Chapter 8: Model Inference and Averaging

DD3364

April 29, 2012

Introduction

Today's Lecture

- This chapter covers a lot of ideas / techniques !
- Will focus more on the later sections.
- Would probably need several lectures to cover the material properly.
- But here goes.....

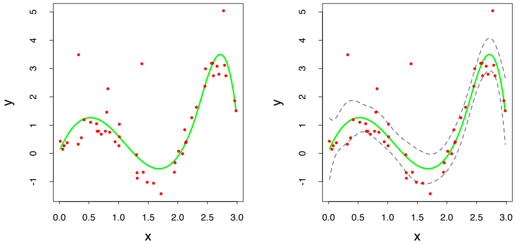
The Bootstrap and Maximum Likelihood Methods

Maximum likelihood estimate

Using training data $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ fit **this model**

$$Y = \sum_{j=1}^{J} \beta_j \, h_j(X)$$

using the ML estimate.



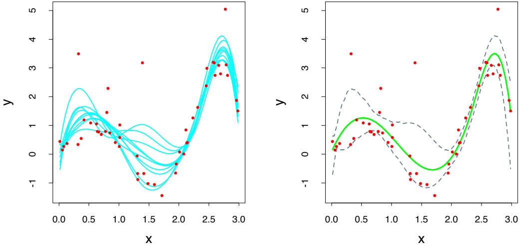
Can also put error bounds on the estimate if assume an additive error model.

Bootstrap estimate and variance estimate

Using training data $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ fit **this model**

$$Y = \sum_{j=1}^{J} \beta_j \, h_j(X)$$

Iterate: Take bootstrap sample and compute the ML estimate.

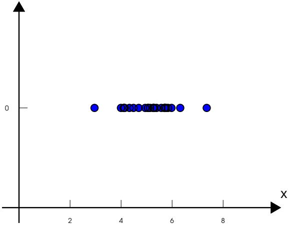


From bootstrap fits can find the mean estimate and put error bounds on the estimates.

Maximum Likelihood Inference

Parameter estimation

Have n independent draws $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $p(\mathbf{x} \mid \Theta)$.

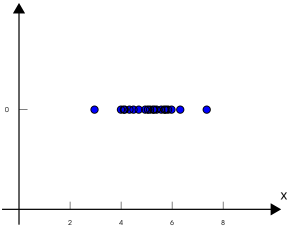


$\longleftarrow$ **1D example**

Each $\mathbf{x}_i \sim N(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma)$ where $\Theta = (\boldsymbol{\mu}, \Sigma)$

Parameter estimation

Have n independent draws $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $p(\mathbf{x} \,|\, \Theta)$.



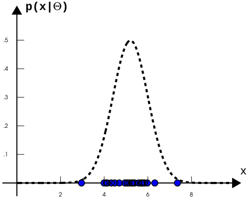
$\longleftarrow$ **1D example**

Each $\mathbf{x}_i \sim N(\mathbf{x} \,|\, \boldsymbol{\mu}, \Sigma)$ where $\Theta = (\boldsymbol{\mu}, \Sigma)$

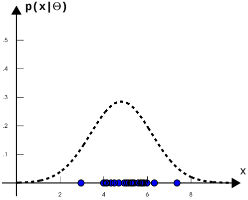
Want to estimate the parameters Θ from the $\mathbf{x}_i$'s

Parameter estimation

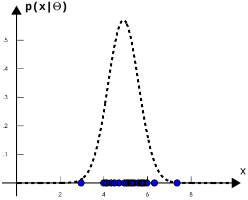
Have n independent draws $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $p(\mathbf{x} \mid \Theta)$.



$\Theta = (5.2, .8)$



$\Theta = (4.8, 1.4)$



$\Theta = (4.9, .7)$

Want to estimate the parameters Θ from the $\mathbf{x}_i$'s.

HOW??

Maximum Likelihood Estimation (MLE)

Choose the Θ which maximizes the likelihood of your data:

$$\Theta^* = \arg\max_{\Theta} p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \,|\, \Theta)$$

Maximum Likelihood Estimation (MLE)

Choose the Θ which maximizes the likelihood of your data:

$$\begin{aligned} l(\Theta; \mathbf{X}) &\equiv p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \,|\, \Theta) \\ &= \prod_{i=1}^{n} p(\mathbf{x}_i \,|\, \Theta) \quad \leftarrow \text{assuming independent samples} \end{aligned}$$

Maximum Likelihood Estimation (MLE)

Choose the Θ which maximizes the likelihood of your data:

$$
\begin{aligned}
l(\Theta; \mathbf{X}) &\equiv p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \mid \Theta) \\
&= \prod_{i=1}^{n} p(\mathbf{x}_i \mid \Theta) \quad \leftarrow \text{assuming independent samples}
\end{aligned}
$$

Easier to work with the log-likelihood

$$
L(\Theta; \mathbf{X}) = \log\left(l(\Theta; \mathbf{X})\right) = \sum_{i=1}^{n} \log\left(p(\mathbf{x}_i \mid \Theta)\right)
$$

Maximum Likelihood Estimation (MLE)

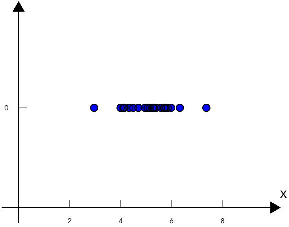
Choose the Θ which maximizes the likelihood of your data:

Note

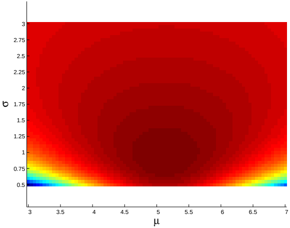
$$\Theta^* = \arg\max_{\Theta} l(\Theta; \mathbf{X}) = \arg\max_{\Theta} L(\Theta; \mathbf{X})$$

An example Log-likelihood function

Our 1D example of points drawn from $N(\mu, \Sigma)$



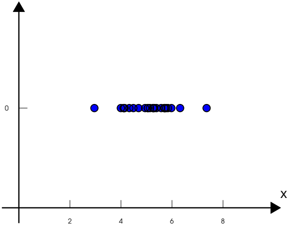
$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$



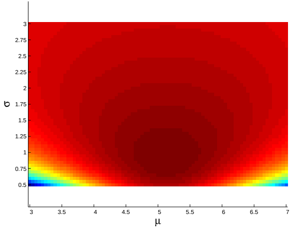
Log-likelihood: $L(\Theta; \mathbf{X})$

An example Log-likelihood function

Our 1D example of points drawn from $N(\mu, \Sigma)$



$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$



Log-likelihood: $L(\Theta; \mathbf{X})$

Want to find the maximum of this function $L(\Theta; \mathbf{X})$.

MLE for a Normal distribution

The formula for a normal distribution for $\mathbf{x} \in \mathcal{R}^d$:

$$p(\mathbf{x} \mid \Theta) = (2\pi)^{-\frac{d}{2}} \, |\Sigma|^{-\frac{1}{2}} \; \exp\left(-.5(\mathbf{x} - \boldsymbol{\mu})^t \, \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

MLE for a Normal distribution

The formula for a normal distribution for $\mathbf{x} \in \mathcal{R}^d$:

$$p(\mathbf{x} \mid \Theta) = (2\pi)^{-\frac{d}{2}} \, |\Sigma|^{-\frac{1}{2}} \, \exp\left(-.5(\mathbf{x} - \boldsymbol{\mu})^t \, \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

The log-likelihood of our n data-points is

$$\begin{aligned}
L(\Theta; \mathbf{X}) &= \sum_{i=1}^{n} \log\left(p(\mathbf{x}_i \mid \Theta)\right) \\
&= \sum_{i=1}^{n} \left[-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log\left(|\Sigma|\right) - .5(\mathbf{x}_i - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu})\right] \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(|\Sigma|\right) - .5\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(|\Sigma|\right) - .5 \, \mathrm{tr}\left[\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu})\right]
\end{aligned}$$

MLE for a Normal distribution

$$
\begin{aligned}
L(\Theta; \mathbf{X}) &= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(|\Sigma|\right) - .5\,\mathrm{tr}\left[\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^t\Sigma^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right] \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(|\Sigma|\right) - .5\,\mathrm{tr}\left[\sum_{i=1}^{n}\Sigma^{-1}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right] \\
&= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(|\Sigma|\right) - .5\,\mathrm{tr}\left[\Sigma^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right]
\end{aligned}
$$

Note Σ is a symmetric positive definite matrix. Thus $\Sigma = T^t T$ therefore

$$
\begin{aligned}
L(\Theta; \mathbf{X}) &= -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\left(\left|T^t T\right|\right) - .5\,\mathrm{tr}\left[(T^t T)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right] \\
&= -\frac{nd}{2}\log(2\pi) - n\log\left(|T|\right) - .5\,\mathrm{tr}\left[(T^t T)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right]
\end{aligned}
$$

Remember

How do we analytically solve for an optimum?

- Take derivative of function wrt each variable.

Remember

How do we analytically solve for an optimum?

- Take derivative of function wrt each variable.
- Set each derivative to zero.

Remember

How do we analytically solve for an optimum?

- Take derivative of function wrt each variable.
- Set each derivative to zero.
- Solve the set of simultaneous equations if possible.

MLE for a Normal distribution

For our Normal distribution

$$L(\Theta; \mathbf{X}) = -\frac{nd}{2}\log(2\pi) - n\log(|T|) - .5\,\mathrm{tr}\left[(T^t T)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right]$$

Take derivative of function wrt each variable:

$$\frac{\partial\, L(\Theta; \mathbf{X})}{\partial\, \boldsymbol{\mu}} = \sum_{i=1}^{n} \Sigma^{-1}(\mathbf{x}_i - \boldsymbol{\mu})$$

$$\frac{\partial\, L(\Theta; \mathbf{X})}{\partial\, T} = -nT^{-t} + T(T^t T)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t (T^t T)^{-1}$$

Remember: *The Matrix Cookbook* is your friend.

MLE for a Normal distribution

For our Normal distribution

$$L(\Theta; \mathbf{X}) = -\frac{nd}{2}\log(2\pi) - n\log(|T|) - .5\,\mathrm{tr}\left[(T^t T)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right]$$

Set each derivative to zero:

$$\mathbf{0} = \Sigma^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})$$

$$\mathbf{0} = -nT^{-t} + T(T^t T)^{-1}\left[\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right](T^t T)^{-1}$$

Remember: *The Matrix Cookbook* is your friend.

MLE for a Normal distribution

For our Normal distribution

$$L(\Theta; \mathbf{X}) = -\frac{nd}{2}\log(2\pi) - n\log\left(|T|\right) - .5\,\mathrm{tr}\left[\left(T^t T\right)^{-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t\right]$$

Solve the set of simultaneous equations if possible:

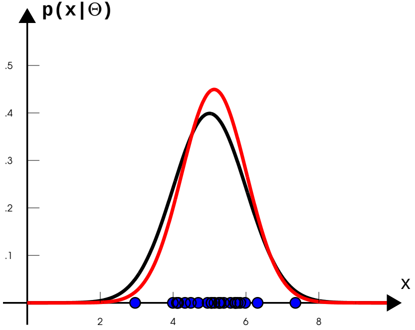
$$\boldsymbol{\mu}^* = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

$$T^{*t}T^* = \Sigma^* = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu}^*)(\mathbf{x}_i - \boldsymbol{\mu}^*)^t$$

Remember: *The Matrix Cookbook* is your friend.

MLE for a Normal distribution

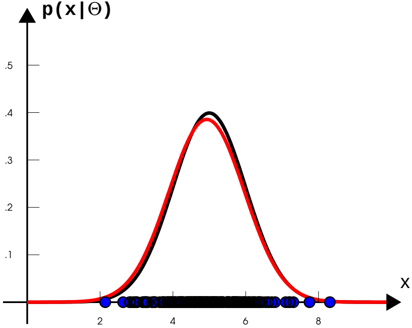
Back to our 1D example:



Red curve is the MLE pdf ($n = 25$)
Black curve is the ground truth

MLE for a Normal distribution

Estimate becomes better as n increases



Red curve is the MLE pdf ($n = 200$)
Black curve is the ground truth

Bootstrap Vs Maximum Likelihood estimate

- Bootstrap is a computer implementation of maximum likelihood estimation.

Bayesian Methods

- Base calculations on the **posterior distribution** for θ

$$\boxed{p(\theta|\mathbf{Z}) = \frac{p(\mathbf{Z}|\theta)\,p(\theta)}{\int p(\mathbf{Z}|\theta')\,p(\theta')\,\mathrm{d}\theta'}}$$

- Use the posterior to estimate the **predictive distribution** for z^{new}

$$\boxed{p(z^{\text{new}}|\mathbf{Z}) = \int p(z^{\text{new}}|\theta)\,p(\theta|\mathbf{Z})\,\mathrm{d}\theta}$$

- This is in contrast to the ML approach which would use $p(z^{\text{new}}|\hat{\theta}_{\text{MLE}})$.

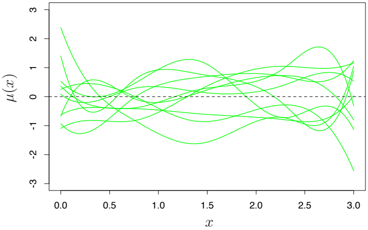
Bayesian approach to 1D smoothing example

- Have observed data $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Assume

$$Y = \sum_{j=1}^{J} \beta_j \, h_j(X) + \epsilon \qquad \text{with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Put a prior on the $\beta = (\beta_1, \ldots, \beta_p)^t$

$$\beta \sim \mathcal{N}(0, \tau^2 I_p)$$



Bayesian approach to 1D smoothing example

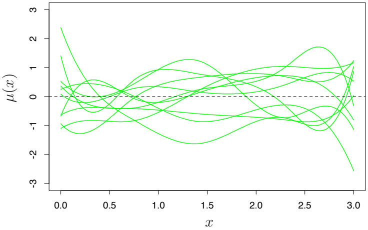
- Have observed data $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- Assume

$$Y = \sum_{j=1}^{J} \beta_j \, h_j(X) + \epsilon \qquad \text{with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Put a prior on the $\beta = (\beta_1, \ldots, \beta_p)^t$

$$\beta \sim \mathcal{N}(0, \tau^2 I_p)$$



Bayesian approach to 1D smoothing example

- The posterior distribution for β is then

$$p(\beta \,|\, \mathbf{Z}) = p(\beta \,|\, \mathbf{X}, y) = \frac{p(y \,|\, \mathbf{X}, \beta)\, p(\beta)}{p(y \,|\, \mathbf{X})}$$

where

$$p(y \,|\, \mathbf{X}, \beta) = \mathcal{N}(y\,;\, \mathbf{H}\beta, \sigma^2 I_n) \quad \text{with } \mathbf{H} = \{h_j(x_i)\}$$

$$\textbf{and} \quad \beta \sim \mathcal{N}(0, \tau^2 I_p)$$

- As have Normal distributions for the likelihood and prior

$$p(\beta \,|\, \mathbf{Z}) = \mathcal{N}(\beta; A^{-1}\, \mathbf{H}^t y,\ A^{-1}\sigma^2)$$

with $A = \mathbf{H}^t\mathbf{H} + \dfrac{\sigma^2}{\tau^2}\, I_p$.

Distribution of the prediction at x_*

- The distribution of the predicted curve at $\mu(x)$

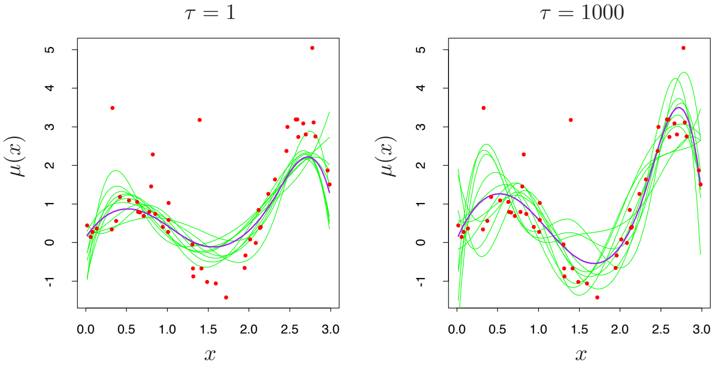
$$
\begin{aligned}
p(y_* \,|\, x_*, \mathbf{Z}) &= \int p(y_* \,|\, x_*, \beta)\, p(\beta \,|\, \mathbf{Z})\, \mathrm{d}\beta \\
&= \int \mathcal{N}(y_* \,;\, h(x_*)^t \beta, \sigma^2)\, \mathcal{N}(\beta \,;\, A^{-1}\mathbf{H}^t y, A^{-1}\sigma^2)\, \mathrm{d}\beta \\
&= \mathcal{N}(y_* \,;\, \mu_{x_*}, \sigma^2_{x_*})
\end{aligned}
$$

  where

$$
\mu_{x_*} = h(x_*)^t\, A^{-1}\mathbf{H}^t y, \qquad \sigma^2_{x_*} = h(x_*)^t\, A^{-1}\, h(x_*) + \sigma^2
$$

- Can re-write these terms μ_{x_*} and $\sigma^2_{x_*}$ so that one can use kernels $\implies$ get **Gaussian process regression**.

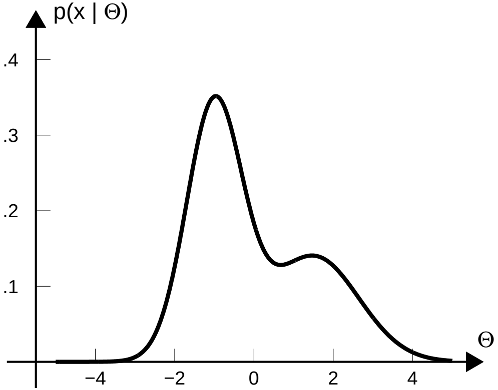
Example curves drawn from the posterior distribution



The EM algorithm

Limitations of Normal distributions

Unfortunately Normal distributions are not very expressive.



What do we do in this situation ??

Gaussian Mixture Models (GMM)

They can accurately represent any distribution.

Mathematical definition

$$p(\mathbf{x} \mid \Theta) = \sum_{k=1}^{K} \pi_k \, N(\mathbf{x}_k; \boldsymbol{\mu}_k, \Sigma_k)$$

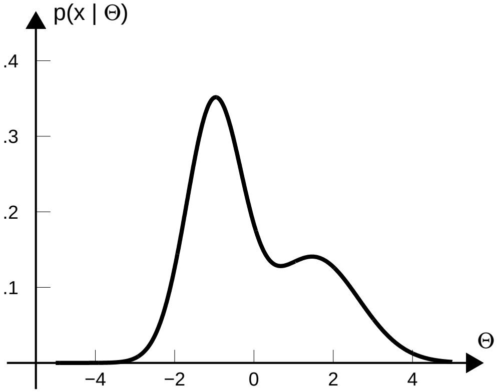
where

$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \text{ for } k = 1, \ldots, K$$

and $\Theta = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \Sigma_1, \ldots, \Sigma_K, \pi_1, \ldots, \pi_K)$

Gaussian Mixture Models (GMM)

They can accurately represent any distribution.

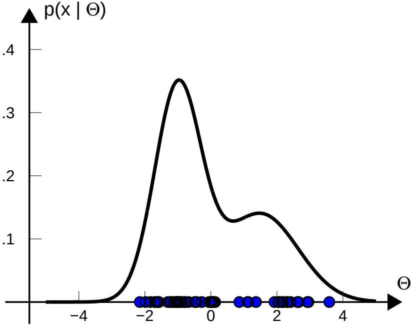


$$p(x \mid \Theta) = \alpha \, \mathcal{N}(x \mid \mu_1, \sigma_1^2) + (1 - \alpha) \, \mathcal{N}(x \mid \mu_2, \sigma_2^2)$$

$$\Theta = (\alpha, \mu_1, \sigma_1, \mu_2, \sigma_2) = (.6, -1, .5, 1.5, 1.3)$$

Parameter estimation for a GMM

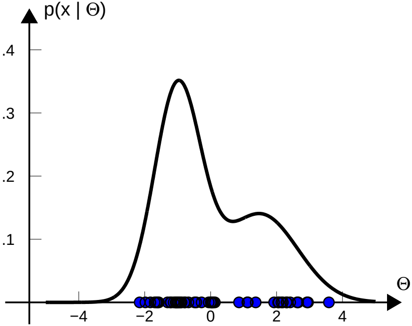
Given n independent samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a GMM.



$\longleftarrow$ **training data**

Parameter estimation for a GMM

Given n independent samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a GMM.



$\longleftarrow$ **training data**

Can still use MLE to estimate Θ from the $\mathbf{x}_i$'s, but...

Attempt 1: Analytic Solution

Attempt 1: Parameter estimation for a GMM

The log-likelihood of the data is

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, \Sigma_k)\right)$$

(Note: We'll assume K is known and fixed.)

Attempt 1: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Let's try to maximize $L(\Theta; \mathbf{X})$ analytically subject to the constraint $\sum_k \pi_k = 1$ and each $\Sigma_k = T_k^t T_k$. Construct the Lagrangian $\mathcal{L}(\Theta, \lambda; \mathbf{X})$.

$$\mathcal{L}(\Theta, \lambda; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right) + \lambda \left(1 - \sum_{k=1}^{K} \pi_k \right)$$

Attempt 1: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k)\right)$$

Let's try to maximize $L(\Theta; \mathbf{X})$ analytically subject to the constraint $\sum_k \pi_k = 1$ and each $\Sigma_k = T_k^t T_k$. Construct the Lagrangian $\mathcal{L}(\Theta, \lambda; \mathbf{X})$.

Take derivatives for $k = 1, \ldots, K$:

$$\frac{\partial\, \mathcal{L}(\Theta, \lambda; \mathbf{X})}{\partial\, \boldsymbol{\mu}_k} = \sum_{i=1}^{n} \frac{\pi_k N(\mathbf{x}_i; \boldsymbol{\mu}_k, T_k^t T_k)}{GMM(\mathbf{x}_i; \Theta)}\, (T_k^t T_k)^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)$$

$$\frac{\partial\, \mathcal{L}(\Theta, \lambda; \mathbf{X})}{\partial\, T_k} = \text{something complicated.....}$$

etc

Attempt 1: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Let's try to maximize $L(\Theta; \mathbf{X})$ analytically subject to the constraint $\sum_k \pi_k = 1$ and each $\Sigma_k = T_k^t T_k$. Construct the Lagrangian $\mathcal{L}(\Theta, \lambda; \mathbf{X})$.

Set derivatives to zero:

$$\sum_{i=1}^{n} \frac{\pi_k N(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)}{GMM(\mathbf{x}_i; \Theta)} \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) = \mathbf{0}$$

etc

Attempt 1: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Let's try to maximize $L(\Theta; \mathbf{X})$ analytically subject to the constraint $\sum_k \pi_k = 1$ and each $\Sigma_k = T_k^t T_k$. Construct the Lagrangian $\mathcal{L}(\Theta, \lambda; \mathbf{X})$.

Solve the set of simultaneous equations

NO ANALYTIC SOLUTION

Attempt 2: Newton based iterative optimzation

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

One iteration

- Have a current estimate $\Theta^{(t)}$.

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

One iteration

- Have a current estimate $\Theta^{(t)}$.
- Approximate $L(\Theta; \mathbf{X})$ in neighbourhood of $\Theta^{(t)}$ with a paraboloid.

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

One iteration

- Have a current estimate $\Theta^{(t)}$.
- Approximate $L(\Theta; \mathbf{X})$ in neighbourhood of $\Theta^{(t)}$ with a paraboloid.
- $\Theta^{(t+1)}$ is set to maximum of the paraboloid.

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

Comments

- Should find a local maximum. ✓

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

Comments

- Should find a local maximum. ✓
- Convergence fast if $\Theta^{(t)}$ close to an optimum. ✓

Attempt 2: Parameter estimation for a GMM

$$L(\Theta; \mathbf{X}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k N(x_i; \boldsymbol{\mu}_k, T_k^t T_k) \right)$$

Could try to maximize $L(\Theta; \mathbf{X})$ iteratively using Newton's Method.

After all $L(\Theta; \mathbf{X})$ is a **scalar valued function** of a **vector** Θ of **variables**.

Comments

- Should find a local maximum. ✓
- Convergence fast if $\Theta^{(t)}$ close to an optimum. ✓
- If $\Theta^{(0)}$ far away from a local maximum method can fail. Paraboloid approximation process can hit problems. ✗

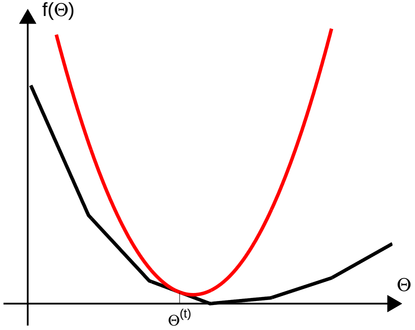
What other options are there??

Now for, what may seem like, a slight diversion

Defintion of Majorization

A function $g(\Theta\,;\,\Theta^{(t)})$ majorizes a function $f(\Theta)$ at $\Theta^{(t)}$ if

$f(\Theta^{(t)}) = g(\Theta^{(t)}\,;\,\Theta^{(t)})$ **and** $f(\Theta) \leq g(\Theta\,;\,\Theta^{(t)})$ for all Θ



$\longleftarrow$ $g(\Theta;\,\Theta^{(t)})$ majorizes $f(\Theta)$

The MM Algorithm

To **minimize** an objective function $f(\Theta)$:

- The MM algorithm is a prescription for constructing optimization algorithms.

Name coined by *David R. Hunter* and *Kenneth Lange*

The MM Algorithm

To **minimize** an objective function $f(\Theta)$:

- The MM algorithm is a prescription for constructing optimization algorithms.
- An MM algorithm creates a surrogate function that majorizes the **objective function**. When the surrogate function is minimized the **objective** function is decreased.

Name coined by *David R. Hunter* and *Kenneth Lange*

The MM Algorithm

To **minimize** an objective function $f(\Theta)$:

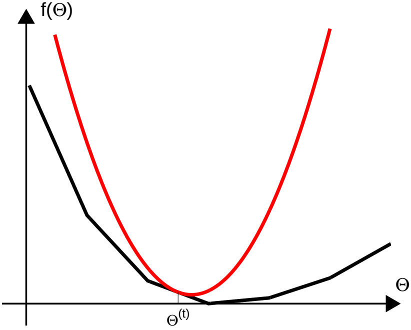
- The MM algorithm is a prescription for constructing optimization algorithms.
- An MM algorithm creates a surrogate function that majorizes the **objective function**. When the surrogate function is minimized the **objective** function is decreased.
- When minimizing MM $\equiv$ majorize/minimize.

Name coined by *David R. Hunter* and *Kenneth Lange*

Some definitions

A function $g(\Theta; \Theta^{(t)})$ majorizes the function $f(\Theta)$ at $\Theta^{(t)}$ if

$f(\Theta^{(t)}) = g(\Theta^{(t)}; \Theta^{(t)})$ **and** $f(\Theta) \le g(\Theta; \Theta^{(t)})$ for all Θ

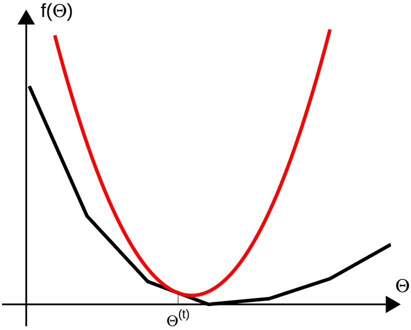


$\longleftarrow$ $g(\Theta; \Theta^{(t)})$ majorizes $f(\Theta)$

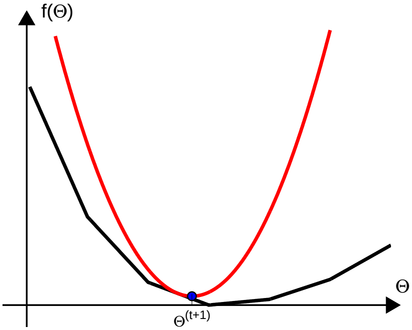
Some definitions

Let

$$\Theta^{(t+1)} = \arg\min_{\Theta} g(\Theta; \Theta^{(t)})$$



Majorize function



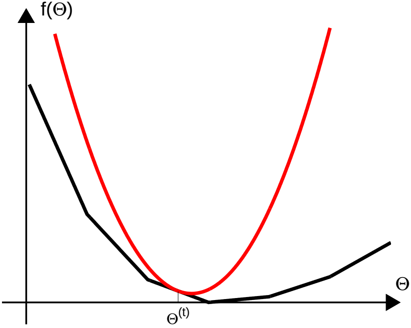
Find minimum of majorizing function

Some definitions

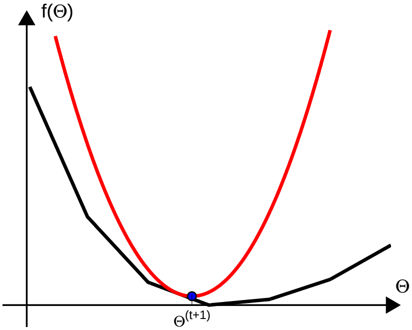
Let

$$\Theta^{(t+1)} = \arg\min_{\Theta} g(\Theta;\, \Theta^{(t)})$$

(so should choose a $g(\Theta;\, \Theta^{(t)})$ which is easy to minimize)



Majorize function



Find minimum of majorizing function

Descent Properties

MM minimization algorithm satisfies the descent property as

$$
\begin{aligned}
f(\Theta^{(t+1)}) &\leq g(\Theta^{(t+1)}; \Theta^{(t)}), \quad \text{as } f(\Theta) \leq g(\Theta; \Theta^{(t)}) \ \forall\Theta \\
&\leq g(\Theta^{(t)}; \Theta^{(t)}), \quad \text{as } \Theta^{(t+1)} \text{ minimizes } g(\Theta; \Theta^{(t)}) \\
&= f(\Theta^{(t)})
\end{aligned}
$$

In summary

$$\boxed{f(\Theta^{(t+1)}) \leq f(\Theta^{(t)})}$$

Descent Properties

MM minimization algorithm satisfies the descent property as

$$\begin{aligned} f(\Theta^{(t+1)}) &\leq g(\Theta^{(t+1)}; \Theta^{(t)}), \quad \text{as } f(\Theta) \leq g(\Theta; \Theta^{(t)}) \ \forall \Theta \\ &\leq g(\Theta^{(t)}; \Theta^{(t)}), \quad \text{as } \Theta^{(t+1)} \text{ minimizes } g(\Theta; \Theta^{(t)}) \\ &= f(\Theta^{(t)}) \end{aligned}$$

In summary

$$\boxed{f(\Theta^{(t+1)}) \leq f(\Theta^{(t)})}$$

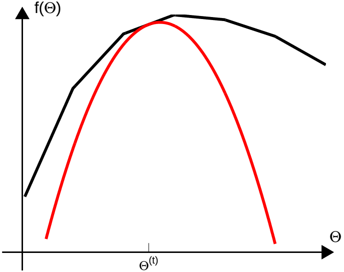
The descent property makes the MM algorithm very stable.

Algorithm converges to local minima or saddle point.

Maximizing a function

To **maximize** an objective function $f(\Theta)$:

- MM algorithm creates a surrogate function that minorize the **objective function**. When the surrogate function is maximized the **objective** function is **increased**.

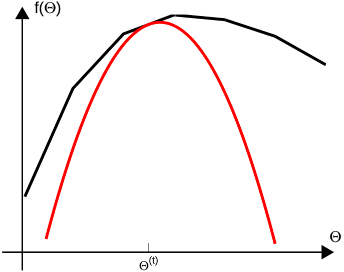


Red curve minorize the black curve

Maximizing a function

To **maximize** an objective function $f(\Theta)$:

- MM algorithm creates a surrogate function that minorize the **objective function**. When the surrogate function is maximized the **objective** function is **increased**.



Red curve minorize the black curve

- When maximizing MM $\equiv$ minorize/maximize.

Big Question?

How do you **majorize** or **minorize** a function??

Here are some generic tricks and tools

- Jensen's inequality
- Chord above the graph property of a convex function
- Supporting hyperplane property of a convex function
- Quadratic upper bound principle
- Arithmetic-geometric mean inequality
- The Cauchy-Schwartz inequality

Presume it would take some practice to use these tricks.

But....

But wait...

You probably have minorized via **Jensen's Inequality**!

Remember **Jensen's Inequality:**

- $h(\cdot)$ be a concave function,

But wait...

You probably have minorized via **Jensen's Inequality**!

Remember **Jensen's Inequality:**

- $h(\cdot)$ be a concave function,
- have K non-negative numbers $\pi_1, \ldots, \pi_K$ with $\sum_k \pi_i = 1$,

But wait...

You probably have minorized via **Jensen's Inequality**!

Remember **Jensen's Inequality:**

- $h(\cdot)$ be a concave function,
- have K non-negative numbers $\pi_1, \ldots, \pi_K$ with $\sum_k \pi_i = 1$,
- K arbitrary numbers $a_1, \ldots, a_K$

But wait...

You probably have minorized via **Jensen's Inequality**!

Remember **Jensen's Inequality:**

- $h(\cdot)$ be a concave function,
- have K non-negative numbers $\pi_1, \ldots, \pi_K$ with $\sum_k \pi_i = 1$,
- K arbitrary numbers $a_1, \ldots, a_K$

then

$$h\left(\sum_{k=1}^{K} \pi_k \, a_k\right) \geq \sum_{k=1}^{K} \pi_k \, h(a_k)$$

Finally we're getting to **E**xpectation**M**aximization

- The EM algorithm is a MM algorithm.

Finally we're getting to **E**xpectation**M**aximization

- The EM algorithm is a MM algorithm.
- Use Jensen's inequality to minorize the log-likelihood.

Finally we're getting to **E**xpectation**M**aximization

- The EM algorithm is a MM algorithm.
- Use Jensen's inequality to minorize the log-likelihood.

Here's how we minorize. Step 1:

$$L(\Theta; \mathbf{X}) = \log\left(p(\mathbf{X} \mid \Theta\right) = \log\left(\sum_{j=1}^{n_z} p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta\right)\right) \quad \leftarrow \text{introduce discrete variable } \mathbf{Z}$$

$$f^{(t)}(\mathbf{Z}) \text{ a pdf} \rightarrow = \log\left(\sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \frac{p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta\right)}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)$$

$$\text{Jensen's inequality} \rightarrow \geq \sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \log\left(\frac{p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta\right)}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)$$

Finally we're getting to **E**xpectation**M**aximization

- The EM algorithm is a MM algorithm.
- Use Jensen's inequality to minorize the log-likelihood.

Here's how we minorize. Step 1:

$$L(\Theta; \mathbf{X}) = \log\left(p(\mathbf{X} \,|\, \Theta\right) = \log\left(\sum_{j=1}^{n_z} p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \,|\, \Theta\right)\right) \quad \leftarrow \text{introduce discrete variable } \mathbf{Z}$$

$$f^{(t)}(\mathbf{Z}) \text{ a pdf} \rightarrow = \log\left(\sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \frac{p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \,|\, \Theta\right)}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)$$

$$\text{Jensen's inequality} \rightarrow \geq \sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \log\left(\frac{p\left(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \,|\, \Theta\right)}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)$$

$$\boxed{L(\Theta; \mathbf{X}) \geq \sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \log\left(\frac{p(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \,|\, \Theta)}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)}$$

Find $f^{(t)}(\mathbf{Z})$

Here's how we minorize. Step 2:

The lower bound must touch the log-likelihood at $\Theta^{(t)}$

$$L(\Theta^{(t)}; \mathbf{X}) = \sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \log \left(\frac{p(\mathbf{X}, \mathbf{Z}=\mathbf{z}_j \mid \Theta^{(t)})}{f^{(t)}(\mathbf{Z}=\mathbf{z}_j)} \right)$$

Find $f^{(t)}(\mathbf{Z})$

Here's how we minorize. Step 2:

The lower bound must touch the log-likelihood at $\Theta^{(t)}$

$$L(\Theta^{(t)}; \mathbf{X}) = \sum_{j=1}^{n_z} f^{(t)}(\mathbf{Z} = \mathbf{z}_j) \log\left(\frac{p(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta^{(t)})}{f^{(t)}(\mathbf{Z} = \mathbf{z}_j)}\right)$$

From this constraint can calculate $f^{(t)}(\mathbf{Z})$. It is:

$$f^{(t)}(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \Theta^{(t)})$$

(Derivation is straight-forward)

EM as MM summary

The log-likelihood function $L(\Theta; \mathbf{X})$ at $\Theta^{(t)}$ is minorized by

$$g(\Theta; \Theta^{(t)}) = \sum_{j=1}^{n_z} p(\mathbf{Z} = \mathbf{z}_j \mid \mathbf{X}, \Theta^{(t)}) \log \left(\frac{p(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta)}{p(\mathbf{Z} = \mathbf{z}_j \mid \mathbf{X}, \Theta^{(t)})} \right)$$

EM as MM summary

The log-likelihood function $L(\Theta; \mathbf{X})$ at $\Theta^{(t)}$ is minorized by

$$g(\Theta; \Theta^{(t)}) = \sum_{j=1}^{n_z} p(\mathbf{Z} = \mathbf{z}_j \mid \mathbf{X}, \Theta^{(t)}) \log \left(\frac{p(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta)}{p(\mathbf{Z} = \mathbf{z}_j \mid \mathbf{X}, \Theta^{(t)})} \right)$$

Maximizing the surrogate function, $g(\Theta; \Theta^{(t)})$, involves:

$$\begin{aligned}
\Theta^{(t+1)} &= \arg\max_{\Theta} \; g(\Theta; \Theta^{(t)}) \\
&= \arg\max_{\Theta} \; \sum_{j=1}^{n_z} p(\mathbf{Z} = \mathbf{z}_j \mid \mathbf{X}, \Theta^{(t)}) \log \left(p(\mathbf{X}, \mathbf{Z} = \mathbf{z}_j \mid \Theta) \right) \\
&= \overbrace{\arg\max_{\Theta} \; \underbrace{E_{p(\mathbf{Z} \mid \mathbf{X}, \Theta^{(t)})} \left[\log \left(p(\mathbf{X}, \mathbf{Z} \mid \Theta) \right) \right]}_{\text{Expectation Step}}}^{\text{Maximization Step}}
\end{aligned}$$

The latent/hidden variables **Z**

There seemed to be some magic in this derivation!

What are the **Z**'s and where did they come from??

Answer:

The latent/hidden variables $\mathbf{Z}$

There seemed to be some magic in this derivation!

What are the $\mathbf{Z}$'s and where did they come from??

Answer:

- $\mathbf{Z}$ is a random variable whose pdf conditioned on $\mathbf{X}$ is completely determined by Θ.

The latent/hidden variables $\mathbf{Z}$

There seemed to be some magic in this derivation!

What are the $\mathbf{Z}$'s and where did they come from??

Answer:

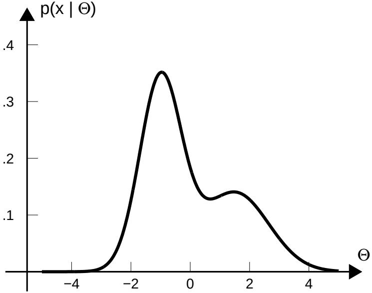
- $\mathbf{Z}$ is a random variable whose pdf conditioned on $\mathbf{X}$ is completely determined by Θ.
- Choice of $\mathbf{Z}$ should make the maximization step **easy**.

Back to our GMM parameter estimation and EM

Attempt 3: Parameter estimation for a GMM

Let's look at a tutorial example using EM:

$$p(x \mid \Theta) = \alpha \, \mathcal{N}(x \mid \mu_1, \sigma_1^2) + (1 - \alpha) \, \mathcal{N}(x \mid \mu_2, \sigma_2^2)$$

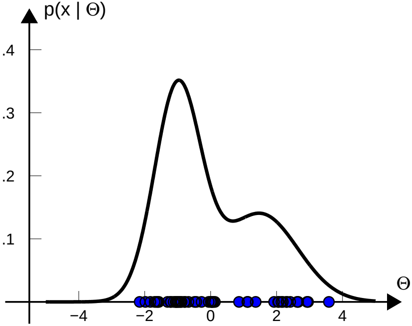


$\longleftarrow$ **Ground truth distribution**

where $\Theta = (\alpha, \mu_1, \sigma_1, \mu_2, \sigma_2) = (.6, -1, .5, 1.5, 1.3)$

Attempt 3: Parameter estimation for a GMM

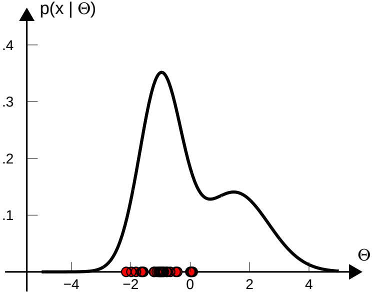
Say all the parameters of Θ are known except α. Then we are given n samples $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ independently drawn from $p(x \mid \Theta)$. Using these samples and EM we can estimate α.



$\longleftarrow$ **training data**

Attempt 3: Parameter estimation for a GMM

If we knew which samples were generated by which component, life would be so much simpler!



Component 1 samples **Component 2 samples**

Attempt 3: EM Solution

Introduce hidden/latent variables:

$\mathbf{Z} = (z_1, \ldots, z_n)$ is a vector of hidden variables.

Each $z_i \in \{0, 1\}$ indicates component generating x_i.

Attempt 3: EM Solution

Introduce hidden/latent variables:

$\mathbf{Z} = (z_1, \ldots, z_n)$ is a vector of hidden variables.

Each $z_i \in \{0, 1\}$ indicates component generating x_i.

E-step:

- Update posteriors for the hidden variables:

$$p(z_i = 0 \,|\, x_i, \alpha^{(t)}) = \frac{p(x_i \,|\, \mu_1, \sigma_1)\, \alpha^{(t)}}{p(x_i \,|\, \mu_1, \sigma_1)\, \alpha^{(t)} + p(x_i \,|\, \mu_2, \sigma_2)\,(1 - \alpha^{(t)})}$$

Attempt 3: EM Solution

Introduce hidden/latent variables:

$\mathbf{Z} = (z_1, \ldots, z_n)$ is a vector of hidden variables.

Each $z_i \in \{0, 1\}$ indicates component generating x_i.

E-step:

- Update posteriors for the hidden variables:

$$p(z_i = 0 \,|\, x_i, \alpha^{(t)}) = \frac{p(x_i \,|\, \mu_1, \sigma_1)\, \alpha^{(t)}}{p(x_i \,|\, \mu_1, \sigma_1)\, \alpha^{(t)} + p(x_i \,|\, \mu_2, \sigma_2)\,(1 - \alpha^{(t)})}$$

- Calculate the conditional expectation

$$g(\alpha; \alpha^{(t)}) = \sum_{\text{all } \mathbf{Z}} p(\mathbf{Z} \,|\, \mathbf{X}, \alpha^{(t)}) \, \log\left(\frac{p(\mathbf{X}, \mathbf{Z} \,|\, \alpha)}{p(\mathbf{Z} \,|\, \mathbf{X}, \alpha^{(t)})}\right)$$

Attempt 3: EM Solution

Introduce hidden/latent variables:

$\mathbf{Z} = (z_1, \ldots, z_n)$ is a vector of hidden variables.

Each $z_i \in \{0, 1\}$ indicates component generating x_i.

E-step:

- Update posteriors for the hidden variables:

$$p(z_i = 0 \mid x_i, \alpha^{(t)}) = \frac{p(x_i \mid \mu_1, \sigma_1)\, \alpha^{(t)}}{p(x_i \mid \mu_1, \sigma_1)\, \alpha^{(t)} + p(x_i \mid \mu_2, \sigma_2)\, (1 - \alpha^{(t)})}$$

- Calculate the conditional expectation

$$g(\alpha; \alpha^{(t)}) = \sum_{\text{all } \mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)}) \, \log \left(\frac{p(\mathbf{X}, \mathbf{Z} \mid \alpha)}{p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)})} \right)$$

M-step: Find $\arg\max_{\alpha} g(\alpha; \alpha^{(t)})$ which gives:

$$\alpha^{(t+1)} = \frac{\sum_i p(z_i = 0 \mid \mathbf{x}_i, \alpha^{(t)})}{n}$$

Attempt 3: EM expectation calculation

$$\sum_{\text{all } \mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)}) \ \log\left(p(\mathbf{X}, \mathbf{Z} \mid \alpha)\right)$$

$$= \sum_{\text{all } \mathbf{Z}} \left[\prod_{s=1}^{n} p(z_s \mid x_s, \alpha^{(t)}) \sum_{i=1}^{n} \log\left(p(x_i \mid z_i, \alpha)\, p(z_i|\alpha)\right)\right]$$

$$= \sum_{j_1=0}^{1} \cdots \sum_{j_n=0}^{1} \left[\prod_{s=1}^{n} p(z_s = j_s \mid x_s, \alpha^{(t)}) \sum_{i=1}^{n} \log\left(p(x_i \mid z_i = j_i, \alpha)\, p(z_i = j_i|\alpha)\right)\right]$$

$$= \sum_{i=1}^{n} \left[\left(\prod_{s=1, s\neq i}^{n} \underbrace{\sum_{j_s=0}^{1} p(z_s = j_s \mid x_s, \alpha^{(t)})}_{=1}\right) p(z_i = j_i \mid x_i, \alpha^{(t)}) \log\left(p(x_i \mid z_i = j_i, \alpha)\, p(z_i = j_i|\alpha)\right)\right]$$

$$= \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \log\left(p(x_i \mid z_i = j_i, \alpha)\, p(z_i = j_i|\alpha)\right)$$

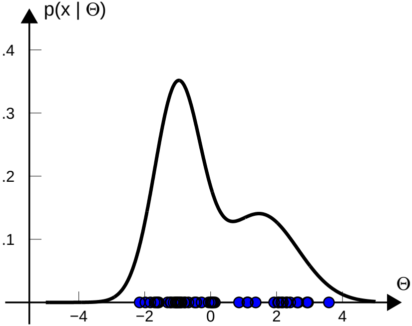
$$= \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \log\left(N(x_i \mid \mu_{j_i}, \sigma_{j_i})\, \alpha^{1-j_i} (1-\alpha)^{j_i}\right)$$

Attempt 3: EM maximization process

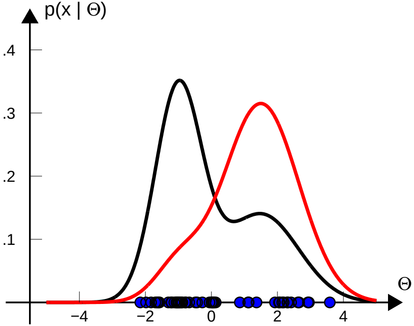
$$
\begin{aligned}
\frac{\partial \sum_{\text{all } \mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)}) \; \log\left(p(\mathbf{X}, \mathbf{Z} \mid \alpha)\right)}{\partial \, \alpha} &= \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \, \frac{\partial \, \log\left(\alpha^{1-j_i}(1-\alpha)^{j_i}\right)}{\partial \, \alpha} \\
&= \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \, \left(\frac{1-j_i}{\alpha} - \frac{j_i}{1-\alpha}\right) \\
&= \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \, (1 - j_i - \alpha) \\
&= (1-\alpha) \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) - \sum_{i=1}^{n} \sum_{j_i=0}^{1} p(z_i = j_i \mid x_i, \alpha^{(t)}) \, j_i \\
&= n(1-\alpha) - \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) \\
&= -n \, \alpha + n - \sum_{i=1}^{n} (1 - p(z_i = 0 \mid x_i, \alpha^{(t)})) \\
&= \sum_{i=1}^{n} p(z_i = 0 \mid x_i, \alpha^{(t)}) - n \, \alpha = 0
\end{aligned}
$$

Therefore $\alpha^{(t+1)} = \frac{\sum_{i=1}^{n} p(z_i = 0 \mid x_i, \alpha^{(t)})}{n}$

Attempt 3: EM Solution starting point



Ground truth distribution



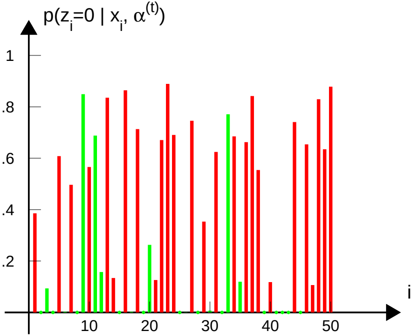
Initial guess of distribution with $\alpha^{(0)} = .1$

Remember $g(\alpha; \alpha^{(t)})$ minorizes $\log\left(p(\mathbf{X} \mid \alpha)\right)$ at $\alpha^{(t)}$.

Let's plot what happens as EM update $\alpha^{(t)}$...

EM one iteration

Compute posterior probabilities of the hidden variables



Graph shows $p(z_i = 0 \,|\, x_i, \alpha^{(0)})$ of each hidden variable.

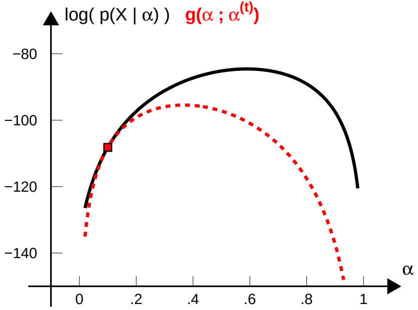
Red $\Longrightarrow$ sample really generated by component 1

Green $\Longrightarrow$ sample really generated by component 2

EM one iteration

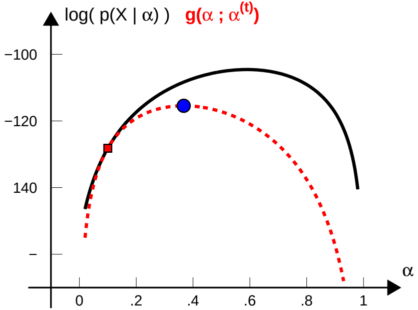
Compute the expectation minorizing the log-likelihood at $\alpha^{(0)} = .1$

$$g(\alpha; \alpha^{(t)}) = \sum_{\text{all } \mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)}) \, \log\left(\frac{p(\mathbf{X}, \mathbf{Z} \mid \alpha)}{p(\mathbf{Z} \mid \mathbf{X}, \alpha^{(t)})}\right)$$



EM one iteration

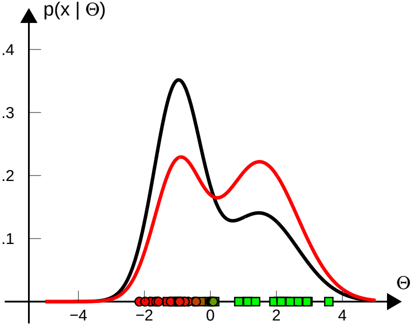
Calculate maximum of $g(\alpha; \alpha^{(0)})$



Maximum of $g(\alpha; \alpha^{(0)})$ gives $\alpha^{(1)} = .3672$

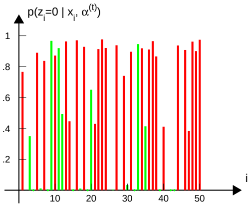
EM one iteration

The estimate of the GMM with $\alpha^{(1)} = .3672$

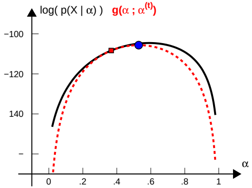


EM Iterations

Iteration 2

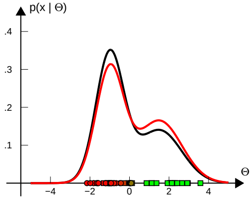


Posterior probabilities



$g(\alpha; \alpha^{(1)})$

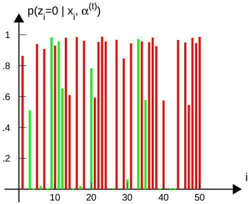
$\alpha^{(2)} = .5287$



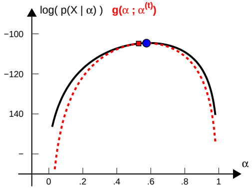
Current GMM estimate

EM Iterations

Iteration 3

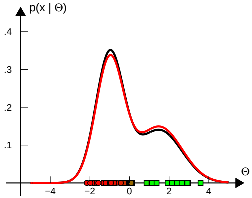


Posterior probabilities



$g(\alpha; \alpha^{(2)})$

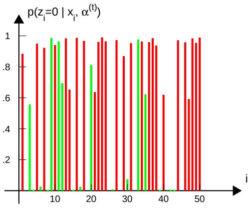
$\alpha^{(3)} = .5748$



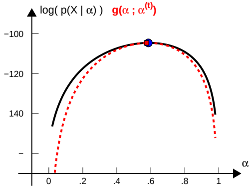
Current GMM estimate

EM Iterations

Iteration 4

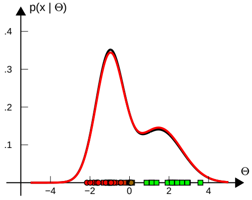


Posterior probabilities



$g(\alpha; \alpha^{(3)})$

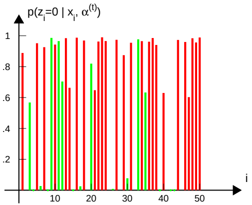
$\alpha^{(4)} = .5859$



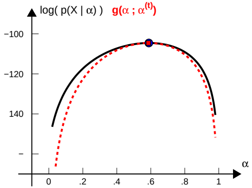
Current GMM estimate

EM Iterations

Iteration 5

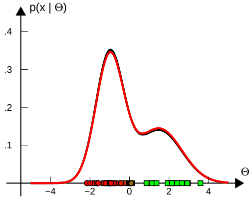


Posterior probabilities



$g(\alpha; \alpha^{(4)})$

$\alpha^{(5)} = .5885$



Current GMM estimate

MCMC for Sampling from the Posterior

Aim:

- Generate independent samples $\{x^{(r)}\}_{r=1}^{R}$ from a pdf $p(x)$.
- Can then use $x^{(r)}$'s to estimate expectations of functions under this distribution

$$\mathrm{E}[\phi(x)] = \int_x \phi(x)\, p(x)\, \mathrm{d}x \approx \frac{1}{R} \sum_{r=1} \phi(x^{(r)})$$

Not an easy task:

- Sampling from $p(x)$ is, in general, hard.
- Especially when $x \in \mathbb{R}^p$ and p is large.

Common approach:

- **Monte Carlo Markov Chain** methods such as *Metropolis-Hastings* and *Gibbs sampling*.

Assumptions:

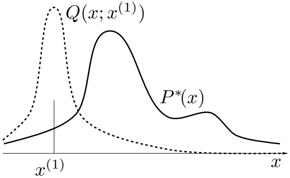
- Want to draw samples from $p(x)$.
- Can evaluate $p(x)$ within a normalization factor.
- That is can evaluate a function $p^*(x)$ such that

$$p(x) = p^*(x)/Z$$

where Z is a constant.

Initially

- Have an initial state $x^{(1)}$.
- Define a **proposal density** $Q(x'; x^{(t)})$ depending on the current state $x^{(t)}$.



- Must be able to draw samples from $Q(x'; x^{(t)})$.

At each iteration

- A tentative new state x' is generated from the proposal density $Q(x'; x^{(t)})$.
- Compute

$$a = \min\left(1, \frac{p^*(x')\, Q(x^{(t)}; x')}{p^*(x^{(t)})\, Q(x'; x^{(t)})}\right)$$

- Accept new state x' with probability a.
- Set

$$x^{(t+1)} = \begin{cases} x' & \text{if state is accepted} \\ x^{(t)} & \text{if state is \textbf{not} accepted} \end{cases}$$

At each iteration

- A tentative new state x' is generated from the proposal density $Q(x'; x^{(t)})$.
- Compute

$$a = \min\left(1, \frac{p^*(x')\, Q(x^{(t)}; x')}{p^*(x^{(t)})\, Q(x'; x^{(t)})}\right)$$

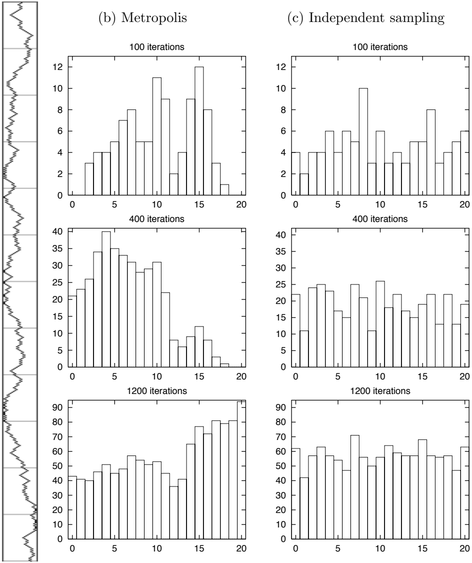
- Accept new state x' with probability a.
- Set

$$x^{(t+1)} = \begin{cases} x' & \text{if state is accepted} \\ x^{(t)} & \text{if state is } \textbf{not} \text{ accepted} \end{cases}$$

Convergence:
For any Q s.t. $Q(x'; x) > 0 \; \forall x, x'$, as $t \to \infty$

the probability distribution of $x^{(t)}$ tends to $p(x) = p^*(x)/Z$.

Example of $x^{(t)}$ for a simple toy example



In Gibbs sampling given a state $x^{(t)} \in \mathbb{R}^p$ generate a new state with

$$x_1^{(t+1)} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots, x_p^{(t)})$$

$$x_2^{(t+1)} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)}),$$

$$x_3^{(t+1)} \sim p(x_2 | x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t)}, \ldots, x_p^{(t)}), \text{etc.}$$

where it is assumed we can generate samples from $p(x_i | \{x_j\}_{j \neq i})$.

In Gibbs sampling given a state $x^{(t)} \in \mathbb{R}^p$ generate a new state with

$$x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \dots, x_p^{(t)})$$

$$x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \dots, x_p^{(t)}),$$

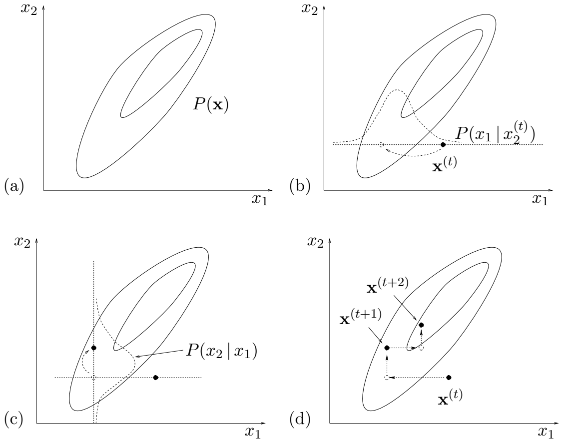
$$x_3^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_2^{(t+1)}, x_4^{(t)}, \dots, x_p^{(t)}), \text{etc.}$$

where it is assumed we can generate samples from $p(x_i|\{x_j\}_{j \neq i})$.

Convergence

As Gibbs sampling is a Metropolis method, the probability distribution of $x^{(t)}$ tends to $p(x)$ as $t \to \infty$, as long as $p(x)$ does not have pathological properties.

Gibbs Sampling: Two dimensional example



Evolution of a state x defined by a Markov chain

- **Markov chain** defined by an initial $p^{(0)}(x)$ and a **transition probability** $T(x'; x)$.

- Let $p^{(t)}(x)$ be the pdf of the state after t applications of the Markov chain.

- The pdf of the state at the $(t+1)$th iteration of the Markov chain is given by

$$p^{(t+1)}(x') = \int_x T(x'; x)\, p^{(t)}(x)\, \mathrm{d}x$$

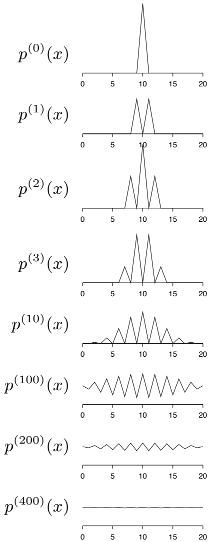
- Want to find a chain s.t. as $t \to \infty$ then $p^{(t)}(x) \to p(x)$.

Example of $p^{(t)}(x)$'s

Transition matrix

$$
\mathbf{T} = \begin{bmatrix}
1/2 & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & \cdot & 1/2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1/2 & 1/2
\end{bmatrix}
$$

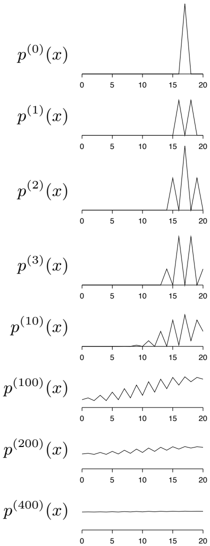
$p^{(t)}(x)$'s



Transition matrix

$$\mathbf{T} = \begin{bmatrix}
1/2&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&\cdot&1/2\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&1/2&1/2
\end{bmatrix}$$

$p^{(t)}(x)$'s



Markov chains for MCMC methods

When designing a MCMC method construct a chain with the following properties

- $p(x)$ is an **invariant distribution** of the chain

$$p(x') = \int_x T(x'; x)\, p(x)\, \mathrm{d}x$$

- The chain is **ergodic** that is

$$p^{(t)}(x) \to p(x) \text{ as } t \to \infty \text{ for any } p^{(0)}(x)$$

Gibbs sampling for mixtures

- Close connection between Gibbs sampling and the EM algorithm in exponential family models.
- Let
  - the parameters, θ, of the distribution **and**
  - the *latent*/*missing* data $\mathbf{Z}^m$

  be parameters for a Gibbs sampler.
- Therefore to estimate the parameters of a GMM at each iteration
  - $\Delta_i^{(t+1)} \sim p(\Delta_i \,|\, \theta^{(t)}, \mathbf{Z})$ for $i = 1, \ldots, n$
  - $\theta^{(t+1)} \sim p(\theta \,|\, \Delta^{(t+1)}, \mathbf{Z})$

  where $\Delta_i \in \{1, \ldots, K\}$ and represents which component training example i is assigned to.

- Close connection between Gibbs sampling and the EM algorithm in exponential family models.

- Let
  - the parameters, θ, of the distribution **and**
  - the *latent*/*missing* data $\mathbf{Z}^m$

  be parameters for a Gibbs sampler.

- Therefore to estimate the parameters of a GMM at each iteration
  - $\Delta_i^{(t+1)} \sim p(\Delta_i \,|\, \theta^{(t)}, \mathbf{Z})$ for $i = 1, \ldots, n$
  - $\theta^{(t+1)} \sim p(\theta \,|\, \Delta^{(t+1)}, \mathbf{Z})$

  where $\Delta_i \in \{1, \ldots, K\}$ and represents which component training example i is assigned to.

Bagging

Starting point

- Have training set $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Let $\hat{f}(x)$ be the prediction at input x learned from $\mathbf{Z}$.

Goal

- Obtain a prediction at input x with lower variance than $\hat{f}(x)$.

Starting point

- Have training set $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Let $\hat{f}(x)$ be the prediction at input x learned from $\mathbf{Z}$.

Goal

- Obtain a prediction at input x with lower variance than $\hat{f}(x)$.

How - Bootstrap aggregation a.k.a. *Bagging*

- Obtain bootstrap samples $\mathbf{Z}^{*1}, \ldots, \mathbf{Z}^{*B}$.
- For each $\mathbf{Z}^{*b}$ fit the model and get prediction $\hat{f}^{*b}(x)$.
- The bagged estimate is then

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

The Bagged Estimate

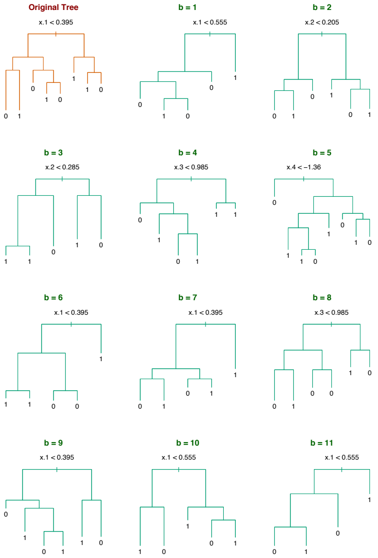
$$\boxed{\hat{f}_{\mathsf{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)}$$

- Remember $\hat{f}(x)$ is the prediction at input x learned from $\mathbf{Z}$.
- $\hat{f}_{\mathsf{bag}}(x)$ differs from $\hat{f}(x)$ when the fitted f is a non-linear or adaptive function of the data.

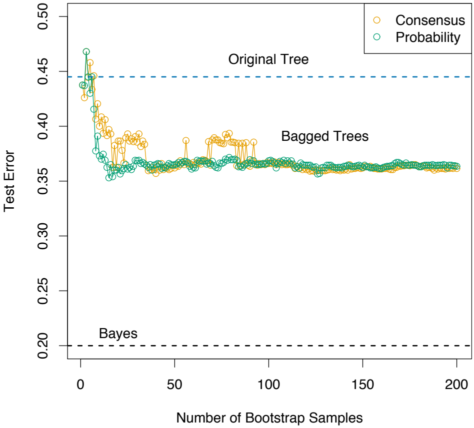
Example when bagging helps significantly

- Have $n = 30$ training examples with two classes and $p = 5$.
- Each feature is $\mathcal{N}(0, 1)$ with pairwise correlations of .95.
- The response Y was generated according to $P(Y = 1|x_1 \leq .5) = .2$ and $P(Y = 1|x_1 > .5) = .8$.
- Test sample of size 2000 was generated.
- The base classifier, $\hat{f}$, is a classification tree.
- $B = 200$

Trees learnt from different bootstrap samples



Bagged tree classifer outperforms one tree classifier



- Bag the $0, 1$ decision returned by each tree.
- Bag the $(P(y = 0|x), P(y = 1|x))$ returned by each tree. Use the ratio of $+$tives to -tives in the terminal node reached by x.

Bagging for classification and $0, 1$ loss

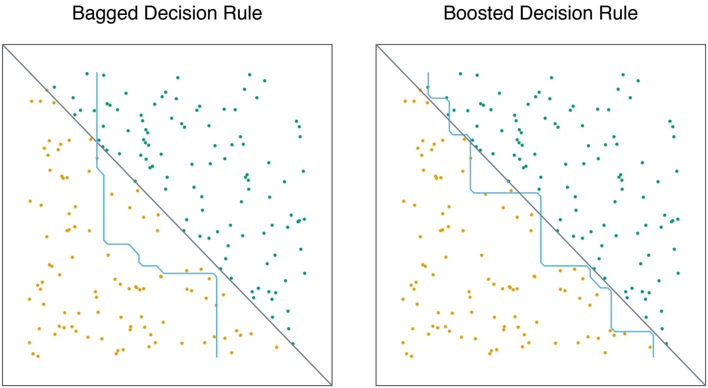
Squared-error loss:

- Bagging can dramatically reduce the variance of unstable procedures, leading to improved prediction.

Classification with $0, 1$ loss

- Bagging a **good** classifier can make it **better**.
- Bagging a **bad** classifier can make things **worse**.
- Can understand the bagging effect in terms of a consensus of independent *weak learners* or the *wisdom of crowds*.

Bagging enlarges the space of models derived from $\hat{f}(x)$



- $\hat{f}(x)$ can either be an oriented vertical or horizontal line.
- In this case bagging the $\hat{f}^{*b}(x)$'s gives some gain but not as much as boosting. $(B = 50)$

Model Averaging and Stacking

Starting point

- Have training set $\mathbf{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- Have a set of candidate models $\mathcal{M}_1, \ldots, \mathcal{M}_M$ to explain $\mathbf{Z}$.

Goal

- Want to estimate quantity ζ - perhaps a prediction of $f(x)$ at x.

A Bayesian solution

- The posterior distribution of ζ is

$$p(\zeta \,|\, \mathbf{Z}) = \sum_{m=1}^{M} p(\zeta \,|\, \mathcal{M}_m, \mathbf{Z})\, P(\mathcal{M}_m \,|\, \mathbf{Z})$$

with posterior mean

$$\mathrm{E}[\zeta \,|\, \mathbf{Z}] = \sum_{m=1}^{M} \mathrm{E}[\zeta \,|\, \mathcal{M}_m, \mathbf{Z}]\, P(\mathcal{M}_m \,|\, \mathbf{Z})$$

$$\boxed{\mathrm{E}[\zeta \,|\, \mathbf{Z}] = \sum_{m=1}^{M} \mathrm{E}[\zeta \,|\, \mathcal{M}_m, \mathbf{Z}] \, P(\mathcal{M}_m \,|\, \mathbf{Z})}$$

- **Committee method** make approximation

$$P(\mathcal{M}_m \,|\, \mathbf{Z}) \approx \frac{1}{M}$$

- **BIC approach** make approximation

$$P(\mathcal{M}_m \,|\, \mathbf{Z}) \approx -2\,\text{loglik} + d_m \log(n)$$

- **Hardcore Bayesian** try to estimate the integral

$$\begin{aligned} P(\mathcal{M}_m \,|\, \mathbf{Z}) &\propto P(\mathcal{M}_m)\, p(\mathbf{Z} \,|\, \mathcal{M}_m) \\ &\propto P(\mathcal{M}_m) \int p(\mathbf{Z} \,|\, \theta_m, \mathcal{M}_m)\, p(\theta_m \,|\, \mathcal{M}_m)\, d\theta_m \end{aligned}$$

Model averaging - Frequentist approach

Starting point

- Have predictions $\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_M(x)$.

Goal

- For squared-error loss find weights $w = (w_1, \ldots, w_M)$ s.t.

$$\hat{w} = \arg\min_w \mathrm{E}_{P_{Y|X=x}} \left[\left(Y - \sum_{m=1}^{M} w_m \, \hat{f}_m(x) \right)^2 \right]$$

Solution if can compute expectations

- Population linear regression of Y on $\hat{F}(x) \equiv [\hat{f}_1(x), \ldots, \hat{f}_M(x)]^t$

$$\hat{w} = \mathrm{E}_P \left[\hat{F}(x) \, \hat{F}(x)^t \right]^{-1} \mathrm{E}_P \left[\hat{F}(x) \, Y \right]$$

(Have dropped the subscript on the distribution P.)

Model averaging - Frequentist approach

For this $\hat{w}$

$$\boxed{\hat{w} = \mathrm{E}_P \left[\hat{F}(x) \, \hat{F}(x)^t \right]^{-1} \mathrm{E}_P \left[\hat{F}(x) \, Y \right]}$$

the full regression model **has smaller error** than any single model

$$\mathrm{E}_P \left[\left(Y - \sum_{m=1}^{M} w_m \, \hat{f}_m(x) \right)^2 \right] \leq \mathrm{E}_P \left[\left(Y - \hat{f}_m(x) \right)^2 \right] \; \forall m$$

Combining models never makes things worse (at a population level)

For this $\hat{w}$

$$\boxed{\hat{w} = \mathrm{E}_P\left[\hat{F}(x)\,\hat{F}(x)^t\right]^{-1}\mathrm{E}_P\left[\hat{F}(x)\,Y\right]}$$

the full regression model **has smaller error** than any single model

$$\mathrm{E}_P\left[\left(Y - \sum_{m=1}^{M} w_m\,\hat{f}_m(x)\right)^2\right] \leq \mathrm{E}_P\left[\left(Y - \hat{f}_m(x)\right)^2\right]\ \forall m$$

Combining models never makes things worse (at a population level)

But cannot estimate the population $\hat{w}$. What is one to do?

Solution: Stacked generalization

- $\hat{f}_m^{-i}(x)$ is the prediction at x using
  - the mth model
  - learnt from the dataset with the ith training example removed.

- Then the **stacking weights** are given by

$$\hat{w}^{\text{st}} = \arg\min_w \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-i}(x_i) \right)^2$$

- The final prediction at point x is

$$\sum_m \hat{w}_m^{\text{st}} \hat{f}_m(x)$$

Solution: Stacked generalization

- $\hat{f}_m^{-i}(x)$ is the prediction at x using
  - the mth model
  - learnt from the dataset with the ith training example removed.

- Then the **stacking weights** are given by

$$\hat{w}^{\text{st}} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-i}(x_i) \right)^2$$

- The final prediction at point x is

$$\sum_{m} \hat{w}_m^{\text{st}} \hat{f}_m(x)$$

Solution: Stacked generalization

- $\hat{f}_m^{-i}(x)$ is the prediction at x using
  - the mth model
  - learnt from the dataset with the ith training example removed.

- Then the **stacking weights** are given by

$$\hat{w}^{\mathsf{st}} = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-i}(x_i) \right)^2$$

- The final prediction at point x is

$$\sum_{m} \hat{w}_m^{\mathsf{st}} \hat{f}_m(x)$$

- Better results by forcing $\hat{w}_m^{\text{st}}$'s to be ≥ 0 and sum to 1.

- Stacking and model selection with via leave-one-out cross-validation are closely related.

- Can apply stacking to other non-linear methods to combine predictions from different models.

Stochastic Search: Bumping

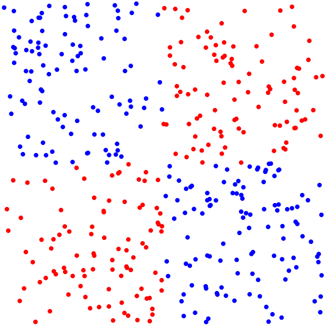
- Draw bootstrap samples $\mathbf{Z}^{*1}, \ldots, \mathbf{Z}^{*B}$.
- for $b = 1, \ldots, B$

  Fit the model to $\mathbf{Z}^{*b}$ giving $\hat{f}^{*b}(x)$.
- Choose the model obtained from bootstrap sample $\hat{b}$ which minimizes training error:

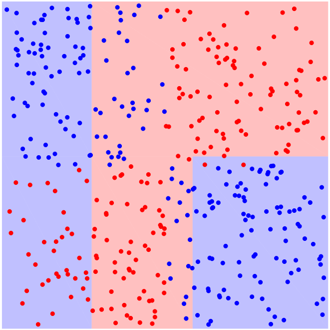
$$\boxed{\hat{b} = \arg\min_{b} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}^{*b}(x_i) \right)^2}$$

The model predictions are then $\hat{f}^{*\hat{b}}(x)$.

Bumping Example: Classification using decision trees



Training data

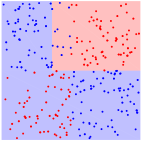


$\hat{f}(x)$ **using all training data**

Forced tree to have at least 80 points in each leaf.

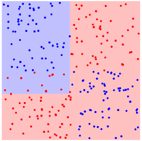
Bumping: Bootstrap sample training data and fit

$\hat{f}^{*1}(x)$



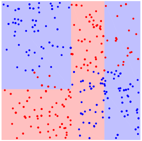
0.2775

$\hat{f}^{*2}(x)$



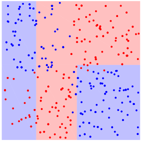
0.3425

$\hat{f}^{*3}(x)$



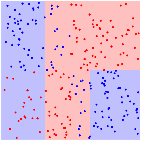
0.2725

$\hat{f}^{*4}(x)$



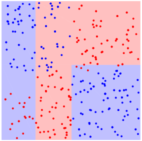
0.2050

$\hat{f}^{*5}(x)$



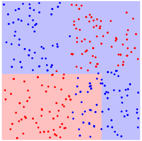
0.2550

$\hat{f}^{*6}(x)$



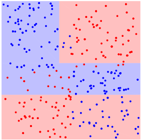
0.1950

$\hat{f}^{*7}(x)$



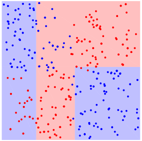
0.3475

$\hat{f}^{*8}(x)$



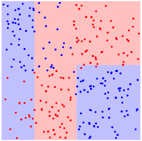
0.2550

$\hat{f}^{*9}(x)$



0.1950

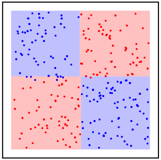
$\hat{f}^{*10}(x)$



0.2100

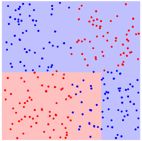
Bumping: Bootstrap sample training data and fit

$\hat{f}^{*11}(x)$



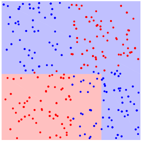
0.0325

$\hat{f}^{*12}(x)$



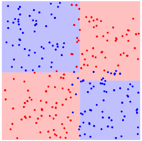
0.3450

$\hat{f}^{*13}(x)$



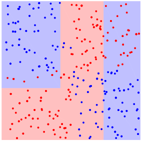
0.3450

$\hat{f}^{*14}(x)$



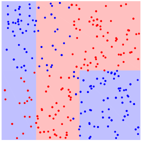
0.1025

$\hat{f}^{*15}(x)$



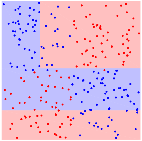
0.2750

$\hat{f}^{*16}(x)$



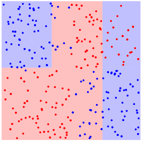
0.2000

$\hat{f}^{*17}(x)$



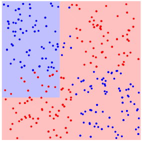
0.3100

$\hat{f}^{*18}(x)$



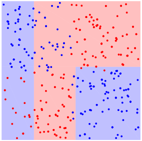
0.2775

$\hat{f}^{*19}(x)$



0.3425

$\hat{f}^{*20}(x)$



0.2000

When & why it works

- Bumping perturbs the training data.
- Therefore explore different areas of the model space.
- Must ensure the complexity of each model fit is comparable.