Lecture 3: Probabilistic models

Helen Yannakoudakis and Oana Cocarascu

Department of Informatics
King's College London

(Version 2.2)

Today

- Introduction
- Inductive Learning
- Probabilistic models 1
- Probabilistic models 2
- Kernel Methods
- Neural Networks
- Evolutionary Algorithms
- Reinforcement Learning 1
- Reinforcement Learning 2
- Learning from Demonstration

KING'S College LONDON

Today

- Bayes theorem.
- Naive Bayes classifier.
- Guassian mixture models.
- EM algorithm.
- K-means clustering.

Probabilistic methods

- We have already come across the idea of machine learning as form of probabilistic inference.
- Models of the following form.
- We can think of classification as:

$$p(y_i|\mathbf{x}_i, \mathcal{D})$$

- "What is the probability that example x_i is in class y_i given training data $\mathcal{D}$".
- Similarly, clustering is:

$$p(\mathbf{x}_i|\mathcal{D})$$

- "What is the probability of example x_i given data $\mathcal{D}$".

KING'S College LONDON

Probabilistic methods

- Key idea is that we don't have a hard boundary between yes and no.

(Eileen Sandá)

Advantages

- Can make probabilistic predictions.
  - Don't have to pretend there is a cliff
- Each training example has an incremental effect on the estimated probability (increase or decrease estimated probability).
  - No cliff
- Prior knowledge is combined with observed data to determine final probability.

Disadvantages

- Need initial knowledge of many probabilities.
  - When not known: estimates and assumptions.
- Computational cost can be high.

Probability basics

- Interested in some set of hypotheses.
  - Could be which class an example is in.
- Write H for all of the possible ones.
- h for a specific one.
- Want the most probable hypothesis.
- Need to calculate:

$$p(h|\mathcal{D})$$

for the various $h \in H$.

Probability basics

- Bayes theorem:

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

Probability basics

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- Write

  $$p(h)$$

  to denote the probability of hypothesis h before we have any data.
- Prior probability.
- May reflect background knowledge, or some assumptions.

Probability basics

- Example of $p(h)$
- How probable it is that a student in this class gets an A?

Probability basics

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- Similarly

$$p(\mathcal{D})$$

  denotes the prior probability that the data will be observed.
- (ie, how probable we think the data is without any knowledge about which hypothesis holds).

Probability basics

- Example of $p(\mathcal{D})$
- How probable is this set of coursework marks before we know whether the student will get an A?

Probability basics

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- Now:

$$p(\mathcal{D}|h)$$

- Probability of observing the data, given that the hypothesis is true.
- Likelihood

Probability basics

- Example of $p(\mathcal{D}|h)$
- When I am told that I'm looking at the coursework of an A student, how probable are these marks?

Probability basics

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- And, finally:

$$p(h|\mathcal{D})$$

- Probability of observing the hypothesis, given the data.
- Posterior probability.
- What we are looking for in ML.

Probability basics

- Example of $p(h|\mathcal{D})$
- Now that I have seen their coursework grades, how probable is it that this student will get an A?

Bayes theorem

- Relationship between the 4 elements:

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- $p(h|\mathcal{D})$ increases with $P(h)$ and $p(\mathcal{D}|h)$.
- $p(h|\mathcal{D})$ decreases with $p(\mathcal{D})$.

KING'S College LONDON

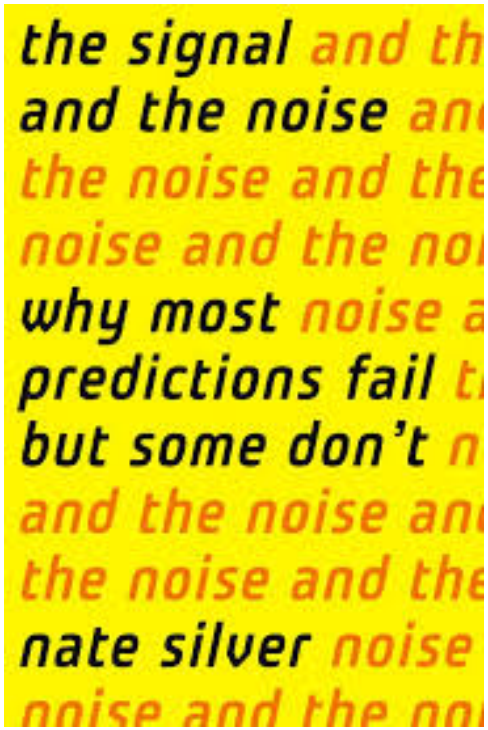
Probability basics

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- So, probability of a student getting an A once we have looked at their coursework $p(h|\mathcal{D})$.
  - Increases with the probability that they will get an A, $P(h)$.
  - Increases with the probability that this coursework is that of someone who gets a A, $p(\mathcal{D}|h)$.
  - Decreases with the probability that anyone can produce this coursework, $p(\mathcal{D})$.

The wrong underwear

- From:



The wrong underwear

You live with your partner.
You come home from a business trip to discover a strange pair of underwear in your drawer.
What should you think?

KING'S College LONDON

The wrong underwear

- Probability that some hypothesis is true, given some data.

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- data = underwear
- hypothesis = partner cheating

The wrong underwear

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

- To apply Bayes rule we need:
  - $p(\mathcal{D}|h)$
  - $p(h)$
  - $p(\mathcal{D})$

The wrong underwear

- $p(\mathcal{D}|h)$
- If they cheated, how likely is the underwear?
- Well, if they cheated, then maybe it is likely the underwear would appear.
- But also, wouldn't they be more careful?
- Say 50% chance.

The wrong underwear

- $p(h)$?
- Prior probability they cheated — before there was any evidence.
- Hard to quantify.
- Studies suggest about 4% of married partners cheat in a given year.
- Say 4%.

The wrong underwear

- $p(\mathcal{D})$
- If an event $\mathcal{D}$ can be divided into exhaustive set of disjoint subcases, then equal to sum of probabilities of subcases:

$$p(\mathcal{D}) = p(\mathcal{D}, h) + p(\mathcal{D}, \neg h)$$

- Re-written as:

$$p(\mathcal{D}) = p(\mathcal{D}|h)p(h) + p(\mathcal{D}|\neg h)(1 - p(h))$$

- Only missing $p(\mathcal{D}|\neg h)$

KING'S College LONDON

The wrong underwear

- $P(\mathcal{D}|\neg h)$
- Is there an innocent explanation?
- A friend stayed over? Gift for you?
- Say 5% chance.

The wrong underwear

- Given:

$$p(\mathcal{D}) = p(\mathcal{D}|h)p(h) + p(\mathcal{D}|\neg h)(1 - p(h))$$

we rewrite Bayes rule as:

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D}|h)p(h) + p(\mathcal{D}|\neg h)(1 - p(h))}$$

The wrong underwear

- The verdict?

$$
\begin{aligned}
p(h|\mathcal{D}) &= \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D}|h)p(h) + p(\mathcal{D}|\neg h)(1 - p(h))} \\
&= \frac{0.5 \times 0.04}{0.5 \times 0.04 + 0.05 \times (1 - 0.04)} \\
&= 0.294
\end{aligned}
$$

- Low because of the low prior.

The wrong underwear

- And if it happens again?
- Well now the prior is 0.294

$$
\begin{aligned}
p(h|\mathcal{D}) &= \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D}|h)p(h) + p(\mathcal{D}|\neg h)(1-p(h))} \\
&= \frac{0.5 \times 0.294}{0.5 \times 0.294 + 0.05 \times (1 - 0.294)} \\
&= 0.806
\end{aligned}
$$

- and the verdict is pretty clear.

Bayesian learning

- Bayes theorem: principled way to calculate the posterior of h given some data D.
- Basis for a learning algorithm that calculates the posterior of different h (not necessarily binary) and outputs the one with the highest probability.
- Maximum a posteriori (MAP) hypothesis.

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} p(h|\mathcal{D}) \\ &= \arg\max_{h \in H} \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})} \\ &= \arg\max_{h \in H} p(\mathcal{D}|h)p(h) \end{aligned}$$

- Why can we remove $p(\mathcal{D})$?

KING'S College LONDON

Bayesian learning

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} p(h|\mathcal{D}) \\
&= \arg\max_{h \in H} \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})} \\
&= \arg\max_{h \in H} p(\mathcal{D}|h)p(h)
\end{aligned}
$$

- In some cases we assume that every hypothesis $h \in H$ is equally likely a priori.
- Then we only need to consider the likelihood:

$$p(\mathcal{D}|h)$$

- In this case, we want the maximum likelihood hypothesis h_{ML}:

$$h_{ML} = \arg\max_{h \in H} p(\mathcal{D}|h)$$

KING'S College LONDON

Naive Bayes Classifier

- Practical approach to Bayesian learning.
- Simple but effective.
- Particularly resistant to overfitting (because simple).

Naive Bayes Classifier

- As a classifier, it fits the mould of:

$$p(y_i|\mathbf{x}_i, \mathcal{D})$$

where we learn an approximation of the mapping from $\mathbf{x}$ to y:

$$y = f(\mathbf{x})$$

- $f(\mathbf{x})$ takes on values from a finite set V.
  - Target value
- $\mathbf{x}_i$ is a tuple/vector of feature values:

$$\mathbf{x}_i = \langle a_1, a_2, \ldots, a_n \rangle$$

KING'S College LONDON

Naive Bayes Classifier

- In this setting, a new instance is classified by looking for:

$$v_{MAP} = \arg\max_{v_i \in V} p(v_i | a_1, a_2, \ldots, a_n)$$

which Bayes' theorem allows us to re-write as:

$$v_{MAP} = \arg\max_{v_i \in V} \frac{p(a_1, a_2, \ldots, a_n | v_i) p(v_i)}{p(a_1, a_2, \ldots, a_n)}$$

$$v_{MAP} = \arg\max_{v_i \in V} p(a_1, a_2, \ldots, a_n | v_i) p(v_i)$$

- How do we get these values?

Naive Bayes Classifier

$$v_{MAP} = \arg\max_{v_i \in V} p(a_1, a_2, \ldots, a_n | v_i) p(v_i)$$

- Estimate them based on training data!
- Easy.
- $p(v_i)$ based on the number of times v_i appears in the training data.
- Similarly for $p(a_1, a_2, \ldots, a_n | v_i)$:

  Proportion of the cases that have v_i which also have $a_1, a_2, \ldots a_n$.
- Number of possible combinations is large, and you need to see them many times to get good estimates.
- So, lots and lots of data is required.

Naive Bayes Classifier

$$v_{MAP} = \arg\max_{v_i \in V} p(a_1, a_2, \ldots, a_n | v_i) p(v_i)$$

- So we simplify (hence "naive" bayes).

1. Assume position does not matter (if applicable).
2. Assume that features are conditionally independent given the target v (strong independence assumption).

- Thus:

$$p(a_1, a_2, \ldots, a_n | v_i) = \prod_j p(a_j | v_i)$$

- $p(a_j | v_i)$: look for all the cases in which v_i occurs and then compute the proportion which have a_j.
- Need much less data to be accurate.

KING'S College LONDON

Naive Bayes Classifier

$$v_{MAP} = \arg\max_{v_i \in V} p(a_1, a_2, \ldots, a_n | v_i) p(v_i)$$

- After calculating the $p(v_i)$ and the $p(a_j|v_i)$, we can make predictions about the v_i for a given $\mathbf{x}_k = \langle a_1, a_2, \ldots, a_n \rangle$.

$$v_{NB} = \arg\max_{v_i \in V} \prod_j p(a_j|v_i) p(v_i)$$

- When the assumption about conditional independence is satisfied,

$$v_{NB} = v_{MAP}$$

  This is provable.
- When the assumption about conditional independence is not satisfied, v_{NB} is often a good solution.
  This is established by experiment.

KING'S College LONDON

(Note)

- When we introduced Bayesian learning, it was in terms of a hypothesis.
- We looked for the MAP hypothesis

$$p(h|\mathcal{D})$$

- In Naive Bayes, the hypothesis is the particular values of:

$$p(v_i)$$
$$p(a_j|v_i)$$

- The data, $\mathcal{D}$, is, as ever, all the examples from which we computed those values.

NB Example

- Anyone for tennis?

NB Example

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sun	Hot	High	Weak	No
D2	Sun	Hot	High	Strong	No
D3	Cloud	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Cloud	Cool	Normal	Strong	Yes
D8	Sun	Mild	High	Weak	No
D9	Sun	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sun	Mild	Normal	Strong	Yes
D12	Cloud	Mild	High	Strong	Yes
D13	Cloud	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

KING'S College LONDON

NB Example

- Use this data to learn a naive Bayes classifier.
- Then we can classify unseen examples, such as:

$$\langle Outlook = Sun, Temp = Cool, \\ Humidity = High, Wind = Strong \rangle$$

NB Example

- Now, normally what we would do to train the classifier is to compute all the probabilites:

$$p(v_i)$$
$$p(a_j|v_i)$$

for all v_i and a_j

$v_i \in \{yes, no\}$

$a_j \in \{Outlook = Sun, Outlook = Cloud, \ldots Wind = Weak\}$

- Then we could give an answer for any example.

NB Example

- Query:

$$\langle Outlook = Sun, Temp = Cool, \\ Humidity = High, Wind = Strong \rangle$$

- Here, since we know the query, we can cheat.
- We will just compute the probabilities we know that we need to answer the query.

NB Example

- Query:

$$\langle Outlook = Sun, Temp = Cool, \\ Humidity = High, Wind = Strong \rangle$$

- What we want to compute is:

$$v_{NB} = \arg \max_{v_i \in \{yes,no\}} p(v_i) \prod_j p(a_j|v_i)$$

$$= \arg \max_{v_i \in \{yes,no\}} p(v_i) \, (p(Outlook = Sunny|v_i) \\ \times p(Temp = Cool|v_i) \\ \times p(Humidity = High|v_i) \\ \times p(Wind = Strong|v_i))$$

(This is the equation from slide 38, with the a_j instantiated.)

NB Example

- We can calculate all the numbers we need from the training data.
- Start with:

$$p(\mathit{Tennis}? = \mathit{yes}) = ?$$
$$p(\mathit{Tennis}? = \mathit{no}) = ?$$

NB Example

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sun	Hot	High	Weak	No
D2	Sun	Hot	High	Strong	No
D3	Cloud	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Cloud	Cool	Normal	Strong	Yes
D8	Sun	Mild	High	Weak	No
D9	Sun	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sun	Mild	Normal	Strong	Yes
D12	Cloud	Mild	High	Strong	Yes
D13	Cloud	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

KING'S College LONDON

NB Example

- We can calculate all the numbers we need from the training data.
- Start with:

$$p(\textit{Tennis?} = \textit{yes}) = \frac{9}{14} = 0.64$$

$$p(\textit{Tennis?} = \textit{no}) = \frac{5}{14} = 0.36$$

- For these we count how many of all the cases have *Tennis*? = *yes* and *Tennis*? = *no*.

NB Example

- Then we can estimate the conditional probabilities.
- Query:

$$\langle Outlook = Sun, Temp = Cool, \\ Humidity = High, Wind = Strong \rangle$$

- For example:

$$p(Wind = Strong | Tennis? = yes) = ? \\ p(Wind = Strong | Tennis? = no) = ?$$

NB Example

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sun	Hot	High	Weak	No
D2	Sun	Hot	High	Strong	No
D3	Cloud	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Cloud	Cool	Normal	Strong	Yes
D8	Sun	Mild	High	Weak	No
D9	Sun	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sun	Mild	Normal	Strong	Yes
D12	Cloud	Mild	High	Strong	Yes
D13	Cloud	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

KING'S College LONDON

NB Example

- Then we can estimate the conditional probabilities.
- For example:

$$p(Wind = Strong|Tennis? = yes) = \frac{3}{9} = 0.33$$

$$p(Wind = Strong|Tennis? = no) = \frac{3}{5} = 0.36$$

- For the first of these we look at all the cases where $Tennis? = yes$ (9 cases) and look at how many of these have $Wind = Strong$.

NB Example

- We then repeat for the other values we need.

NB Example

- Now we can put these values back into the equation for v_{NB}:

$$\begin{aligned} v_{NB} &= \arg\max\,(p(yes)p(Sunny|yes)p(Cool|yes) \\ &\qquad p(High|yes)p(Strong|yes), \\ &\qquad p(no)p(Sunny|no)p(Cool|no) \\ &\qquad p(High|no)p(Strong|no)) \\ &= \arg\max(0.0053, 0.0206) \\ &= no \end{aligned}$$

- Thus the answer is *Tennis*? $= no$ for this example.

NB Example

- Normalizing will give us the probability of the answer being correct:

$$p(v_{NB}) = \frac{0.0206}{0.0206 + 0.0053}$$
$$= 0.795$$

Naive Bayes output

- Much of the time we will want to know the classification.
  Do we play tennis or not?
- The probabilities can be important, however.
  For sampling a decision based on class for example.
- If the classifier says “right” with probability 0.4 and “left” with probability 0.6, you may want to go right 4 times in 10.

Small numbers

- Note the small probabilities that we compute:

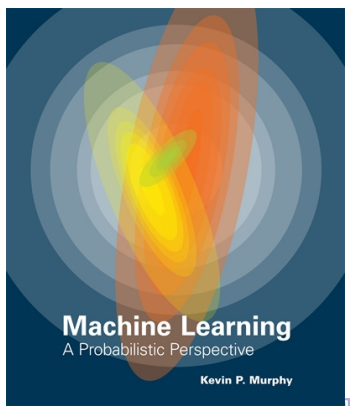
$$\langle 0.0053, 0.0206 \rangle$$

- With more features can get easily underflow errors.
- Avoid this by taking logs:

$$v_{NB} = \arg\max_{v_i \in V} p(v_i) \prod_j p(a_j|v_i)$$

$$v_{NB} = \arg\max_{v_i \in V} logP(v_i) + \sum_j logP(a_j|v_i)$$

- See Murphy, page 86.



KING'S College LONDON

Estimating probabilities

- So far we computed probabilities by just counting.
- Estimated $p(A = a|B = b)$ by:

$$\frac{n_c}{n}$$

  where:
  - n is total number of cases where $B = b$; and
  - n_c is the number of those cases where $A = a$.
- Good estimate much of the time.
- But can be poor when n_c is close to zero.
  Which can happen we have only a few samples.
- Overfitting!
- Solution is to use a better estimator.

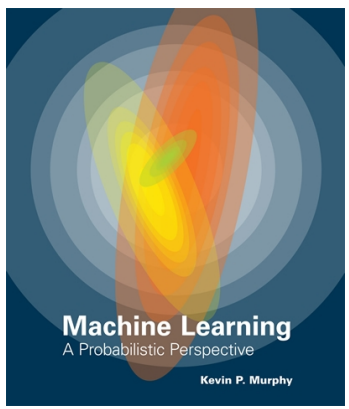
Estimating probabilities

- m-estimate:

$$\frac{n_c + m.p}{n + m}$$

- n_c and n are as before.
- p is a prior estimate (before taking any data into account).
- Typical value of p is $\frac{1}{k}$ if the feature has k values.
- m is a constant, equivalent sample size.
  - Weight given to the prior.
- Can also use other, more sophisticated priors when appropriate.

- See Murphy, pages 84–85.



KING'S College LONDON

(Last words on Naive Bayes)

- When we compute:

$$v_{NB} = \arg\max_{v_i \in V} p(v_i) \prod_j p(a_j|v_i)$$

what we are doing is comparing:

$$p(v_i|a_1, a_2, \ldots a_n) = \arg\max_{v_i \in V} p(v_i) \prod_j p(a_j|v_i)$$

but everything is relative to the data, so we are really computing:

$$p(v_i|a_1, a_2, \ldots a_n, \mathcal{D}) = \arg\max_{v_i \in V} p(v_i|\mathcal{D}) \prod_j p(a_j|v_i, \mathcal{D})$$

Last words on Naive Bayes

- In the notation of slide 37, the key step in the Naive Bayes classifier is:

$$p(a_1, a_2, \ldots, a_n | v_i) = \prod_j p(a_j | v_i)$$

  for n features.
- We computed these conditionals by *counting* instances.
- This is fine for *discrete-valued features*.
- But the model is more general.

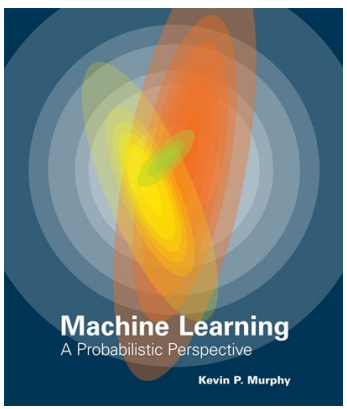
Last words on Naive Bayes

- For example, if the features are real-valued, we might model each as a Gaussian and compute:

$$p(a_1, a_2, \ldots, a_n | v_i) = \prod_j^n \mathcal{N}(a_j | \mu_{jc}, \sigma_{jc}^2)$$

  where μ_{jc} and σ_{jc}^2 are the mean and variance of feature j in instances of class $c = v_i$.

- See Murphy, pages 82–83.

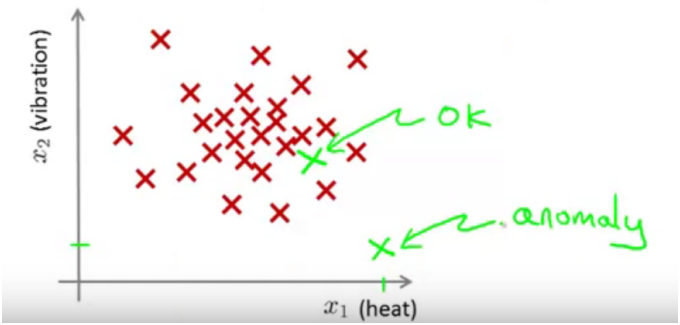


KING'S College LONDON

Unsupervised learning

$$p(\mathbf{x}'|\mathcal{D})$$

- "What is the probability of an example $\mathbf{x}'$ given data $\mathcal{D}$".
- Given a dataset D of non-anomalous instances, is a new $\mathbf{x}'$ anomalous? (eg, anomalous aircraft engine, fraud detection / detect "unusual" users).
- Need to model $p(\mathbf{x})$ using D.
- Use $p(\mathbf{x})$ to detect anomalous cases: $p(\mathbf{x}') < \textit{threshold}$.



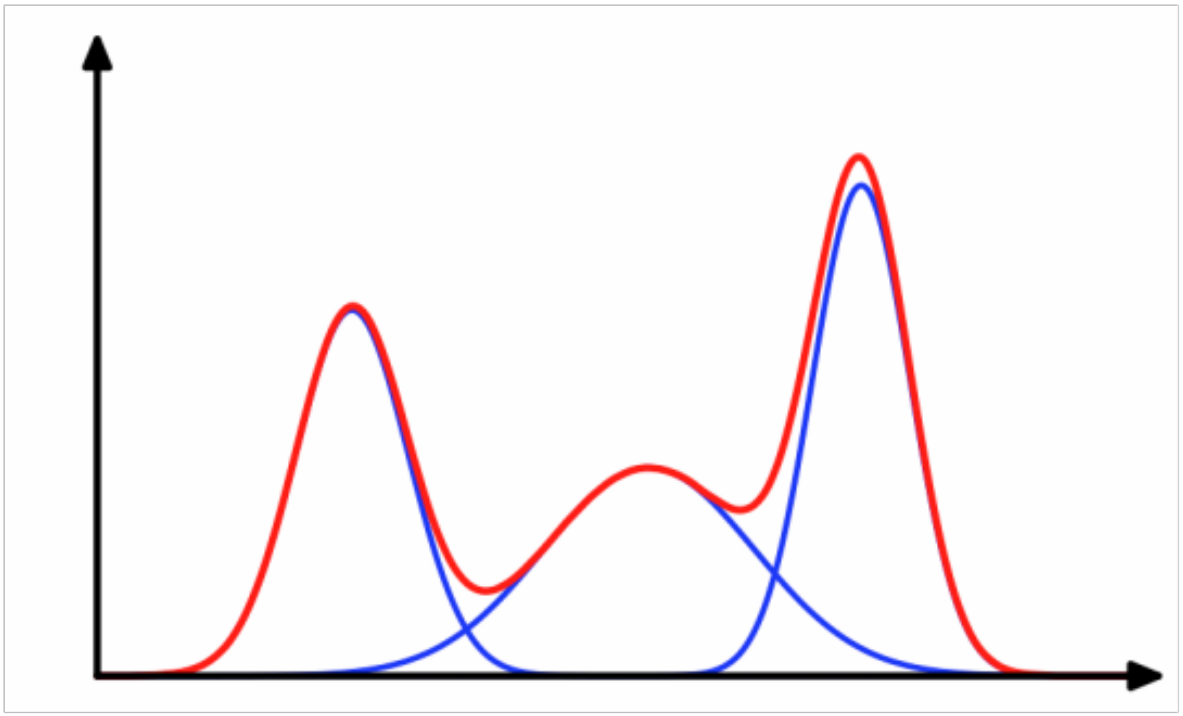
[Andrew Ng]

Mixture models

- Might model each feature x as a Gaussian distribution with mean μ and variance σ^2.
  - (Probability of x taking different values).
- Often, the data we are trying to model is much more complex.
- Model the data in terms of a mixture of several components.

Mixture models

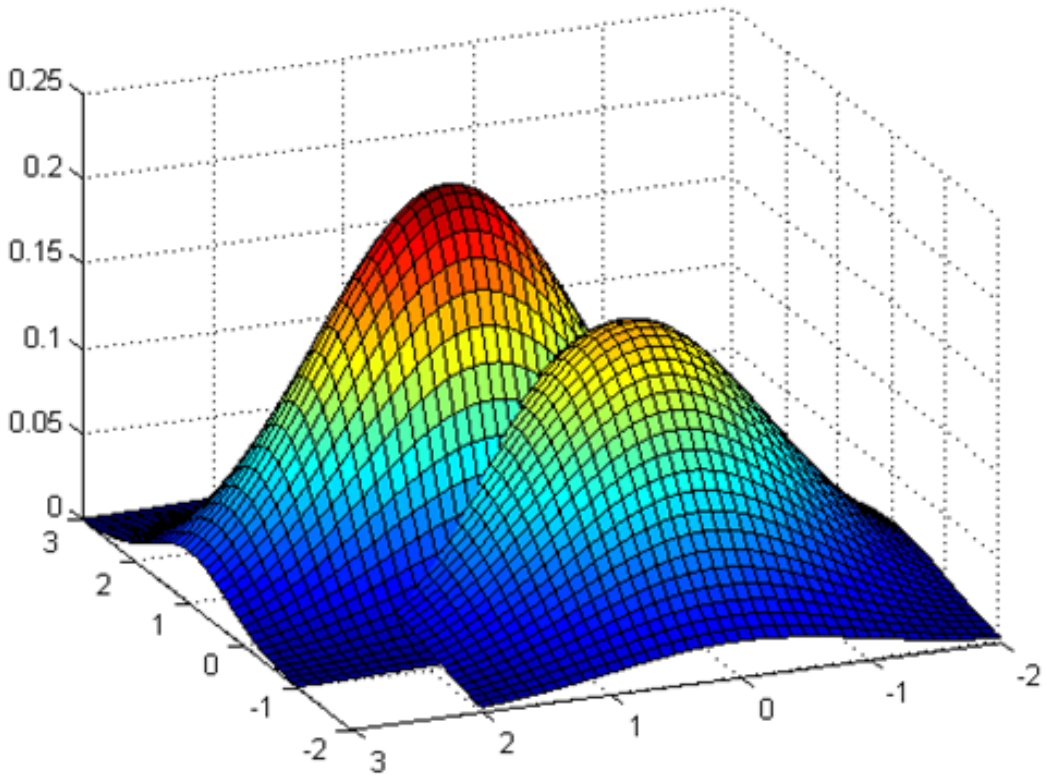
- A mixture model is combination of probability models:



- A univariate Gaussian mixture.

Mixture models

- A mixture model is combination of probability models:



(César Souza)

- A multivariate Gaussian mixture.

KING'S College LONDON

Mathematically

- In the general case, we mix K base distributions (can be of any type) so that:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}_i)$$

  where the π_k are the mixing weights.
- In other words, the probability of an element $\mathbf{x}_i$ is the sum of the probabilities assigned to $\mathbf{x}_i$ by each of the probability models in the mixture.

Mathematically

- The most widely used mixture model is probably the mixture of Gaussians.
  (What those examples on the previous slides were).
- Each model is a multivariate Gaussian with mean $\boldsymbol{\mu}_k$ and covariance Σ_k (matrix $n \times n$).
- Mixture is:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)$$

  which again just assigns $\mathbf{x}_i$ the sum of the probabilities given to it by each of the components of the mixture.

Mathematically

- When all Gaussians are univariate we have:

$$p(x_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \sigma_k^2)$$

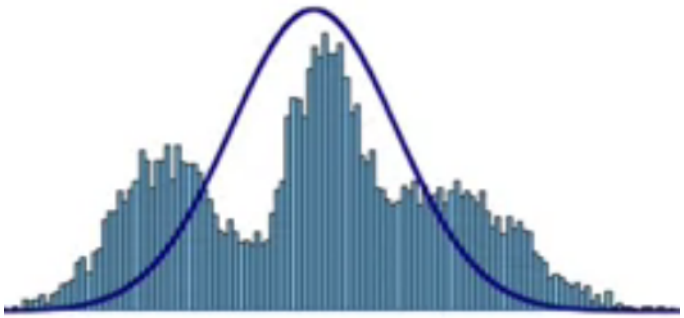
where σ_k^2 is the variance.

Mathematically

- When all Gaussians are univariate we have:

$$p(x_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \sigma_k^2)$$

  where σ_k^2 is the variance.
- Probabilistic model for representing a number of normally distributed sub-populations within an overall population.
- Example: modelling human height (fitting a single Gaussian doesn't work)



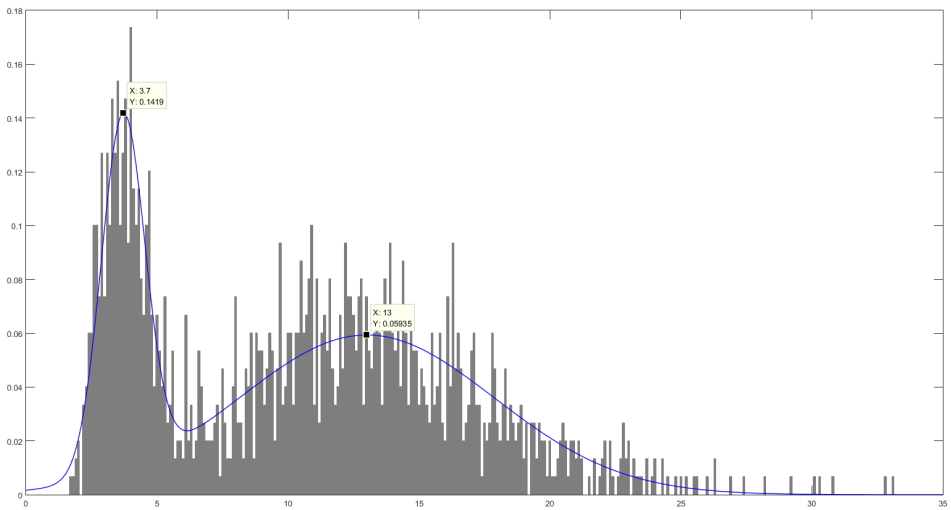
[UPenn: Robotics: Estimation and Learning]

Why are these useful?

- Lots of processes are Gaussian.
- Meaning lots of stuff in the world generates values that are normally distributed.
- Combinations of these things will be mixture models.

Estimating the means of k Gaussians

- Assume that $\mathcal{D}$ is generated by a mixture model.
- First, pick the distribution with equal probability $\frac{1}{K}$.
- Then pick a value according to the distribution.
- Repeat.



(Mathworks)

- How can you recover the Gaussians? (estimate the parameters)

Estimating the means of k Gaussians

- Assume that each Gaussian has the same variance. (Not like the previous picture)
- Then we want to learn a hypothesis that describes the means of each distribution

$$h = \langle \mu_1, \ldots, \mu_k \rangle$$

  such that $p(\mathcal{D}|h)$ is maximum.
- Maximum likelihood hypothesis, h_{ML}

Estimating the means of k Gaussians

- Now, if there was just one Gaussian, this job would be easy.
- The solution is the sample mean:

$$\mu_{ML} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

Estimating the means of k Gaussians

- But that is not what we need. Because we have k Gaussians.
- If we knew which Gaussian distribution each x_i came from, the solution would be easy.
- But we don't.
- We have a hidden variable — knowledge of which Gaussian distribution each x_i is from.
- Instead of x_i, you can think of the description of an instance as a tuple:

$$\langle x_i, z_{i1}, \ldots, z_{ik} \rangle$$

  where z_{ij} indicate which distribution generates x_i:

  each z_{ij} equals 1 if x_i came from that distribution and 0 otherwise.
- Solve this problem using the Expectation Maximisation (EM) algorithm.

KING'S College LONDON

EM algorithm

- In this case, EM works as the following.
- Given a current hypothesis:

$$h_j = \langle \mu_1, \ldots, \mu_k \rangle$$

  estimate the expected values of the hidden variables z_{ij}.
- Then re-calculate the maximum likelihood hypothesis given those values.

EM algorithm

- Let's start by considering the two Gaussian case.
- We initialise EM by picking $h = \langle \mu_1, \mu_2 \rangle$, for arbitrary initial values.
- Then repeat:
  1. Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij} assuming $h = \langle \mu_1, \mu_2 \rangle$ is true.
  2. Calculate a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$ assuming each z_{ij} takes its expected value $E[z_{ij}]$. Replace $h = \langle \mu_1, \mu_2 \rangle$ by $h' = \langle \mu'_1, \mu'_2 \rangle$

  Until convergence (change in parameters becomes very small).

EM algorithm

- Step 1: calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}.
- $E[z_{ij}]$ is the probability that x_i was generated by the jth normal distribution.

$$E[z_{ij}] = \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i | \mu = \mu_n)}$$
$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

- Relative ratio of jth Gaussian in ith example.
- Then the first step is just plugging the current values of μ_1, μ_2 and x_i into the above equation.

KING'S College LONDON

EM algorithm

- Step 2: calculate a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$.
- Adjust the values of the means to maximise the likelihood.

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

- The sample mean for μ_j, weighted by the expectation $E[z_{ij}]$.
- If $E[z_{ij}]$ is small (prob that x_i belongs to jth distribution), then x_i contributes less to μ_j (the mean of jth).

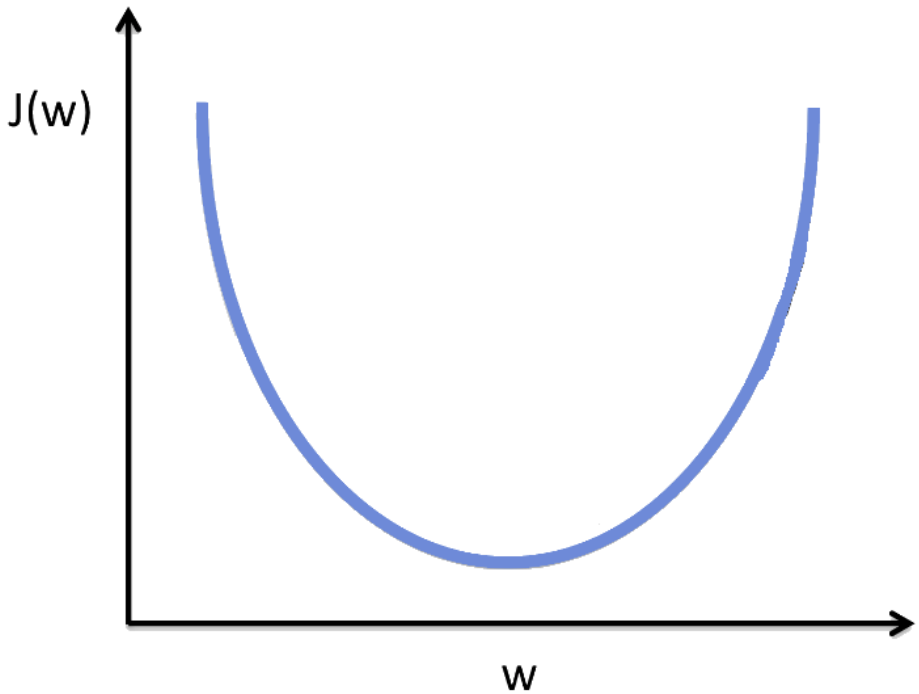
EM algorithm

- Can show that each iteration adjusts h to increase the likelihood $p(\mathcal{D}|h)$ unless likelihood is at a local maximum.
- Thus EM converges to a local maximum of the likelihood.
- Global vs. local maximum/minimum.

EM algorithm

- Can show that each iteration adjusts h to increase the likelihood $p(\mathcal{D}|h)$ unless likelihood is at a local maximum.
- Thus EM converges to a local maximum of the likelihood.
- Global vs. local maximum/minimum.

(side note)



(https://rasbt.github.io)

EM algorithm more generally

- EM can be applied to any situation in which we want to estimate parameters

$$\theta$$

that describe a distribution, given only a portion of the data generated by the distribution.

EM algorithm more generally

- In general, we have:

$$\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$$

as m independently drawn observations, and:

$$Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$$

as the corresponding unknown data which identifies which distribution the $\mathbf{x}_1$ come from.

- Then:

$$Y = \mathcal{D} \cup Z$$

is the full data.

EM algorithm more generally

- h is our current hypothesis about the value(s) of θ.
- And θ is the set of parameters for the model we are trying to learn.
- For example, for a set of k Gaussians with the same variance:

$$\theta = \langle \mu_1, \mu_2, \ldots, \mu_k \rangle$$

- For example, for a set of k Gaussians with the different variance:

$$\theta = \langle \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \ldots, \mu_k, \sigma_k^2, \rangle$$

- Or whatever set of parameters are appropriate.

KING'S College LONDON

EM algorithm more generally

- EM searches for the ML hypothesis h' by looking for the h' that maximises:

$$E[\ln p(Y|h')]$$

- Here $p(Y|h')$ is the likelihood of all the data given h'.
- Remember:

$$Y = \mathcal{D} \cup Z$$

where $\mathcal{D}$ is the data, and Z is the hidden data saying which element of $\mathcal{D}$ comes from which distribution.

EM algorithm more generally

- If we maximise its log, we maximise it.
- We look for expectation $E[\ln p(Y|h')]$, because Y is a random variable (expected value over the probability distribution governing the random variable Y).
- Since Y contains the unobserved elements Z, we have to compute a weighted average over possible values of Z, where weights are the probabilities of the values of Z.
- If we knew the probability distribution over Y, we could compute this.
- But we don't. Because it depends on θ.

  (θ is the set of parameters we want to learn.)

EM algorithm more generally

- Since we don't know θ, EM uses the current hypothesis h (about θ) to estimate the distribution over Y.
- Define $Q(h'|h)$ as:

$$Q(h'|h) = E[\ln p(Y|h')|h, \mathcal{D}]$$

giving $E[\ln p(Y|h')]$ under the assumption that $\mathcal{D}$ and $\theta = h$

EM algorithm more generally

- So, EM repeats:
  1. Expectation (E) step
     Calculate $Q(h'|h)$ using the current hypothesis h and observed data $\mathcal{D}$ to estimate the probability over Y:

$$Q(h'|h) \leftarrow E[\ln p(Y|h')|h, \mathcal{D}]$$

  2. Maximisation (M) step
     Replace h by h' which maximises Q:

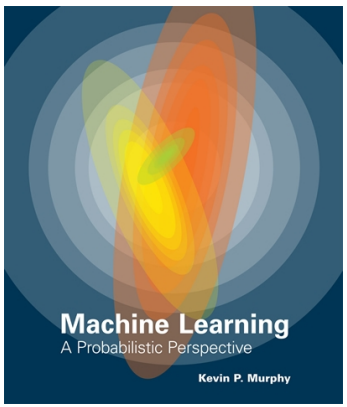
$$h \leftarrow \arg\max_{h'} Q(h'|h)$$

until convergence.

EM algorithm more generally

- If Q is continuous, then EM converges to a stationary point of the likelihood function $p(Y|h')$.
- Global maximum if the likelihood function has a single maximum.
- Otherwise a local maximum.
- So, random restart, or just careful selection of starting points.
- (`k-means++`)

- See Murphy, page 355.



KING'S College LONDON

K-means

- Our example of establishing the means of K is common enough that its a thing in ML.
- K-means algorithm.
- Reframe as a method for clustering.
- Each μ_j is a cluster centre, and each x_i is assigned to the cluster closest to it.
- Leads to another statement of the algorithm (clustering algorithm).

K-means

- Pick random values for $\boldsymbol{\mu}_k$.
- Then repeat:
  1. Assign each $\mathbf{x}_i$ to its closest cluster centre:

$$z_i = \arg\min_z |\mathbf{x}_i - \boldsymbol{\mu}_k|$$

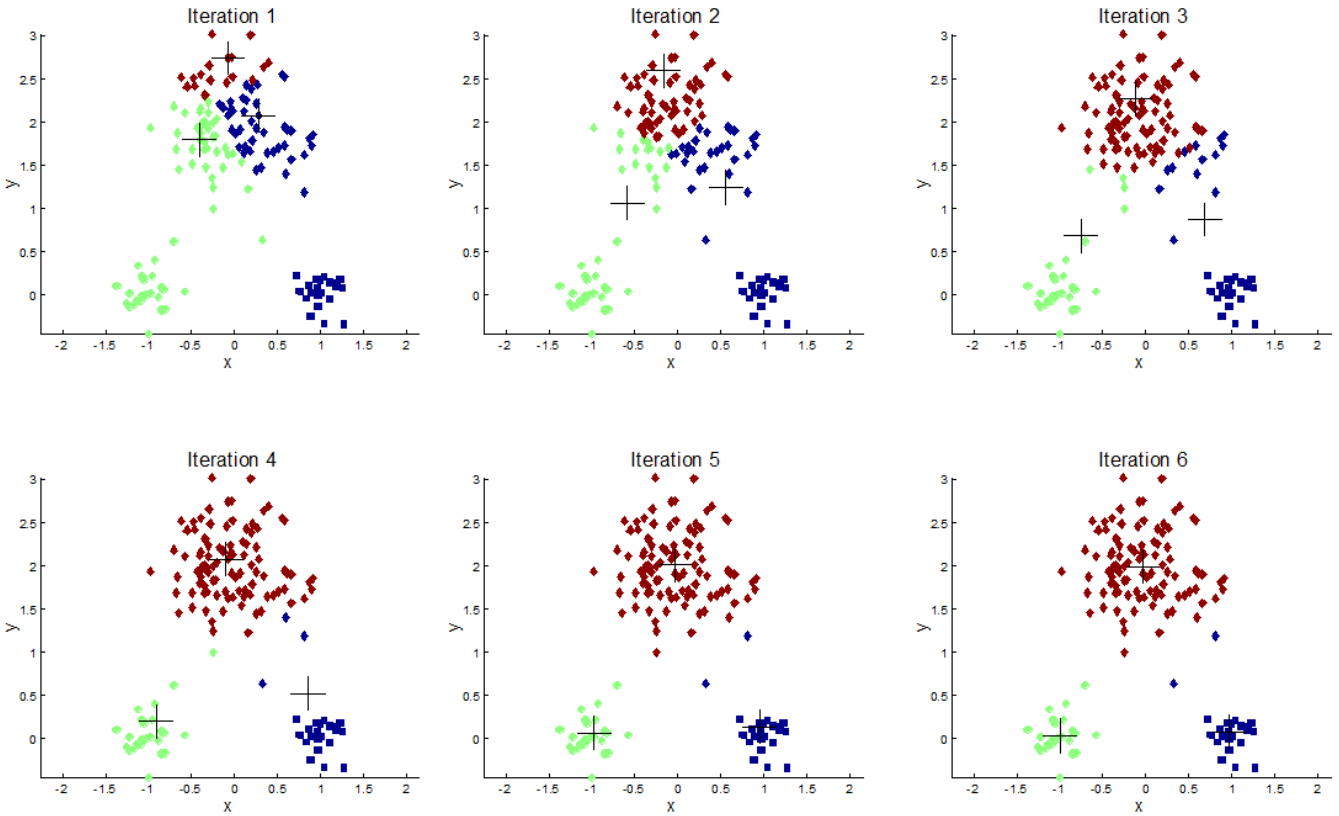
  2. Update each cluster centre as the mean of the points assigned to that cluster:

$$\boldsymbol{\mu}_k = \frac{1}{|\{j : z_j = k\}|} \sum_{i \in \{j : z_j = k\}} \mathbf{x}_i$$

  until converged.
- Note that the notation allows for multi-dimensional $\mathbf{x}_i$ and $\boldsymbol{\mu}_k$.

KING'S College LONDON

K-means



(Andrei Pandre)

Two K-means

- Hard clustering: the K-means we just looked at just reports which cluster $\mathbf{x}_i$ is part of.
- Soft clustering: probability that $\mathbf{x}_i$ is in the cluster.
- Can think of this in the context of EM and K-Gaussians.
- So for every point $\mathbf{x}_i$ we have K probabilities:

$$p_k(\mathbf{x}_i)$$

Summary

- Today we talked about the role of probability in machine learning.
- Considered machine learning as an exercise in probabilistic inference.
- But mainly focused on:
  - Naive Bayes classifier
  - Gaussian Mixture Models
  - K-means clustering

  and how these models may be learnt from data.
- EM algorithm.