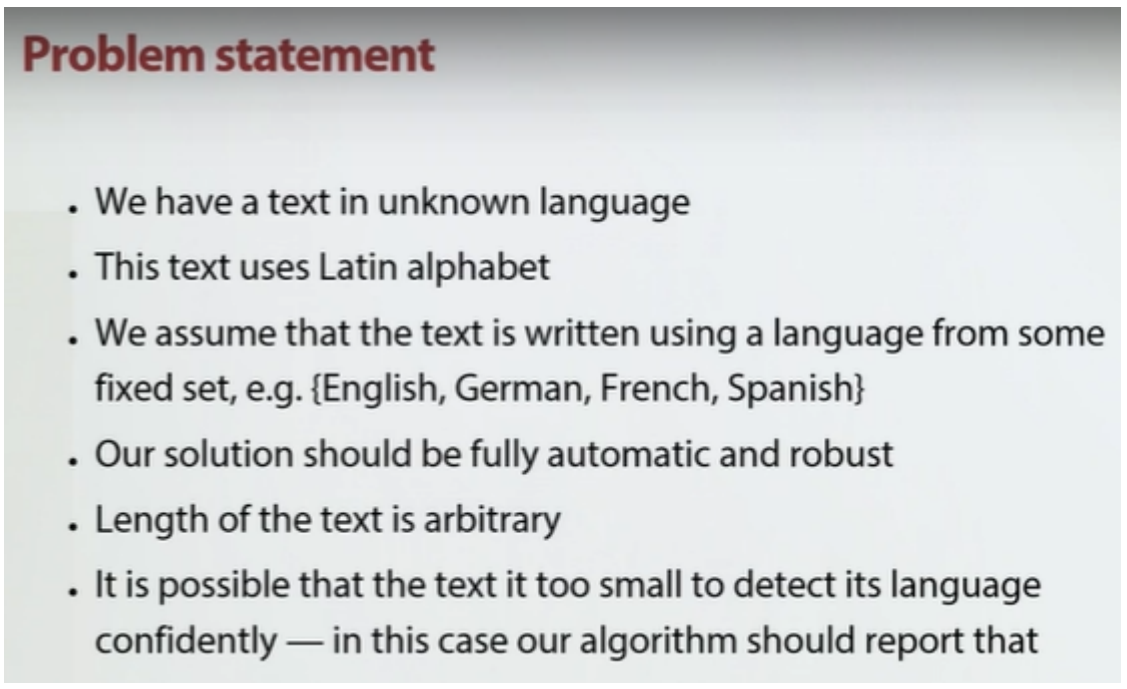
Learning Objectives:

6.1 Know definition of multinomial distribution.
6.2 Apply notions of conditional probability, random variables and their distributions to design classifier algorithm.
6.3 Apply Bayes' rule to select the most plausible hypothesis.

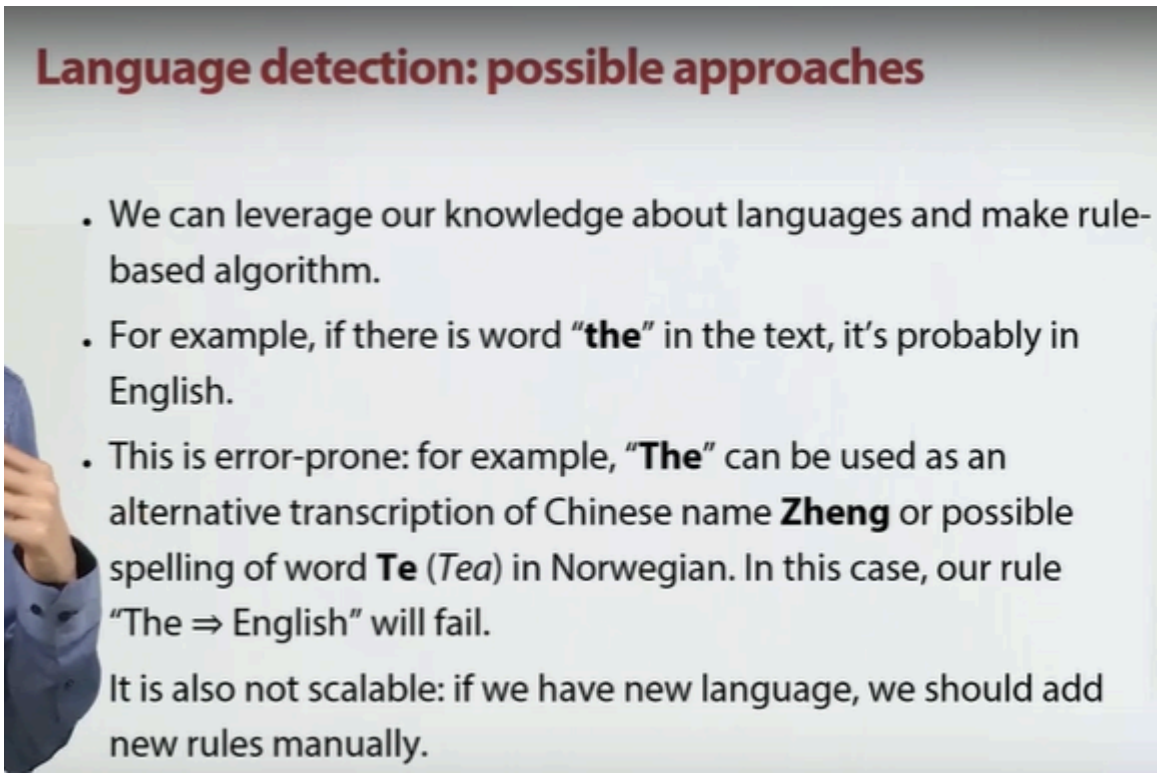
Оглавление:

Problem statement and approach





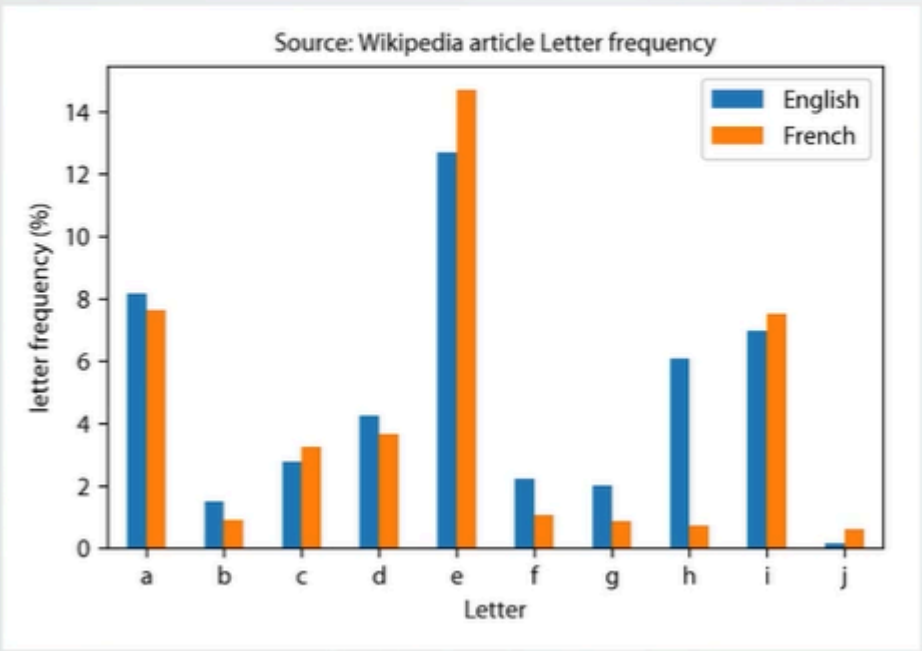
Final project will be devoted to development of algorithm of text classifications. Our problem: we have a text that is written on some unknown language for us. We have to find the algorithm which ananlyse this thext and say to which language it belongs. It there are slight mistakes in the text like typos or unusual language(slang), it should anyway detect the language correctly at least as far as it's possible.



Language detection: statistical approach

- Different languages have different statistical properties.
- The simplest characteristics is letter frequencies.

Letter frequencies



We can see that we can meet h letter more frequently in English than in French.

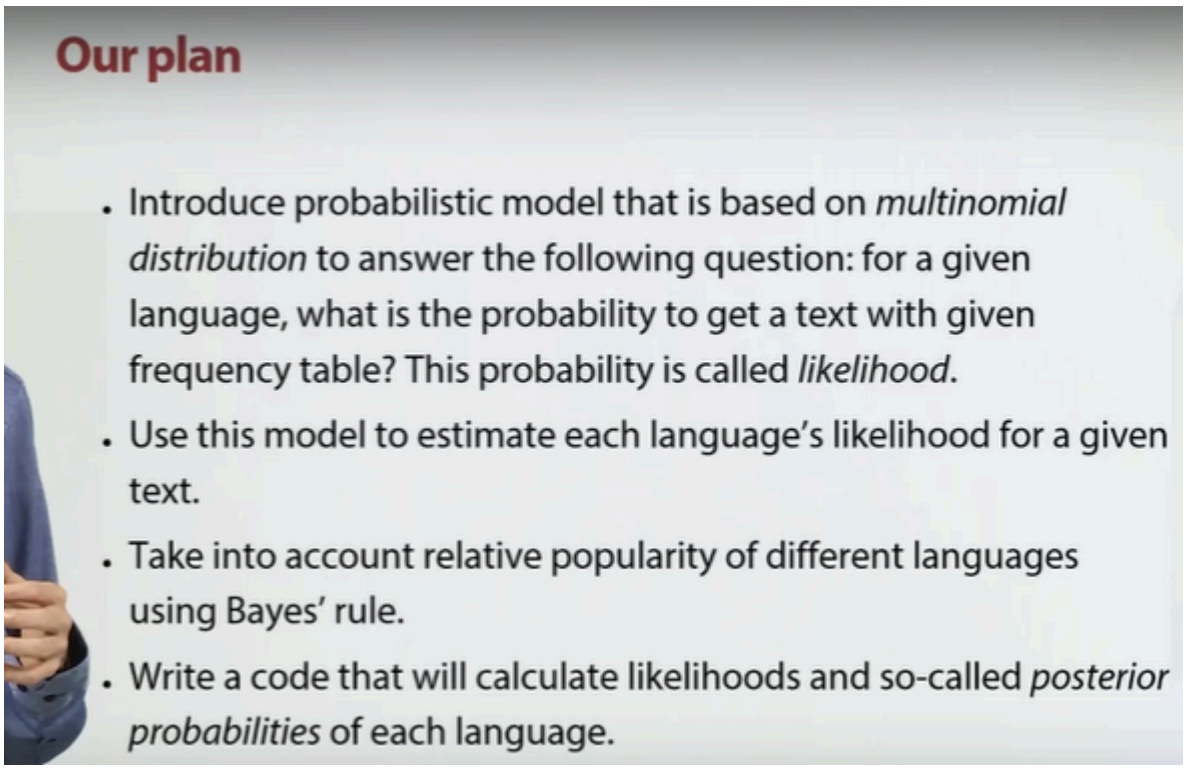
Our approach

- Find frequencies of letters in all languages that we consider using some large corpora.
- Find frequencies of letters in our text.
- Compare our frequencies with language's frequencies.

Our approach: pro and contra

- **(+)** It's fully automatic. To add a new language, we just have to add its frequency table.
- **(+)** It's robust. For example, if a text contains typos or slang, it will not affect letter frequencies too much.
- **(-)** If our text is too small, letter frequencies can become unreliable.
- **(?)** We have to define what does it mean for frequency tables to be "similar" to each other.

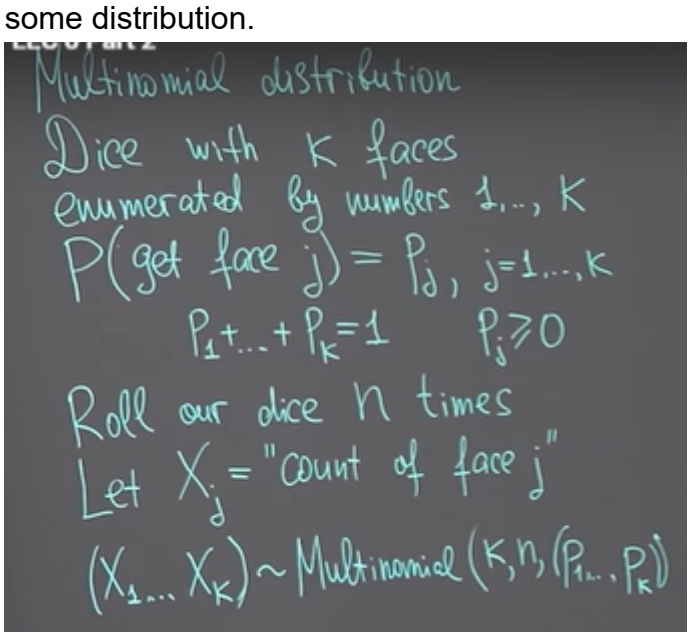
Use relative frequence which can't be affected by large lenght of text.



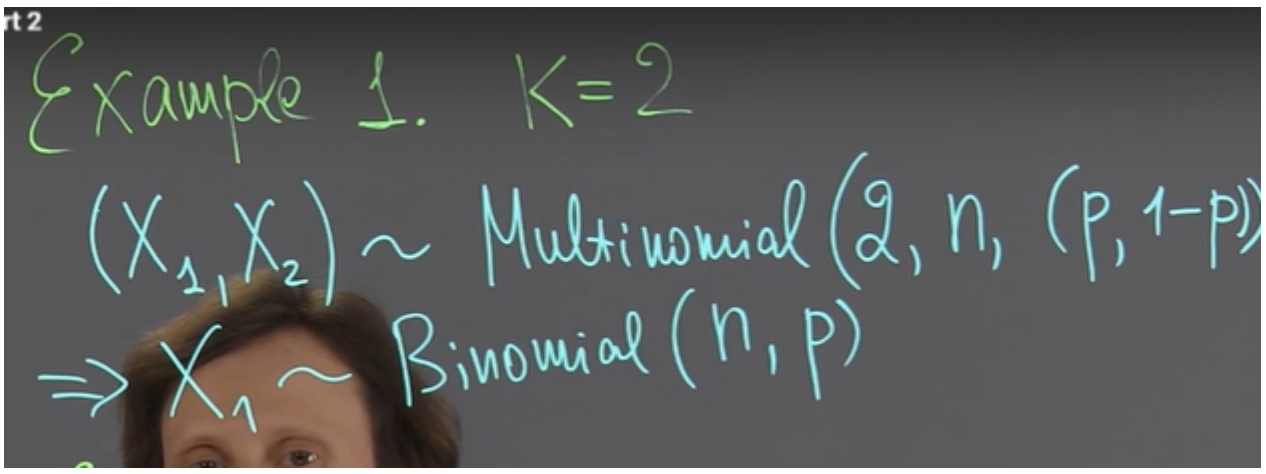
Mathematical model & multinomial distribution

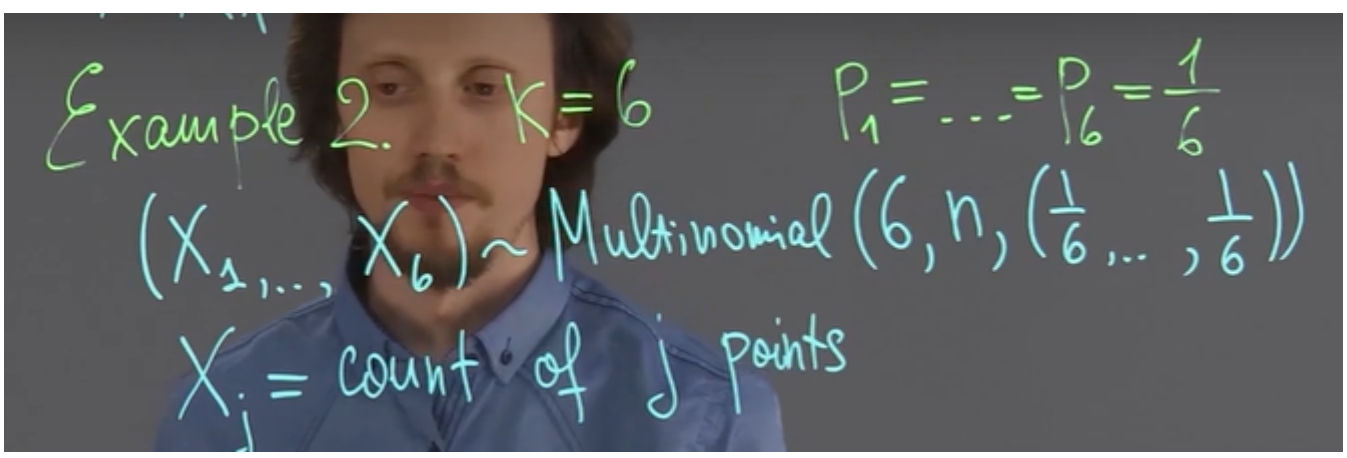
In our language detection problem, we will use frequencies of characters in the text to determine the language of the text. To make our reasoning regorous, we have to consider some mathematical model of this distribution. Under some assumption this distribution can be modelled by so called multinomial distribution. Let it discuss now:

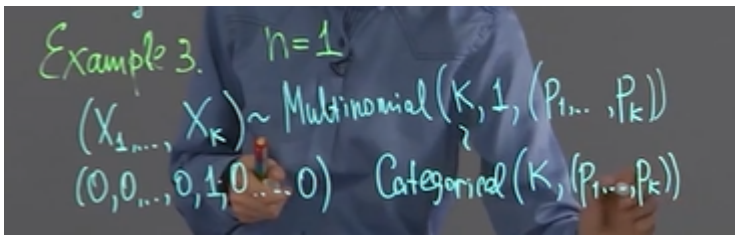
In multinomial distribution, we assume that we have a kind dice, but not actual dice, but some kind of generalized dice with K sides or K faces. And we assume that this dice can be non-symmetric. Probabilities of each face can be not equal to each other, but are given by some distribution.



1. In this case multinomial distribution becomes binomail distribution

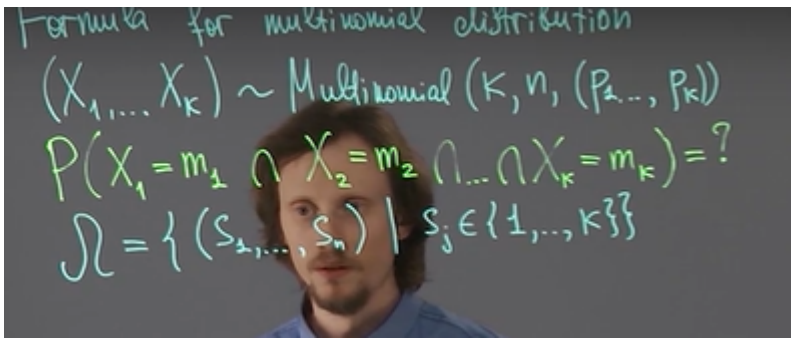




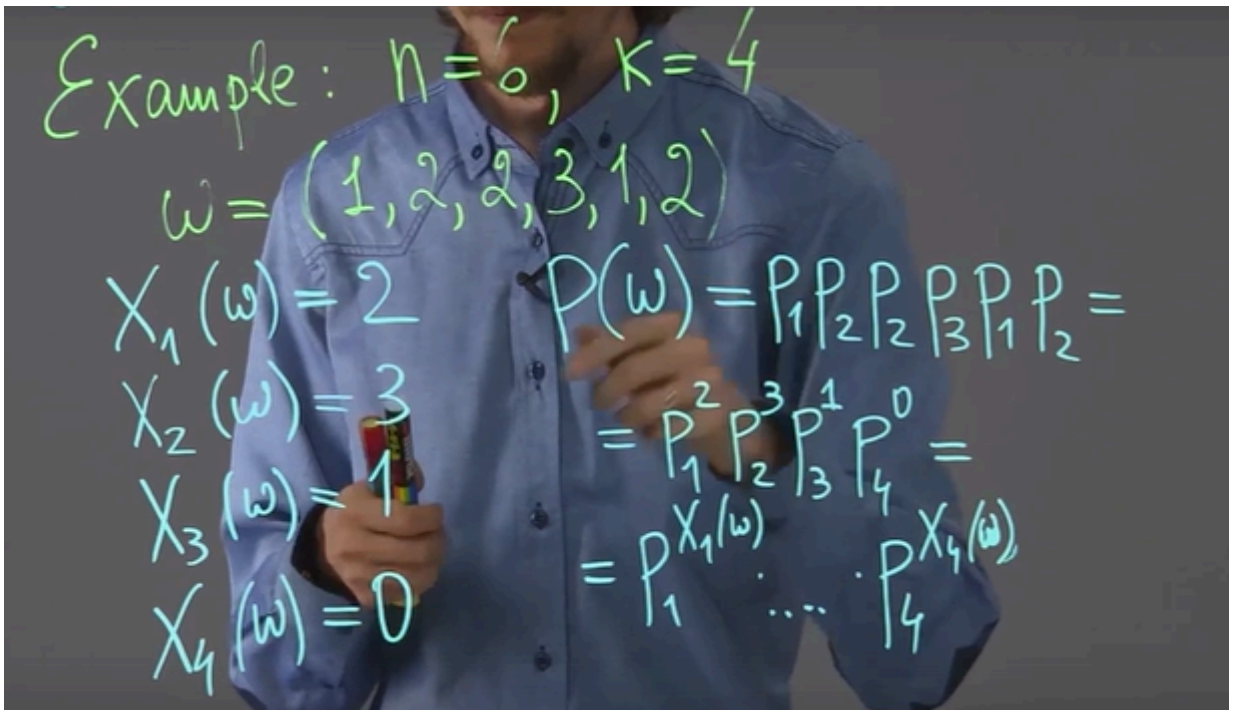


This kind vectors is used in so-called **one-hot encoding** when we have to encode a one of several categories using some numeric vector. And also this distribution called **categorical distribution.**

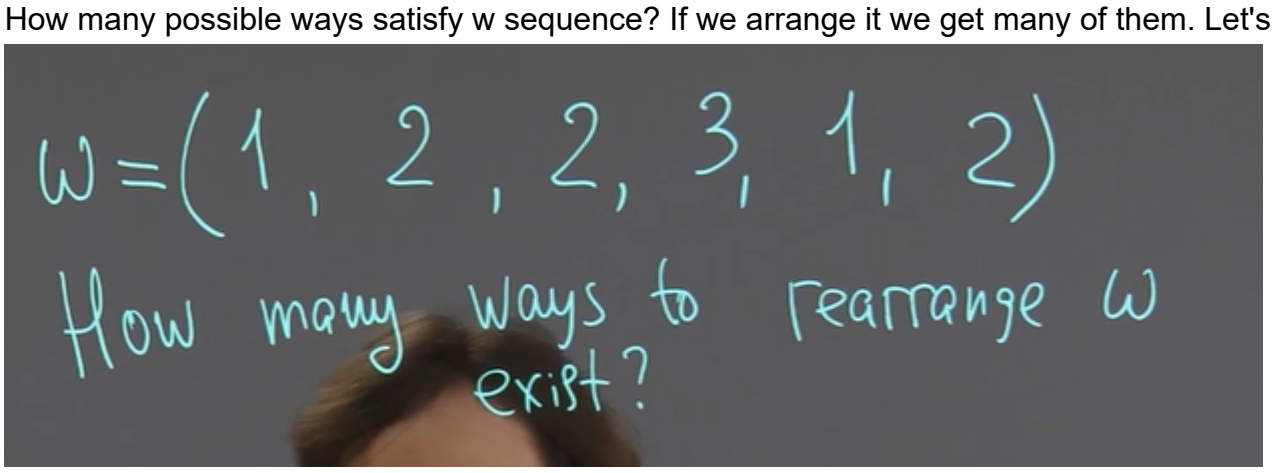
Explicit formula of multinomial distribution

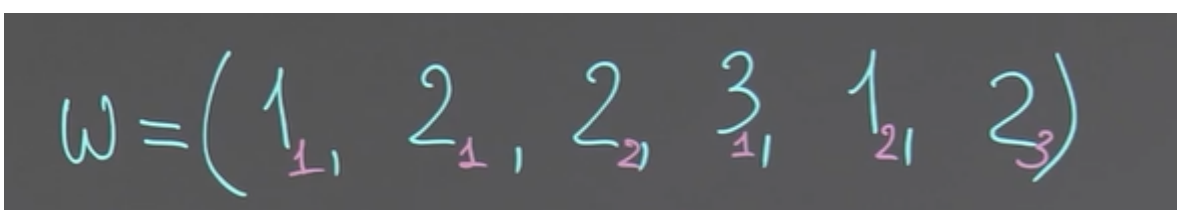


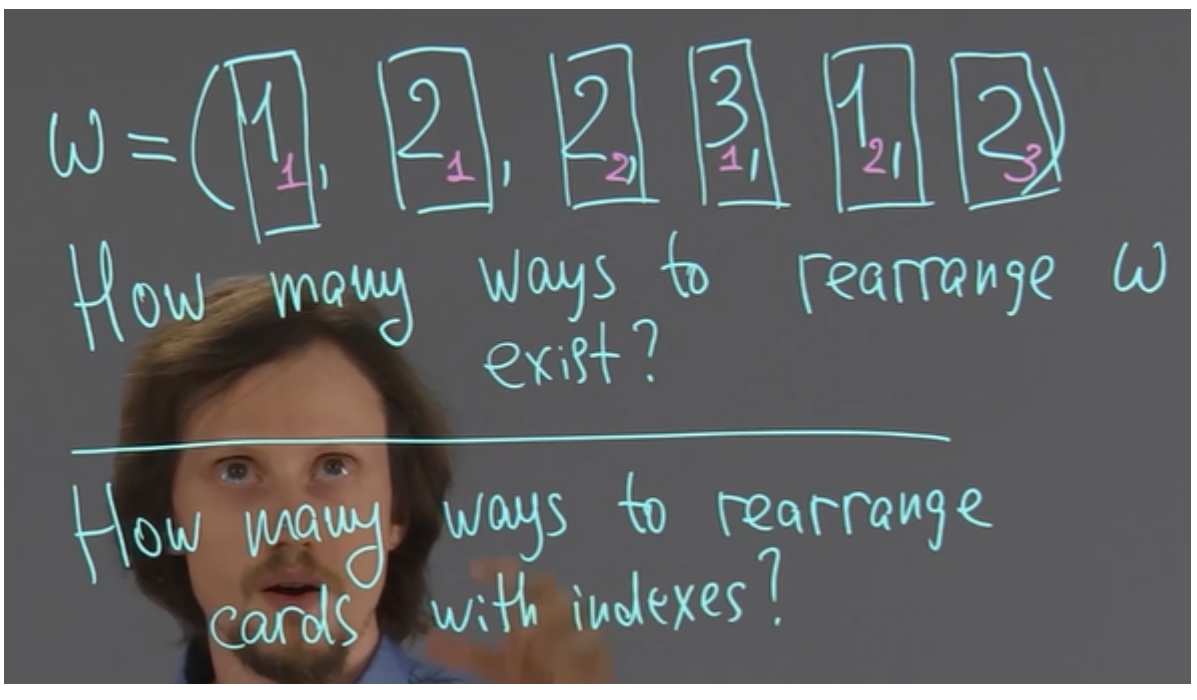
Each S is just a result of corresponding rolling dice.



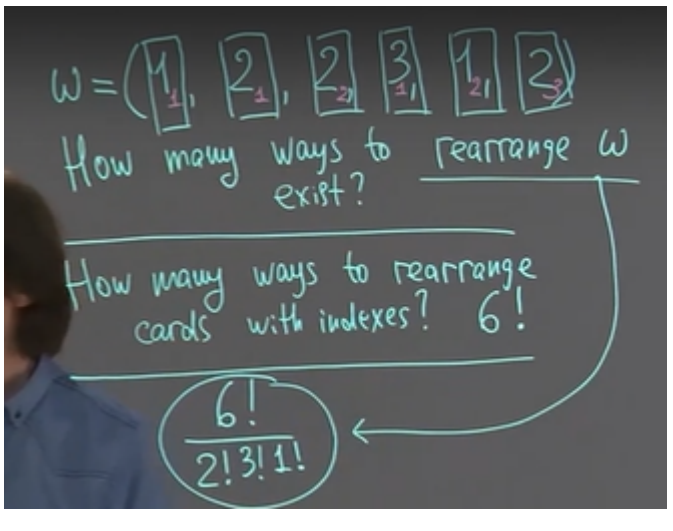
How many possible ways satisfy w sequence? If we arrange it we get many of them. Let's count by combinatorics:



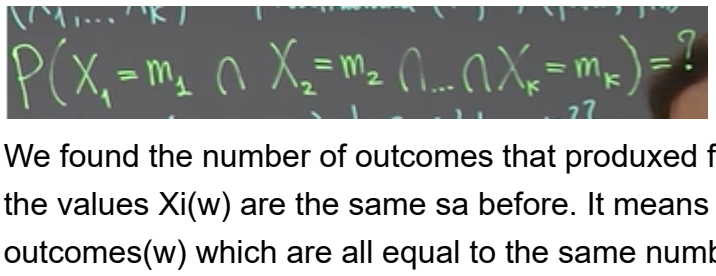




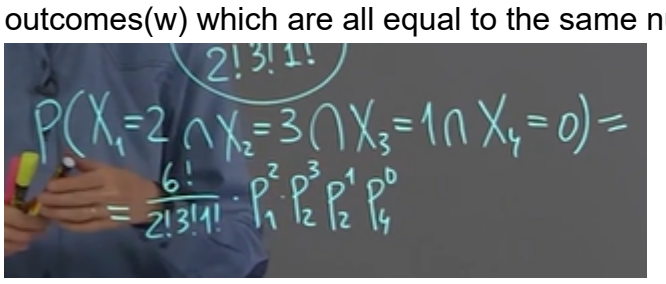
All this card are different to each other, so number of ways to rearrange them is equal to n! =6!
But no of this rearranges are different if we will ignore indexes. Now let us try to remove some indexes and check how will affect this number(6!).
We can 1 rearrange 2 times = 2!
We can 2 rearrange 3 times = 3!
We can 3 rearrange 3 times = 1!



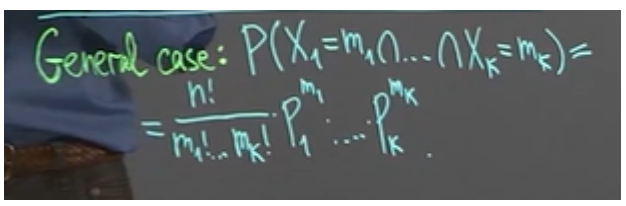
Let's return to original question:



We found the number of outcomes that produxed from this outome w by rearranging its elements. And we seen that for each such outcome the values Xi(w) are the same sa before. It means that to find thr probability that we need, we just have to multiply the probability of all such outcomes(w) which are all equal to the same number p by the number of these outcomes.



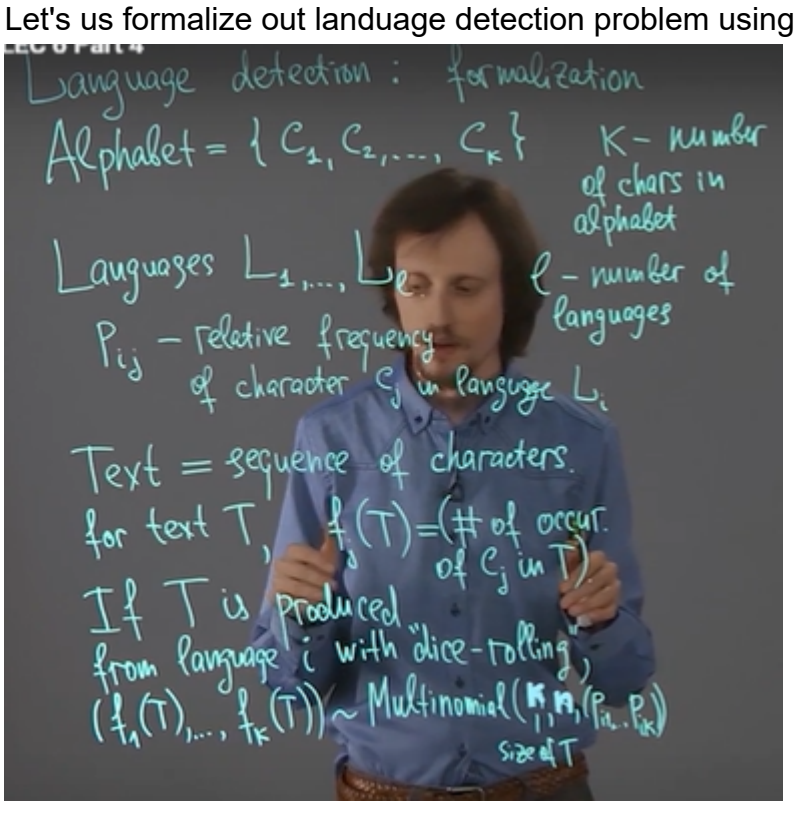
That's is universal formula



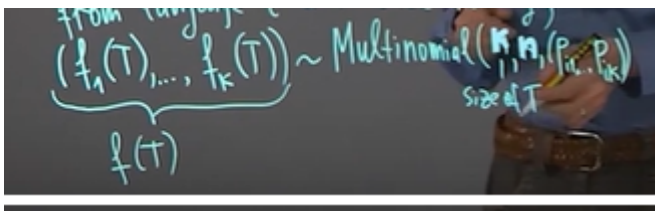
This formula gives us multinomial distribution in general case.

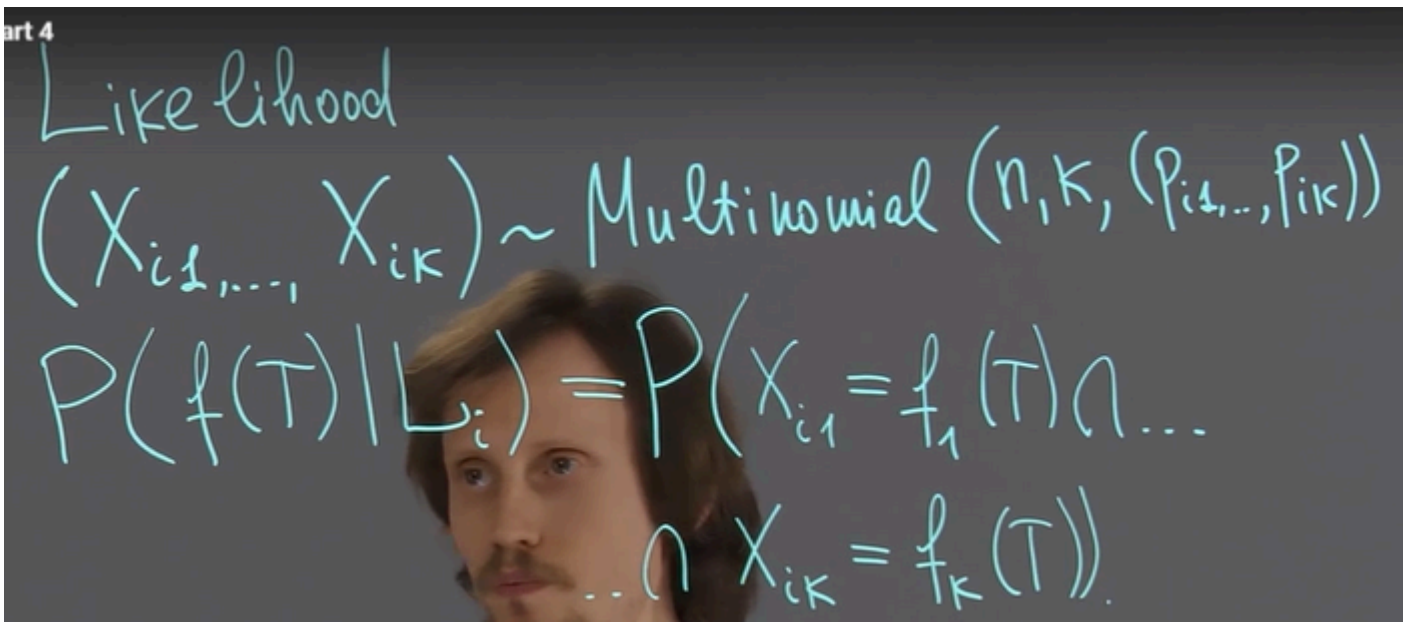
Language detection: formalization & maximum likelihood principle

Let's us formalize out landuage detection problem using multinomial distribution.

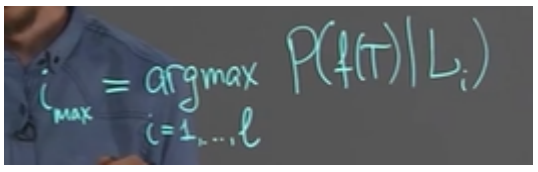


We believe that our text, which is also given and is just a sequence of characters, is just a result of random experiment that's is similar to the dice rolling with k faces and on each face we have a letter from our alphabet. We throw this dice several time, number of times is equal n - size of text, and the text is generated just by this dice. Of course, this model is not correct. No real texts are generated by this random generation. But this formula is useful for our purpose.
So now we want to answer on : Which language came according to our model?
Likelyhood is a probability to observe the data that we observed under assumption that some model for our data is valid. In this case, likelyhood is the probability that some r.v. that is multinomially distributed with some properties, have values that we get from our text.





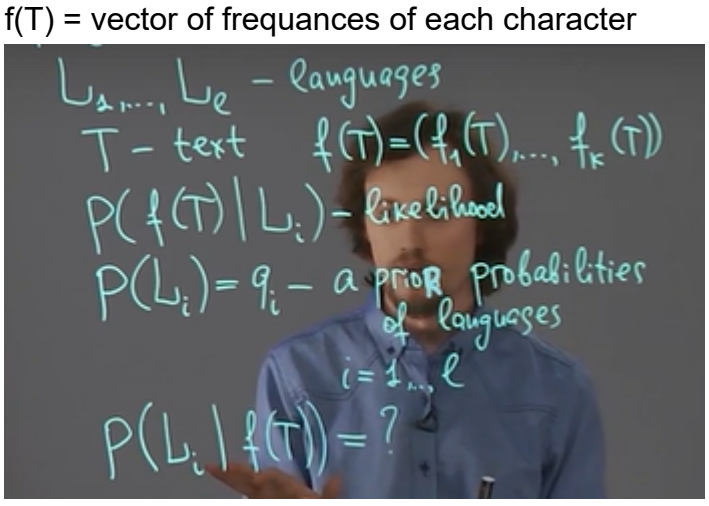
So the likelyhood is a probability to obtain the values of f(T) that we really obtained provided that our model Li takes place.
How can we use it to select the language?
In maximum likelyhood principle, we just select the most likelyhood model. So we just find i such that P(f(T), Li) is maximum.



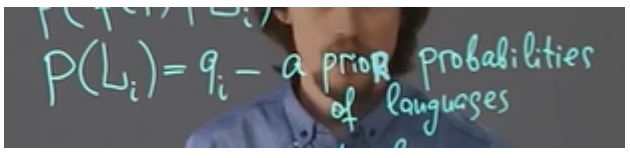
So in maximum likelihood settings, we just select the model for which likelihood of our data is maximum. However, this is not the only possible solution.
In fact, it's possible that we higher priority know that the probability of one language is less than probability of other language. Let us assume, for example, that we have some rare languages, and it's quitely inlikely to meet it in these languages. In these case, we need to have much more evidence in favor of these languages to accept them, at least from naive point of view. To formalize this approach we have to switch maximum likelihood perspective to Bayesian perspective. Now, let us discuss how to do it.

Bayesian inference and interpretation of results

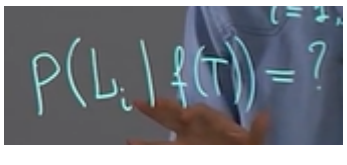
Bayesian approach allows us to take into account some prior information about the probability to get a text of some language. So let us discuss how Bayes' rule allows us to do it.
f(T) = vector of frequances of each character

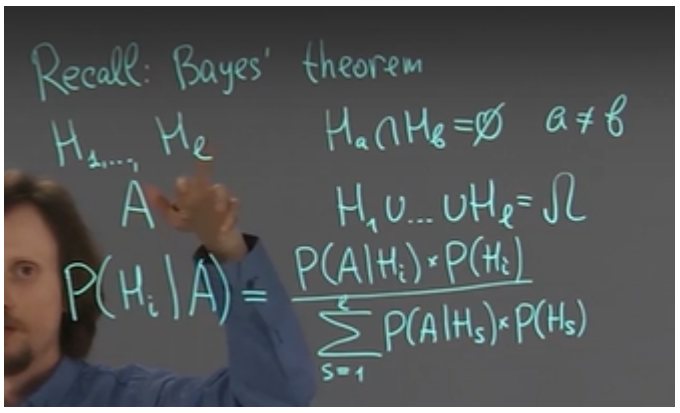


What we really want to select this language Li.
In Bayesian approach, we believe that Li is also chosen randomly before we start generating our text. So we make our modek a little bit more complicated by adding one step to it. We believe, that first we select the language and then we use this language to generate out text (f(T)) and we some a prior distribution on the set of languages.

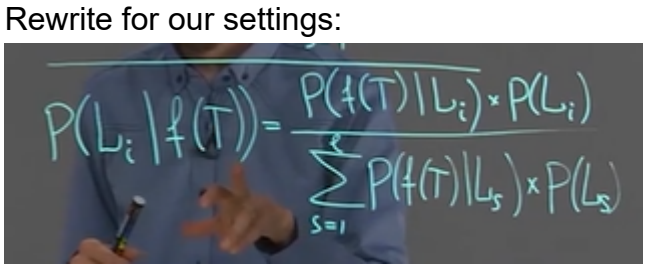


What do we want to estimate in Bayesian approach is a probability that our text belongs to some language after we looked at this text and calculated P(Li) and took into account likelihood.

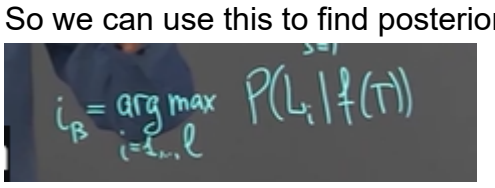




Rewrite for our settings:



So we can use this to find posteriori probability that we need to select the language with highest probabilirty.



So if you want to choose select the language which maximizes P(Li | f(T)), then you can ignore the denominator and compare only numerators, becayse for all values P(Li | f(T)) we have the same denominator and it does not depend on i. But if you want this full probability P(Li | f(T)) you have to take into account this denominator.
You can intepreter P(Li | f(T)) as a measure of how certain algorithm, how confident it's in its prediction. For example , if you see that the probability of even the best language is not so high, it means that model (your algorithm) isn't too much confident about this answer and probably you have to investigate it by different tools.

SGA: Final proof

The log-likelihood function

$$L((f_1, \dots, f_k), (p_1, \dots, p_k)) = \sum_{i=1}^{k} f_i \log p_i$$

that you (hopefully) obtained in final project is also know as cross entropy. It is extremely popular in machine learning, namely, in classification problems. It allows you to measure how good probability distribution $(p_1, \dots, p_k)$ fits the actual absolute frequencies obtained from the data $(f_1, \dots, f_k)$. Assume that frequencies $(f_1, \dots, f_k)$ are fixed. What is the best distribution $(p_1, \dots, p_k)$ from likelihood's perspective? Intuitively, it seems that we have to put relative frequencies

$$r_i = \frac{f_i}{\sum_{j=1}^{k} f_j}$$

as p_i to get best fit. In fact, it is true. To prove it, let us use Jensen's inequality for logarithms. It is stated as follows:

For any values $\alpha_1, \dots, \alpha_k$, such that $\sum_{j=1}^{k} \alpha_j = 1$ and $\alpha_j \geq 0$ for all $j = 1, \dots, k$, and any positive values $x_1, \dots, x_k$, the following inequality holds:

$$\log \sum_{j=1}^{k} \alpha_j x_j \geq \sum_{j=1}^{k} \alpha_j \log(x_j).$$

Use this inequality to prove that

$$\sum_{j=1}^{k} r_j \log p_j - \sum_{j=1}^{k} r_j \log r_j \leq 0,$$

then prove that to obtain maximum log-likelihood (and therefore maximum likelihood) for fixed $(f_1, \dots, f_k)$ we have to put $p_i = r_i, i = 1, \dots, k$.

Hint. Use properties of logarithm to transform left-hand part of the last inequality to right-rand part of the previous inequality.

Solution:

We need to prove inequality:

$$\sum_{j}^{k} r_j \log p_j - \sum_{j=1}^{k} r_j \log r_j \leq 0$$

To prove that to obtain maximum log-likelihood (and therefore maximum likelihood) for fixed $(f_1, \dots, f_k)$ we have to put $p_i = r_i$, $i = 1, \dots, k$.
We can Jensen's inequality to prove it.
For any values $\Sigma a_i = 1$, $a_i > 0$ and for all $i = 1, \dots, k$ any positive values $x_1, \dots, x_k$ the inequality looks like:

$$\log \sum_{j=1}^{k} a_j x_j \geq \sum_{j=1}^{k} a_j \log(x_j)$$

Let's apply our inequlity to our log likelihood function:

$$L((f_1, \dots, f_k), (p_1, \dots, p_k)) = \sum_{i=1}^{k} f_i \log p_i$$

So, we get:

$$\sum_{i=1}^{k} f_i \log(p_i) \geq \sum_{i=1}^{k} f_i \log(r_i)$$

We need to prove inequality:

$$\sum_{j}^{k} r_j \log p_j - \sum_{j=1}^{k} r_j \log r_j \leq 0$$

To prove that to obtain maximum log-likelihood (and therefore maximum likelihood) for fixed $(f_1, \dots, f_k)$ we have to put $p_i = r_i$, $i = 1, \dots, k$.
We can Jensen's inequality to prove it.
For any values $\Sigma a_i = 1$, $a_i > 0$ and for all $i = 1, \dots, k$ any positive values $x_1, \dots, x_k$ the inequality looks like:

$$\log \sum_{j=1}^{k} a_j x_j \geq \sum_{j=1}^{k} a_j \log(x_j)$$

Let's apply our inequlity to our log likelihood function:

$$L((f_1, \dots, f_k), (p_1, \dots, p_k)) = \sum_{i=1}^{k} f_i \log p_i$$

So, we get:

$$\sum_{i=1}^{k} f_i \log(p_i) \geq \sum_{i=1}^{k} f_i \log(r_i)$$

We need to show that $\sum_{i=1}^{k} f_i \log(r_i) \leq 0$. In order to prove that obtain maximum likelihood for fixed $(p_i = r_i, i = 1, \ldots, k)$.

Let's simplify:

$$\sum f_i p_i \log(r_i) = \sum f_i (p_i \log(p_i))$$

Since $\sum f_i \cdot p_i = 1$:

$$\sum f_i p_i \log(r_i) - \sum f_i p_i \log(p_i) = \sum f_i p_i \log\left(\frac{r_i}{p_i}\right)$$
$$\sum f_i p_i \log(r_i) \leq \sum f_i p_i \log(p_i)$$
$$\sum f_i p_i \log(r_i) - \sum f_i p_i \log(p_i) \geq 0$$
$$\sum f_i p_i \log\left(\frac{r_i}{p_i}\right) \geq 0$$

Since $L((f_i)(p_i))$ is the log likelihood for the given probabilities is the lokelihood function for the relative frequencies. Therefore we have shown that $\sum_{i=1}^{k} f_i p_i \log(r_i) \leq 0$. This means that in order to maximaze the log likelihood function for the fixed frequency $(\underbrace{f_1, \ldots, f_k}_{f_i})$ we need to set $p_i = r_i$.