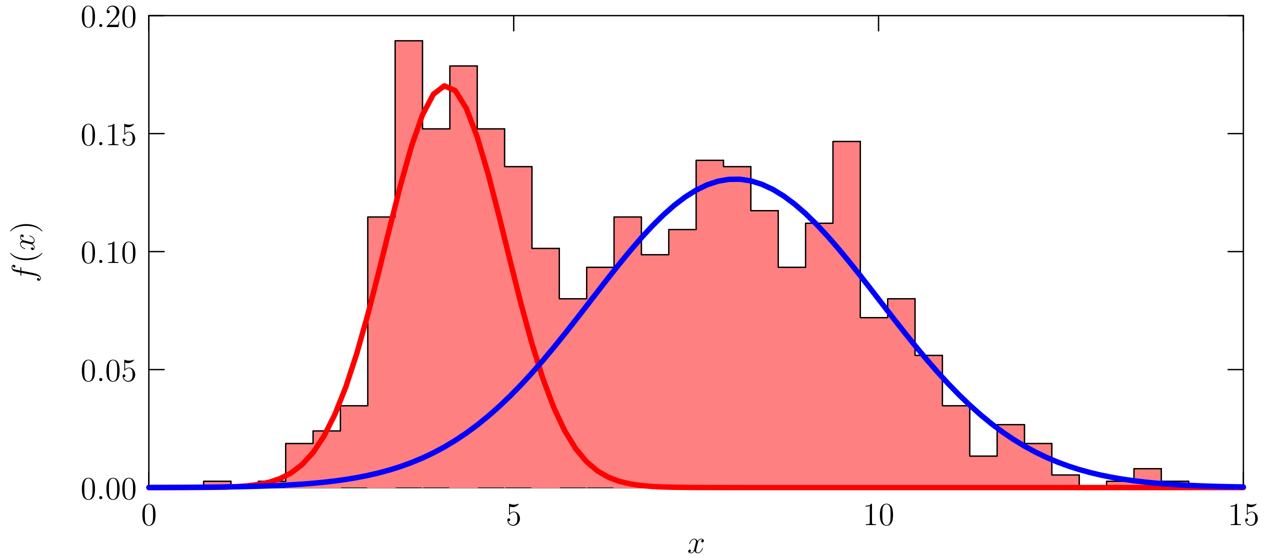
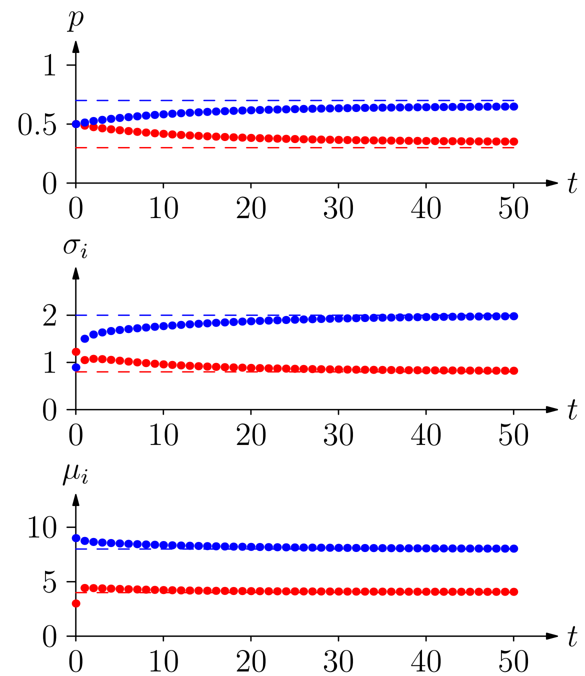
Advanced Machine Learning

Probabilistic Inference

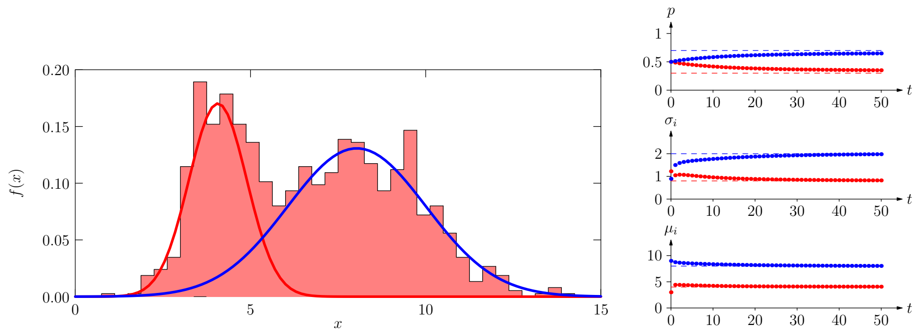




Hierarchical Models, Mixture of Gaussians, Expectation Maximisation

Outline

1. **Building Probabilistic Models**
2. Mixture of Gaussians
3. Expectation Maximisation



Building Probabilistic Models

- To describe a system with uncertainty we use random variables, X, Y, Z, etc.
- We use the convention of writing random variables in capitals (this is sometimes confusing as when you observe a random variables it is no longer random)
- The variables are described by probability mass function $\mathbb{P}(X,Y,Z)$ or if our variables are continuous, but probability densities $f_{X,Y,Z}(x,y,z)$
- A major rule of probability is

$$\sum_{X} \mathbb{P}(X,Y,Z) = \mathbb{P}(Y,Z)$$

Conditional Probabilities

- When developing models it is often useful to consider conditional probabilities e.g. $\mathbb{P}(X,Y|Z)$ or $f_{X|Y,Z}(x|y,z)$

- A second major rule in probabilistic modelling is

$$\mathbb{P}(X,Y) = \mathbb{P}(X|Y)\,\mathbb{P}(Y) = \mathbb{P}(Y|X)\,\mathbb{P}(X)$$

- This is a mathematical identity that does not imply causality (it defines conditional probability)

- It is the origins of Bayes' rule: $\mathbb{P}(X|Y) = \dfrac{\mathbb{P}(Y|X)\,\mathbb{P}(X)}{\mathbb{P}(Y)}$

Discriminative Models

- We often think of our observations as given and the predictions as random variables
- For example we might be given some features $\boldsymbol{x}$ and we wish to predict a class $C \in \mathcal{C}$
- Our objective is then to find the probability $\mathbb{P}(C|\boldsymbol{x})$
- This is known as a **discriminative model**
- E.g. in *foundations of machine learning* you learnt how to find the Bayes' optimal discrimination surface

Generative Models

- Sometimes it is easy to think about the joint process of generating the features and outputs together
- This leads to a joint distribution $\mathbb{P}(\boldsymbol{X}, Y)$ where $\boldsymbol{X}$ are your features and Y is your output you are trying to predict
- This is known as a **generative model**
- Generative models are often more natural to think about
- We can use them to do discrimination using

$$\mathbb{P}(Y|\boldsymbol{X}) = \frac{\mathbb{P}(\boldsymbol{X}, Y)}{\mathbb{P}(\boldsymbol{X})} = \frac{\mathbb{P}(\boldsymbol{X}, Y)}{\sum_{Y} \mathbb{P}(\boldsymbol{X}, Y)}$$

Latent Variables

- Sometimes we have models that involve random variables that we don't observe and we don't care about
- These are called **latent variables**
- If we have a latent variable Z and observed variable $\boldsymbol{X}$ and we are predicting a variable Y then we would **marginalise** over the latent variable

$$\mathbb{P}(\boldsymbol{X}, Y) = \sum_{Z} \mathbb{P}(\boldsymbol{X}, Y, Z)$$

Modelling Virus

- Suppose we want to estimate the number of hospitalisation from Corona virus in the next month
- Our observable is the number of reported cases
- In our model we might want to estimate the number of actual cases
- This would be a latent variable (it is not an observable or our final target, but it is very useful intermediate in our model)
- This will be a random variable (we are uncertain, but we can build a probabilistic model giving a distribution of number of actual cases)

Hierarchical Models

- Of course, if I was really modelling the spread of a disease I would care about the probability, $f(C|A,V)$, of catching the disease, C, given the persons age A and the variant of the disease V
- I would want to know the distribution of ages $f(A)$ and try to infer the probability of different variants $\mathbb{P}(V)$
- I would care about the probability, $f(R|A,V)$, of cases being reported given age and variant
- And the probability, $f(H|A,V)$, of hospitalisation given A and V
- This would involve an elaborate (hierarchical) model with a large number of latent variables

Probabilistic Inference

- We can use Bayes' rules to learn a set of parameter $\boldsymbol{\Theta}$ that occur in our likelihood function

$$\mathbb{P}(\boldsymbol{\Theta}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\boldsymbol{\Theta})\,\mathbb{P}(\boldsymbol{\Theta})}{\mathbb{P}(\mathcal{D})}$$

- This provides us a full probabilistic description of the parameters
- It doesn't overfit (we are not choosing the best)
- Bayesian inference provides a description of its own uncertainty
- We need to specify a likelihood and prior, but this is usually not difficult

Problem with Bayes

- Bayes is problematic because it is often hard
- The posterior is often not expressible as a nice probability function
- We need to compute the *evidence* or *margin likelihood* we use

$$\mathbb{P}(\mathcal{D}) = \sum_{\mathbf{\Theta}} \mathbb{P}(\mathcal{D}|\mathbf{\Theta})\,\mathbb{P}(\mathbf{\Theta})$$

- But sometimes the number of values that $\mathbf{\Theta}$ can take are so large that we cannot easily compute this
- Nevertheless we can usually do this using Monte Carlo techniques

Maximum A Posteriori (MAP) Solution

- One work around is to compute the mode of the posterior

$$\boldsymbol{\Theta}_{\mathsf{MAP}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, f(\mathcal{D}|\boldsymbol{\Theta})\, f(\boldsymbol{\Theta}) = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \log(f(\mathcal{D}|\boldsymbol{\Theta})) + \log(f(\boldsymbol{\Theta}))$$

- We don't need to calculate $f(\mathcal{D})$ or explicitly calculate the posterior distribution

- But it is not Bayesian (despite what you are sometime told)—its not properly probabilistic

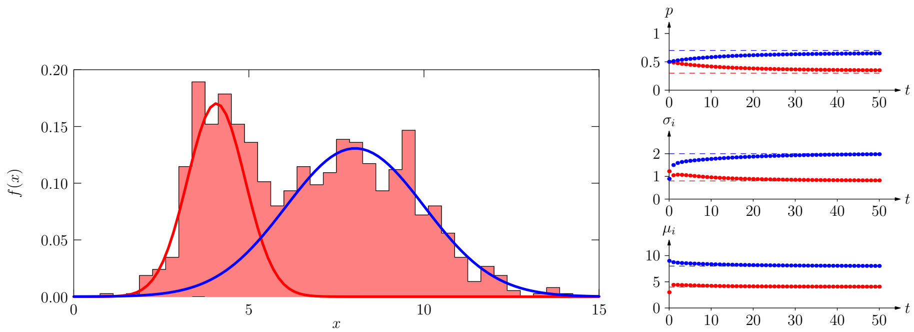
- You can overfit and you don't get an estimate of the error in your inference

Maximum Likelihood

- When we assume a uniform prior then the MAP solution is just maximising the likelihood
- Weirdly this hack was accepted as part of mainstream statistics even when Bayesian statistics was considered unscientific
- Maximum likelihood is often sufficient for *government work*, but it isn't the best you can do
- In high-dimensional problems using a non-uniform prior can make a big difference
- And, of course, doing a full probabilistic calculation has real advantages

Outline

1. Building Probabilistic Models
2. **Mixture of Gaussians**
3. Expectation Maximisation



Mixture of Gaussians

- Suppose we were observing the decays from two types of short-lived particle, A or B
- We observe the half life, X_i, but not the particle type
- We assume X_i is normally distributed with unknown means and variances: $\boldsymbol{\Theta} = \{\mu_A,\, \sigma_A^2,\, \mu_B,\, \sigma_B^2\}$
- Let $Z_i \in \{0,1\}$ be an indicator that particle i is of type A
- The probability of X_i is given by

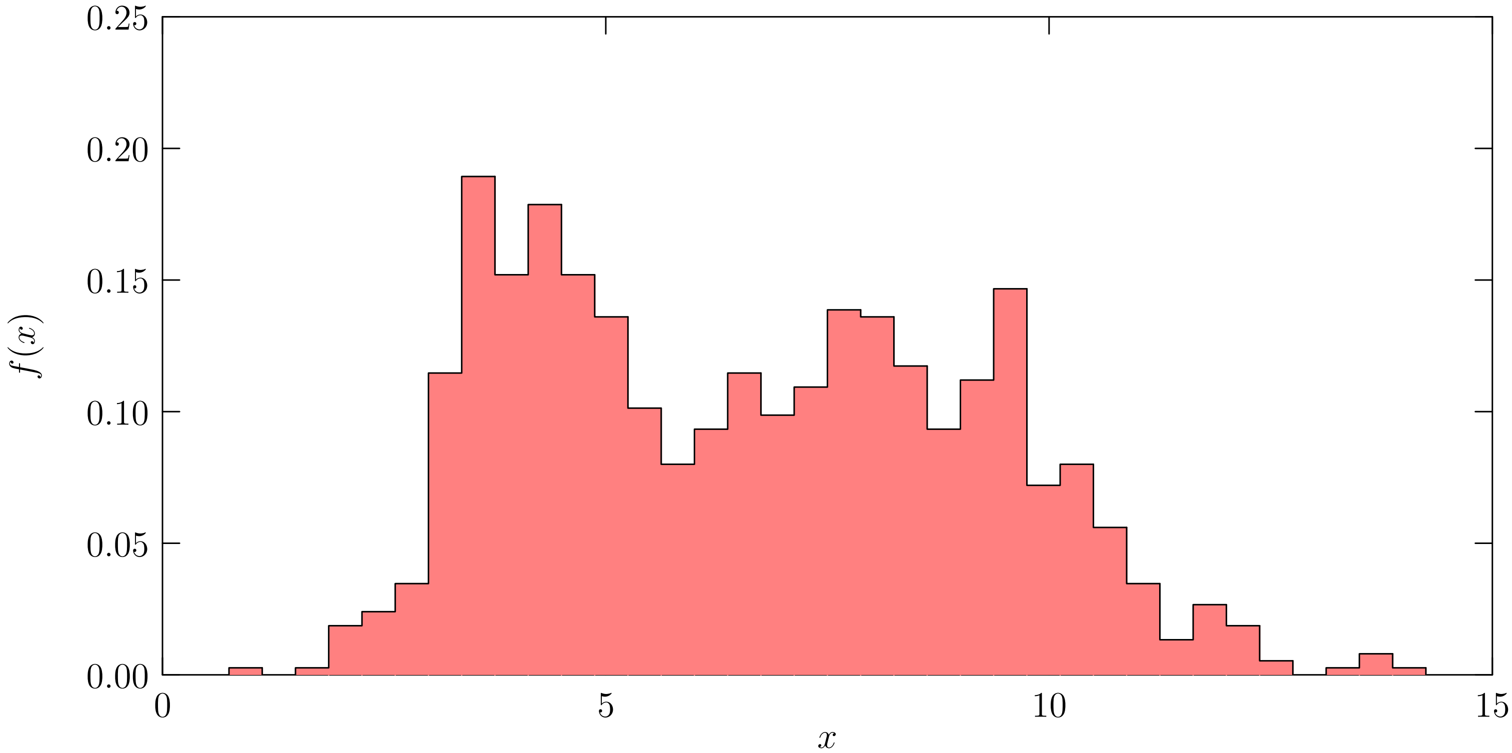
$$f(X_i|Z_i,\boldsymbol{\Theta}) = Z_i \mathcal{N}\big(X_i \big| \mu_A,\sigma_A^2\big) + (1 - Z_i)\mathcal{N}\big(X_i \big| \mu_B,\sigma_B^2\big)$$

Data

- Note that

$$f(X_i|\boldsymbol{\Theta}) = \sum_{Z_i \in \{0,1\}} f(X_i, Z_i|\boldsymbol{\Theta}) = \sum_{Z_i \in \{0,1\}} f(X_i|Z_i, \boldsymbol{\Theta})\mathbb{P}(Z_i)$$

$$= \mathbb{E}_{Z_i}[f(X_i|Z_i, \boldsymbol{\Theta})] = p\mathcal{N}\left(X_i \middle| \mu_A, \sigma_A^2\right) + (1-p)\mathcal{N}\left(X_i \middle| \mu_B, \sigma_B^2\right)$$



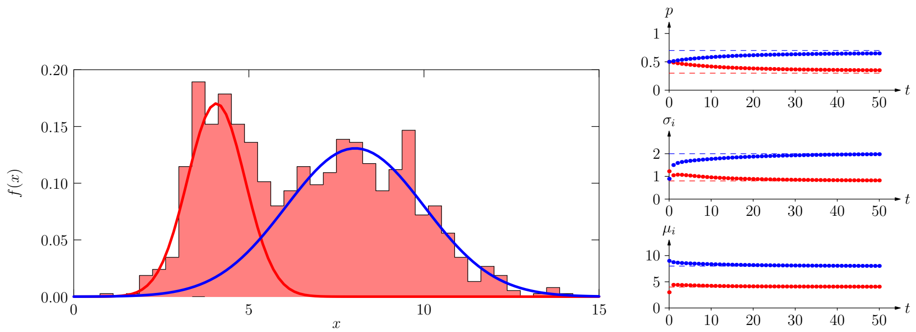
Maximum Likelihood

- To solve the model as a Bayesian we would have to assign priors to our parameters $\boldsymbol{\Theta} = (\mu_A, \sigma_A, \mu_B, \sigma_B, p)$
- This is doable, but complicated (we would also end up with a distribution for our parameters)
- Often we only want a reasonable estimate for some of our parameters (e.g. the half-lives μ_A and μ_B)
- A reasonable approach is to seek those parameters that maximise the likelihood of our observed data

$$f(\mathcal{D}|\boldsymbol{\Theta}) = \prod_{X \in \mathcal{D}} f(X|\boldsymbol{\Theta})$$

Outline

1. Building Probabilistic Models
2. Mixture of Gaussians
3. **Expectation Maximisation**



Maximum Likelihood with Latent Variables

- The maximum likelihood is a non-linear function of the parameters so cannot be immediately maximised
- If we knew which type of particle a data-point belongs to (Z_i) then it would be straightforward to maximise the likelihood
- As we don't we need to estimate $\mathbb{P}(Z_i = 1)$, but this depends on μ_A, σ_A^2, μ_B, σ_B^2 and p
- We could use a standard optimiser, but this is slightly inelegant

EM Algorithm

- Instead we can use an **expectation-maximisation algorithm** usually known as an **EM algorithm**

- We proceed iteratively by maximising the expected log-likelihood with respect to the current set of parameters

$$\Theta^{(t+1)} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{\boldsymbol{Z}} \mathbb{P}\left(\boldsymbol{Z} \middle| \mathcal{D}, \boldsymbol{\Theta}^{(t)}\right) \log(f(\mathcal{D}|\boldsymbol{Z}, \boldsymbol{\Theta}))$$

- It isn't obvious why this works

Why EM Algorithm Works

- The argument around why this works is quite involved
- Note that at each step we maximise

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \sum_{\boldsymbol{Z}\in\{0,1\}^m} \mathbb{P}\Big(\boldsymbol{Z}\Big|\mathcal{D},\boldsymbol{\Theta}^{(t)}\Big) \log(f(\mathcal{D}|\boldsymbol{Z},\boldsymbol{\Theta}))$$

- We can show that the maximum, $\boldsymbol{\Theta}^{(t+1)}$, is such that

$$\log\Big(f(\mathcal{D}|\boldsymbol{\Theta}^{(t+1)})\Big) - \log\Big(f(\mathcal{D}|\boldsymbol{\Theta}^{(t)})\Big) \geq Q(\boldsymbol{\Theta}^{(\boldsymbol{t}+\boldsymbol{1})}|\boldsymbol{\Theta}^{(t)}) - Q(\boldsymbol{\Theta}^{(\boldsymbol{t})}|\boldsymbol{\Theta}^{(t)}) \geq 0$$

- The details are given in the supplemental notes

Conditional Latent Variables

- We need to compute the distribution of latent variables conditioned on the data and current estimated parameters

- For our problem

$$\mathbb{P}\left(\boldsymbol{Z}\middle|\mathcal{D},\boldsymbol{\Theta}^{(t)}\right) = \prod_{i=1}^{m} \mathbb{P}\left(Z_i\middle|X_i,\boldsymbol{\Theta}^{(t)}\right)$$

where

$$\mathbb{P}\left(Z_i = 1\middle|X_i,\boldsymbol{\Theta}^{(t)}\right) = \frac{p^{(t)}\mathcal{N}\left(X_i\middle|\mu_A^{(t)},\sigma_A^{2(t)}\right)}{p^{(t)}\mathcal{N}\left(X_i\middle|\mu_A^{(t)},\sigma_A^{2(t)}\right) + (1-p^{(t)})\mathcal{N}\left(X_i\middle|\mu_B^{(t)},\sigma_B^{2(t)}\right)}$$

$$\mathbb{P}\left(Z_i = 0\middle|X_i,\boldsymbol{\Theta}^{(t)}\right) = 1 - \mathbb{P}\left(Z_i = 1\middle|X_i,\boldsymbol{\Theta}^{(t)}\right)$$

EM for Mixture of Gaussians

- Maximise with respect to parameters $\boldsymbol{\theta}$

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{\boldsymbol{Z}} \mathbb{P}\left(\boldsymbol{Z}|\mathcal{D},\boldsymbol{\Theta}^{(t)}\right) \log(f(\mathcal{D}|\boldsymbol{Z},\boldsymbol{\Theta})) = \sum_{i=1}^{m} \sum_{Z_i} \mathbb{P}\left(Z_i|\mathcal{D},\boldsymbol{\Theta}^{(t)}\right) \log(f(X_i|Z_i,\boldsymbol{\Theta}))$$

$$= \sum_{i=1}^{m} \sum_{Z_i \in \{0,1\}} \mathbb{P}\left(Z_i|X_i,\boldsymbol{\theta}_i^{(t)}\right) \Big(Z_i \log(p) + (1 - Z_i)\log(1-p)$$

$$- \frac{(X_i - \mu_{Z_i})^2}{2\sigma_{Z_i}^2} - \log\left(\sqrt{2\pi}\sigma_{Z_i}\right)\Big)$$

- Compute update equations

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \mu_k} = 0, \qquad \frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \sigma_k} = 0, \qquad \frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial p} = 0$$

Update Equations

- Means

$$\mu_{Z_i}^{(t+1)} = \frac{\sum_{i=1}^{n} \mathbb{P}\big(Z_i|X_i,\boldsymbol{\theta}^{(t)}\big)\,X_i}{\sum_{i=1}^{n} \mathbb{P}\big(Z_i|X_i,\boldsymbol{\theta}^{(t)}\big)},$$

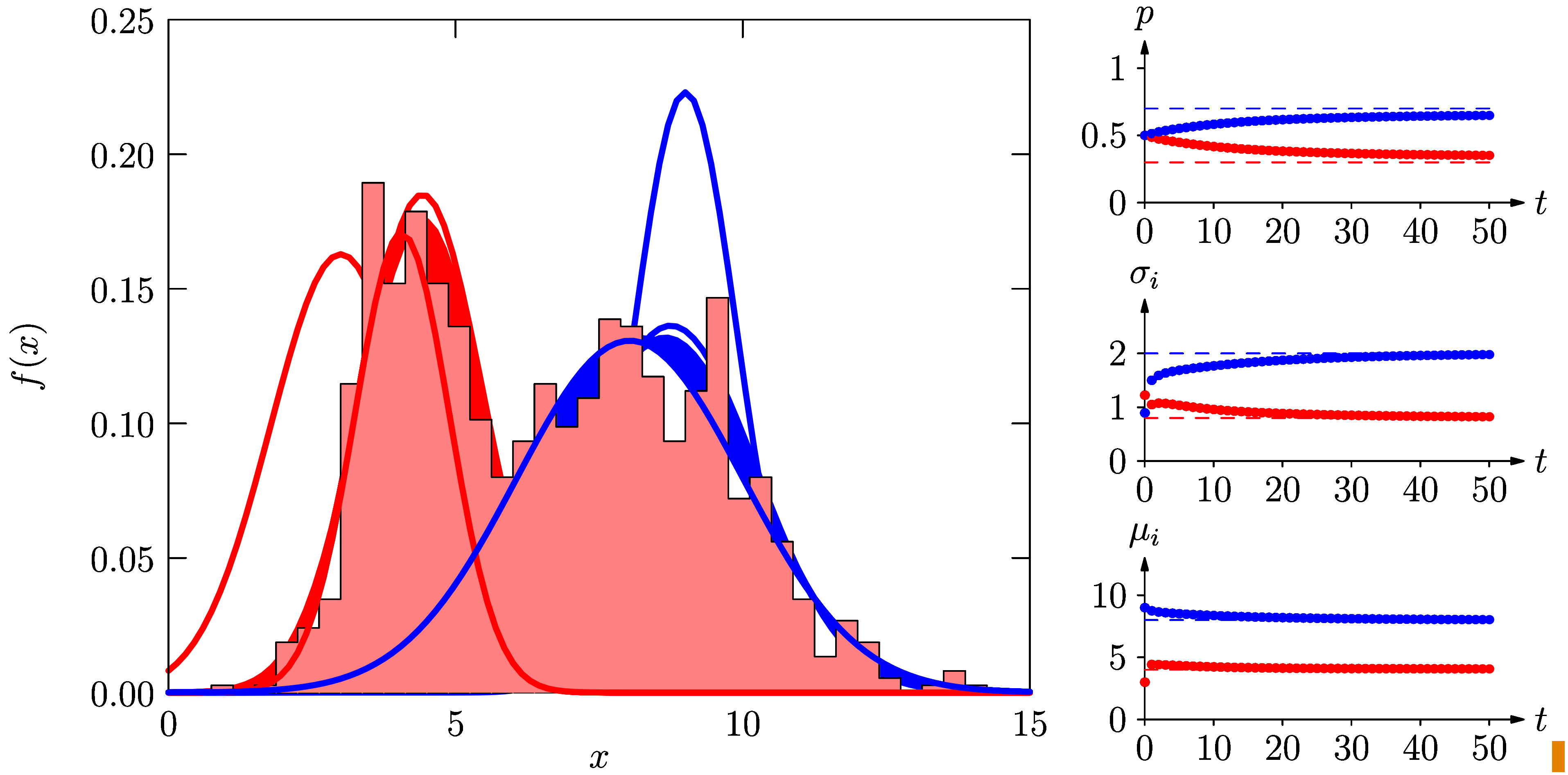
- Variances

$$(\sigma_{Z_i}^{(t+1)})^2 = \frac{\sum_{i=1}^{n} \mathbb{P}\big(Z_i|X_i,\boldsymbol{\theta}^{(t)}\big)\,(X_i - \mu_{Z_i}^{(t+1)})^2}{\sum_{i=1}^{n} \mathbb{P}\big(Z_i|X_i,\boldsymbol{\theta}^{(t)}\big)}$$

- Probability of being type 1

$$p^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\Big(Z_i = 1 \mid X_i,\boldsymbol{\theta}_i^{(t)}\Big)$$

Example



0.25
0.20
0.15
0.10
0.05
0.00
$f(x)$
0
5
10
15
x
p
1
0.5
0
t
0
10
20
30
40
50
σ_i
2
1
0
t
0
10
20
30
40
50
μ_i
10
5
0
t
0
10
20
30
40
50

Summary

- Building probabilistic models is an intricate process
- Identifying random variables that describe the system is the first step
- Often we need to introduce variables that we don't observe and need to be marginalised out
- The EM algorithm provide one approach to maximising likelihoods or MAP solutions when we have latent variables
- It often gives nice update equations, but convergence can be slow