Mixture Modeling with Latent Variables and the EM Algorithm

Rob McCulloch

1. Deviance, AIC, BIC

AIC, (A information criterion) and BIC (Bayesian information criterion) are widely used for model selection.

Suppose we have a set of candidate models

$\mathcal{M}_m$, $m = 1, 2, \ldots, M$.

Model $\mathcal{M}_m$ has parameter vector θ_m associate with it and

$$p(Z \mid \theta_m, \mathcal{M}_m)$$

represents the model of the data Z under model $\mathcal{M}_m$.

The Deviance:

Let $\hat{\theta_m}$ be the MLE under model $\mathcal{M}_m$.

Let

$$\hat{L}_m = p(Z \mid \hat{\theta}_m, \mathcal{M}_m),$$

the maximized likelihood under model $\mathcal{M}_m$.

Then the deviance is

$$D_m = -2\log(\hat{L}_m).$$

and the BIC is

$$BIC_m = D_m + log(n)\, d_m$$

where d_m is the dimension of θ_m and n is the sample size.

You choose the model with the smallest BIC

$$BIC = D + d\, log(n) = -2 log(\hat{L}) + log(n)\, d$$

The Deviance:

Measures the in sample fit, with a smaller deviance indicating a better fit.

Complexity Penalty:

The term $d\, log(n)$ is a "complexity penalty" in that a higher dimensional parameter θ corresponds to a more complex model. BIC charges you $log(n)$ for a parameter.

As you add parameters, the deviance will go down, but the complexity penalty will go up, giving you a "U".

AIC:

The AIC is “an information criterion” or, “the Akaike information criterion”.

$$AIC = D + d\,2 = -2 log(\hat{L}) + 2\,d$$

Use: Choose the model with the smallest AIC.

The AIC charges you 2 for a parameter!!

Clearly, for non-tiny n, the BIC charges more for a parameter so it will give you a smaller model.

2. Latent Variables and the EM Algorithm

A very general and powerful probabilistic modeling techigue involves the use of *latent variables*.

Suppose we have a vector variable X.

Suppose we want to build a model for X which represents some kind of complex structure.

A general approach to this is to make up a probabilistic model for (Z, X) such that the marginal distribution of X has the desired dependent structure.

The idea is that even though Z may be *latent, unobserved* quantity, it is easier to think about things with Z in the picture.

Example:

Suppose we give a person two different kinds of tests, both of which are different ways of measuring their abilities.

Let $X = (X_1, X_2)$ where X_i is the score on test i.

We might imagine that a person has an unobserved intelligence Z and X is dependent because:

$$\begin{aligned} X_1 &= \alpha_1 + \beta_1 Z + \epsilon_1 \\ X_2 &= \alpha_2 + \beta_2 Z + \epsilon_2 \end{aligned}$$

where the ϵ_i are independent.

This is an example of *factor analyis* in which a high dimensional vector is a linear function of a small set of factors + uncorrelated noise.

We are going to look at the analysis of *mixtures of normals* using latent variables.

We will look at the Expectation-Maximization (**EM**) algorithm which is used for estimation of the models with latent variables.

This is an important special case of the latent variable approach.

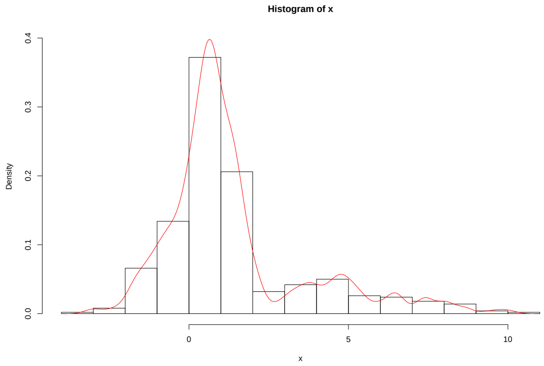
3. Univariate Mixtures of Normals

We are just measuring a single number y.

Often, data y_i, $i = 1, 2 \ldots, n$ does not “look normal”.

Here is some data I simulated with a kernel smooth plotted on top.

does not look normal



A kernel smooth is

$$f(y) = \frac{1}{n} \sum_{i=1}^{n} f(y \mid y_i, \sigma^2)$$

where $f(y \mid \mu, \sigma^2)$ is a normal density.

An alternative, somewhat simpler approach, is to imagine that there is a small number of normals we are mixing together with unequal weights.

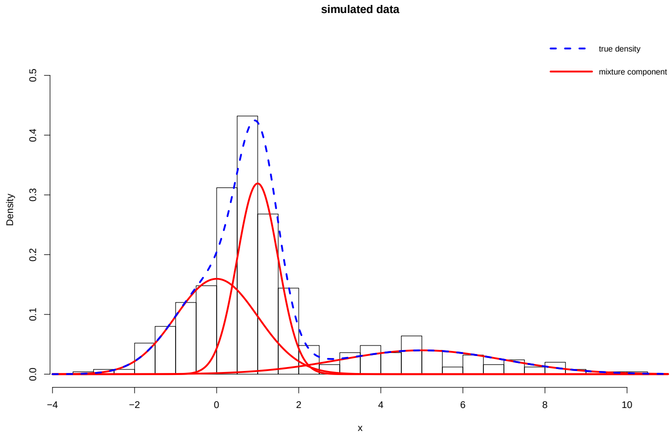
Assume we have J mixture components where each component is a $f(y \mid \mu_j, \sigma_j^2)$ distibution.

Let $\theta_j = (\mu_j, \sigma_j^2)$ and $\theta = (\theta_1, \theta_2, \ldots, \theta_J)$.

Our model is

$$p(y \mid \theta, p) = \sum_{j=1}^{J} p_j \, f(y \mid \theta_j)$$

Here is how I simulated the data.
The red curves are $p_j\, f(y \mid \theta_j), j = 1, 2, 3$
The blue is the sum of the red.



Even though the data looks nothing like "normal" there is a simple underlying structure mixing just three normals.

Here are the mixture weights, means, and standard deviations.

```
> pv
[1] 0.4 0.4 0.2
> mv
[1] 0 1 5
> sv
[1] 1.0 0.5 2.0
```

Can we model real data this way?

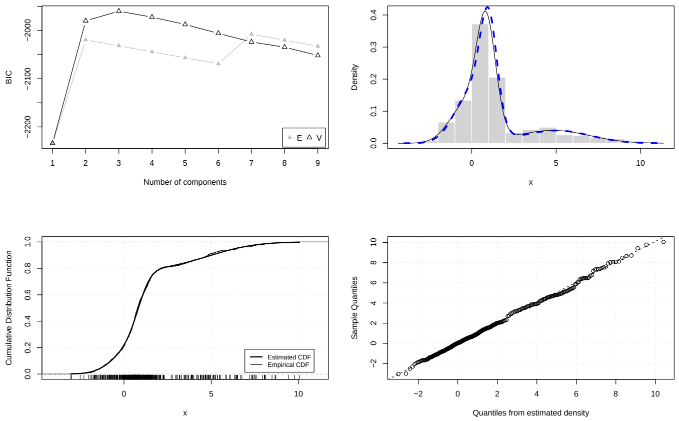
it works amazingly well.

Mixture Model Estimates for the Simulated Data

Using the R package `mclust`:

```
> modsim = densityMclust(x)
> summary(modsim)
-------------------------------------------------------
Density estimation via Gaussian finite mixture modeling
-------------------------------------------------------

Mclust V (univariate, unequal variance) model with 3 components:

 log.likelihood   n df      BIC       ICL
      -955.1265 500  8 -1959.97 -2171.702

Clustering table:
  1   2   3 
108 294  98 
```

`mclust` estimates the number of components using BIC!!.
This is a triumph for BIC !!!



topleft: BIC for different number of components
  E:$\sigma_j = \sigma$, V:σ_j unconstrained.
topright: density estimate (true in blue),
botleft: Empirical CDF vs model CDF,
botright: empirical quantiles vs model quantiles

```
> mvf = modsim$parameters$mean
> svf = sqrt(modsim$parameters$variance$sigmasq)
> pvf = modsim$parameters$pro
> mvf
         1          2          3
-0.2854986  0.8576423  4.8019094
> svf
[1] 0.9986226 0.5637910 2.1545613
> pvf
[1] 0.3031896 0.4795940 0.2172164
> mv
[1] 0 1 5
> sv
[1] 1.0 0.5 2.0
> pv
[1] 0.4 0.4 0.2
```

Note that even though the parameter estimates don't match up perfectly, the density fit is very close !!!

Clustering:

Given the estimated θ_j and p_j, how do we get the clustering??

We imagine that each particular observation is generated by one of the mixture components and then infer the component.

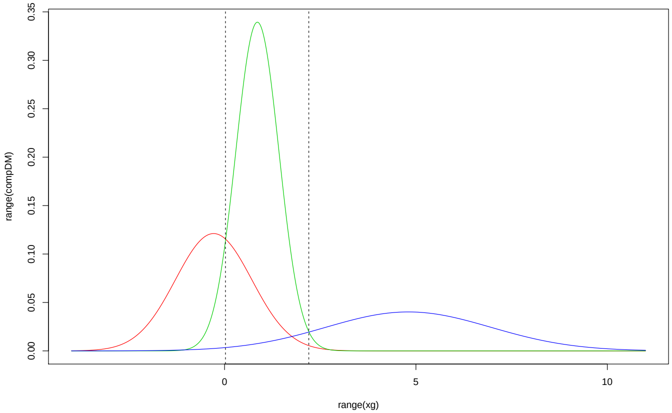
Let $I = j$ if y comes from component j, then

$$p(I = j \mid y, \theta, p) \propto p(I = j) p(y \mid I = j, \theta, p) = p_j f(y \mid \theta_j)$$

by Bayes Theorem.

For each y_i we can assign it to the most probable component.

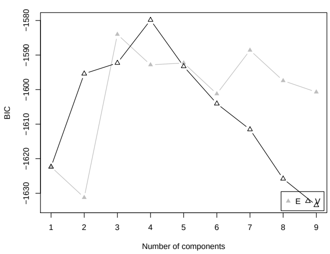
If we classify an observation to the most probable component then we pick the component such that $p_j f(y \mid \theta_j)$ is highest.

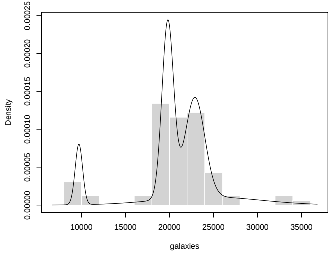


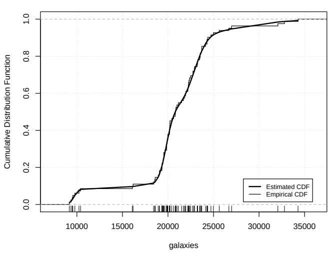
Galaxies Data

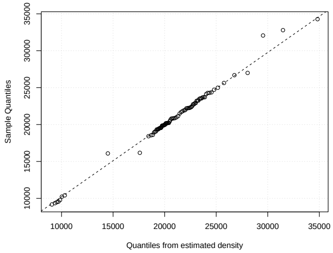
```
Description:

     A numeric vector of velocities in km/sec of 82 galaxies from 6
     well-separated conic sections of an unfilled survey of the
     Corona Borealis region.  Multimodality in such surveys is evidence
     for voids and superclusters in the far universe.
```

```
-------------------------------------------------------
Density estimation via Gaussian finite mixture modeling
-------------------------------------------------------

Mclust V (univariate, unequal variance) model with 4 components:

 log-likelihood  n df       BIC       ICL
       -765.694 82 11 -1579.862 -1598.907
```


BIC
Number of components
E
V
−1580
−1590
−1600
−1610
−1620
−1630
1 2 3 4 5 6 7 8 9


Density
galaxies
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025
10000 15000 20000 25000 30000 35000

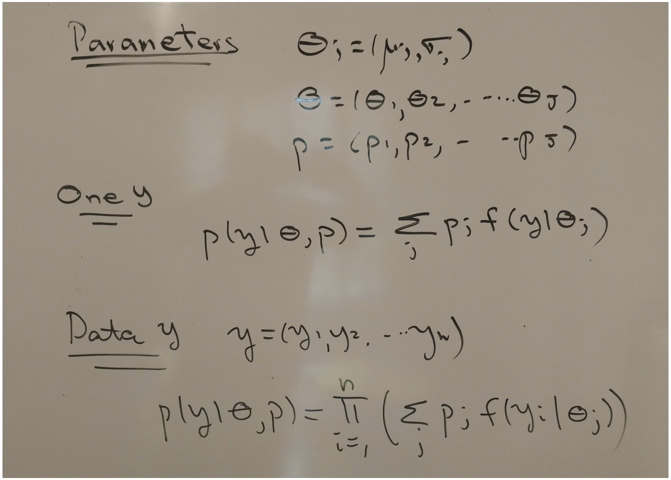

Cumulative Distribution Function
galaxies
Estimated CDF
Empirical CDF
0.0
0.2
0.4
0.6
0.8
1.0
10000 15000 20000 25000 30000 35000


Sample Quantiles
Quantiles from estimated density
10000
15000
20000
25000
30000
35000
10000 15000 20000 25000 30000 35000

Amazing!!

4. The EM Algorithm for Univariate Mixtures of Normals

Here is our model:



Parameters $\theta_j = (\mu_j, \sigma_j)$

$\theta = (\theta_1, \theta_2, \ldots \theta_J)$

$p = (p_1, p_2, \ldots p_J)$

One y

$$p(y \mid \theta, p) = \sum_j p_j f(y \mid \theta_j)$$

Data y $y = (y_1, y_2, \ldots y_n)$

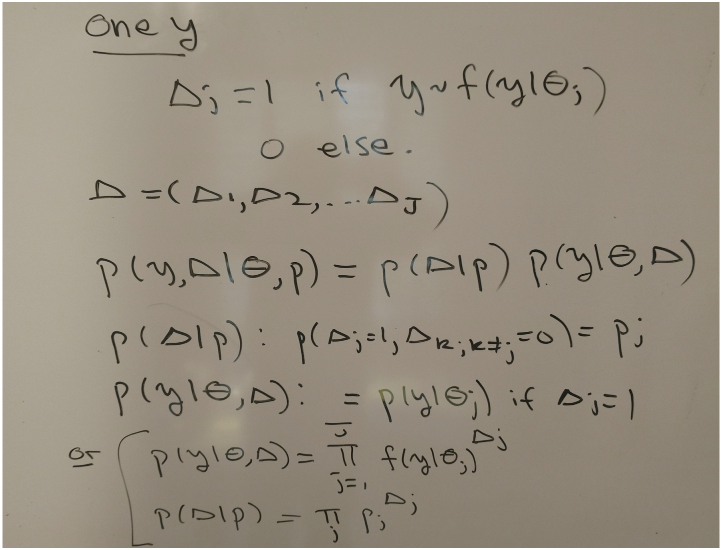
$$p(y \mid \theta, p) = \prod_{i=1}^{n} \left(\sum_j p_j f(y_i \mid \theta_j) \right)$$

Usually we log the likelihood and then the product turns into a sum.

In this case the terms we are summing are the log of the sums over the mixture components and this is not friendly to optimize.

We introduce a latent variable indicating which mixture component a y is from.

For one y:



One y

$\Delta_j = 1$ if $y \sim f(y|\theta_j)$

0 else.

$\Delta = (\Delta_1, \Delta_2, \ldots \Delta_J)$

$p(y, \Delta | \theta, p) = p(\Delta | p)\, p(y | \theta, \Delta)$

$p(\Delta|p)$: $p(\Delta_j = 1, \Delta_{k; k \neq j} = 0) = p_j$

$p(y|\theta,\Delta)$: $= p(y|\theta_j)$ if $\Delta_j = 1$

or

$p(y|\theta,\Delta) = \prod_{j=1}^{J} f(y|\theta_j)^{\Delta_j}$

$p(\Delta|p) = \prod_j p_j^{\Delta_j}$

Then $p(y, \Delta \mid \theta, p)$ has the mixture model $p(y \mid \theta, p)$ as it's marginal.

For a sample $y = (y_1, y_2, \ldots, y_i, \ldots, y_n)$, each y_i gets it's own $(\Delta_{i1}, \Delta_{i2}, \ldots, \Delta_{ij}, \ldots, \Delta_{iJ})$. so the full model is now

$$\begin{aligned} & p(y, \Delta \mid \theta, p) \\ &= p(y \mid \Delta, \theta)\, p(\Delta \mid p) \\ &= \left[\prod_i \prod_j f(y_i \mid \theta_j)^{\Delta_{ij}} \right] \left[\prod_i \prod_j p_j^{\Delta_{ij}} \right] \end{aligned}$$

and now, taking the log will help!!

But we have a lot of Δ_{ij} to deal with!!

Here is the EM idea.

It is an interative scheme. At each iteration we have current estimates of (θ, p).

(1) **E step**.

Given the current values of (θ', p') compute the expected value of

$$\log(p(y, \Delta \mid \theta, p))$$

where the expectation is over $\Delta \mid y, \theta', p'$.

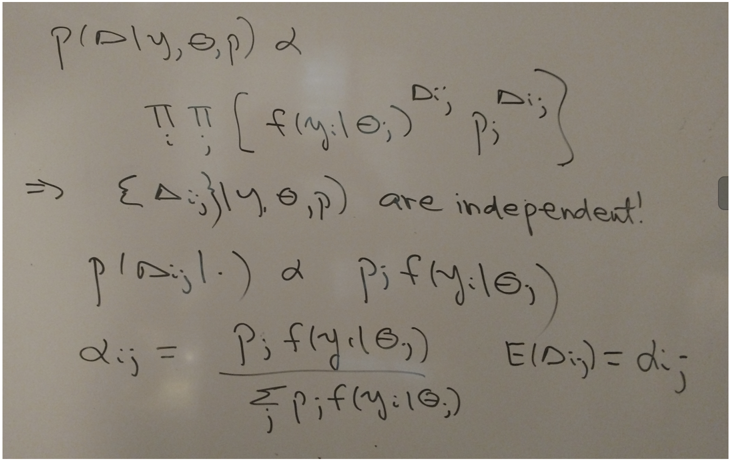
(2) **M step**.

Get new values of (θ, p) by optimizing the expected log likelihood computed in the E step.

As usual, iterate until convergence.

E Step :

The log likelihood is linear in the Δ_{ij} we we just need the expectations.



$$p(\Delta | y, \theta, p) \propto$$

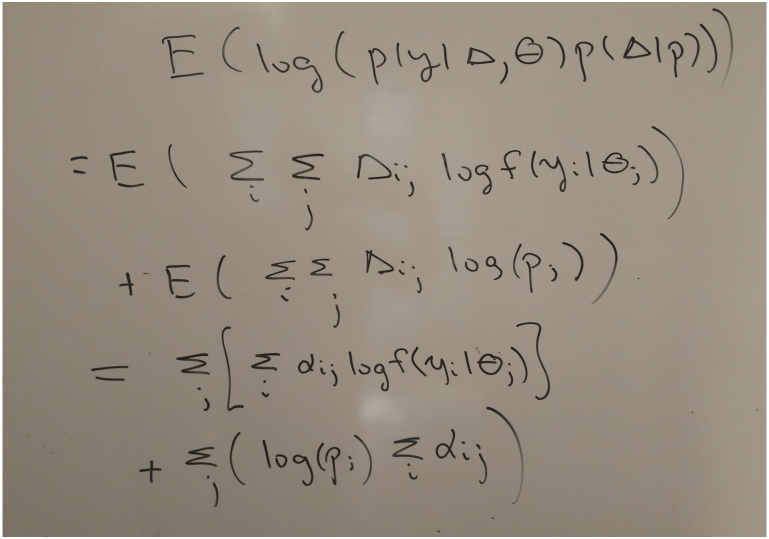
$$\prod_i \prod_j \left[f(y_i | \theta_j)^{\Delta_{ij}} \, p_j^{\Delta_{ij}} \right]$$

$$\Rightarrow \quad \{\Delta_{ij}\} | y, \theta, p) \text{ are independent!}$$

$$p(\Delta_{ij} | \cdot) \propto p_j f(y_i | \theta_j)$$

$$\alpha_{ij} = \frac{p_j f(y_i | \theta_j)}{\sum_j p_j f(y_i | \theta_j)} \qquad E(\Delta_{ij}) = \alpha_{ij}$$

The $\Delta_{i,j}$ are independent over i (observations) not over j (components) obviously.



$$E\left(\log\left(p(y \mid \Delta, \Theta)\, p(\Delta \mid p)\right)\right)$$

$$= E\left(\sum_i \sum_j \Delta_{ij} \log f(y_i \mid \theta_j) \right)$$

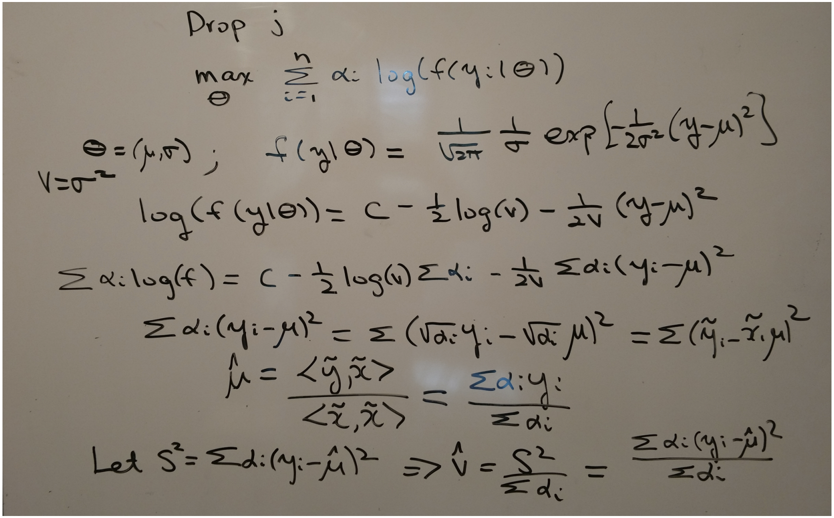
$$+ E\left(\sum_i \sum_j \Delta_{ij} \log(p_j) \right)$$

$$= \sum_j \left[\sum_i \alpha_{ij} \log f(y_i \mid \theta_j) \right]$$

$$+ \sum_j \left(\log(p_j) \sum_i \alpha_{ij} \right)$$

so, in the M step, we can optimize over each θ_j and p separately!!!!

M step for the θ_j:



Drop j

$$\max_{\theta} \sum_{i=1}^{n} \alpha_i \log(f(y_i|\theta))$$

$$\theta=(\mu,\sigma) \; ; \quad f(y|\theta) = \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} \exp\left[-\frac{1}{2\sigma^2}(y-\mu)^2\right\}$$

$$V=\sigma^2$$

$$\log(f(y|\theta)) = C - \frac{1}{2}\log(v) - \frac{1}{2v}(y-\mu)^2$$

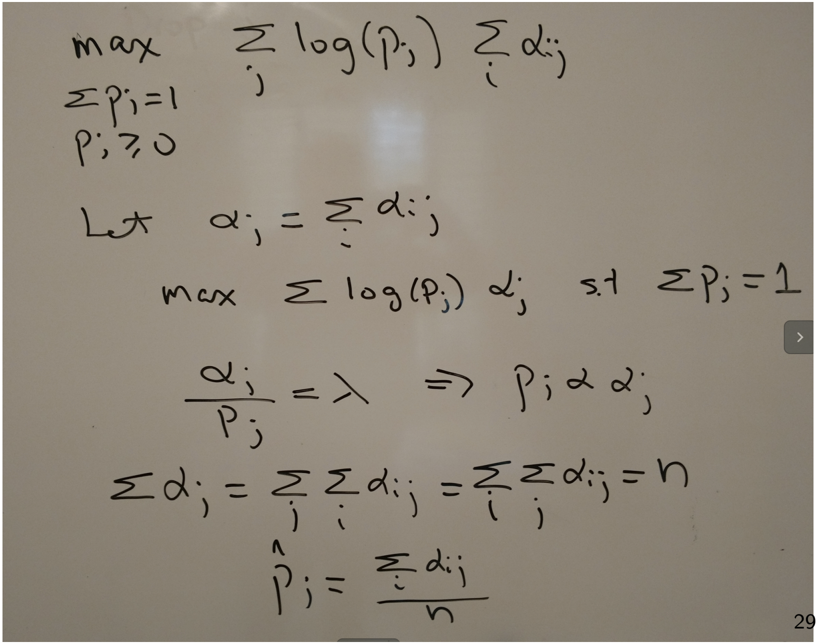
$$\sum \alpha_i \log(f) = C - \frac{1}{2}\log(v)\sum \alpha_i - \frac{1}{2v}\sum \alpha_i (y_i-\mu)^2$$

$$\sum \alpha_i (y_i-\mu)^2 = \sum(\sqrt{\alpha_i}\, y_i - \sqrt{\alpha_i}\, \mu)^2 = \sum(\tilde{y}_i - \tilde{x}_i \mu)^2$$

$$\hat{\mu} = \frac{\langle \tilde{y},\tilde{x}\rangle}{\langle \tilde{x},\tilde{x}\rangle} = \frac{\sum \alpha_i y_i}{\sum \alpha_i}$$

$$\text{Let } S^2 = \sum \alpha_i (y_i-\hat{\mu})^2 \Rightarrow \hat{v} = \frac{S^2}{\sum \alpha_i} = \frac{\sum \alpha_i (y_i-\hat{\mu})^2}{\sum \alpha_i}$$

M step for the p, λ is the Lagrange multiplier:



$$\max \sum_j \log(p_j) \sum_i \alpha_{ij}$$

$\sum p_j = 1$

$p_j \geq 0$

Let $\alpha_j = \sum_i \alpha_{ij}$

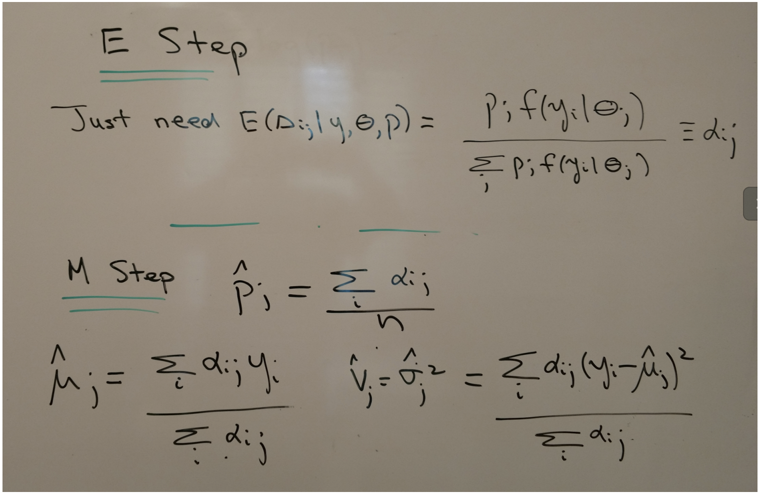
$$\max \sum \log(p_j)\, \alpha_j \quad \text{s.t} \quad \sum p_j = 1$$

$$\frac{\alpha_j}{p_j} = \lambda \Rightarrow p_j \propto \alpha_j$$

$$\sum \alpha_j = \sum_j \sum_i \alpha_{ij} = \sum_i \sum_j \alpha_{ij} = n$$

$$\hat{p}_j = \frac{\sum_i \alpha_{ij}}{n}$$

EM Algorithm, Mixture of Univariate Normals:



E Step

Just need $E(\Delta_{ij} | y, \theta, p) = \frac{p_j f(y_i | \theta_j)}{\sum_j p_j f(y_i | \theta_j)} \equiv \alpha_{ij}$

M Step $\hat{p}_j = \frac{\sum_i \alpha_{ij}}{n}$

$\hat{\mu}_j = \frac{\sum_i \alpha_{ij} y_i}{\sum_i \alpha_{ij}}$ $\hat{V}_j = \hat{\sigma}_j^2 = \frac{\sum_i \alpha_{ij} (y_i - \hat{\mu}_j)^2}{\sum_i \alpha_{ij}}$

See Algorithm 8.5, page 275, "The Elements of Statistical Learning".

Starting Values:

For the case $J = 2$, “The Elements” (page 274) says:

> *A good way to construct initial guesses for $\hat{\mu}_1$ and $\hat{\mu}_2$ is simply to choose two of the y_i at random. Both $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ can be set equall to the overall sample variance $(\sum_{i=1}^{n}(y_i - \bar{y})^2)/n$. The mixing proportion $\hat{\pi}$ can be started at the value .5.*

Notes:

- After the dust settles, it is a very simple algorithm.
- What happens when all the α_{ij} are close to 0 or 1, what does this mean?
- The α_{ij} are called the "responsibilities", they give a "soft assignment" of observation i to component j.
- Can converge to local minimum so starting values matter and you may want to try multiple runs to find a useful minimum.

Label Switching:

Note that the model is fundamentally unidentified in that the labels for the components does not matter.

For example if I just switch p_1 and p_2 and θ_1 and θ_2 then I have the exact same model for the data.

In the simple univariate case normal mixture model you can identify the labels by imposing constraints such as

$$\hat{\mu}_j < \hat{\mu}_{j+1}$$

Note:

We started with the mixture model:

$$p(y \mid \theta, p) = \sum_{j=1}^{J} p_j \, f(y \mid \theta_j)$$

We then added the latent variables Δ_{ij}. We can think of the latents two different ways:

- A computation device to get the mle of (θ, p).
- *Maybe we really want to think of our data as coming from different sources !!!!!*. The Δ_{ij} really reflect how we think about the model, about how the model "relates to the real world".

The second case is the really powerful idea underlying the use of latent variables in many complex models.

Maybe there are a set of different kinds of galaxies out there!!
Maybe there is one kind of intrinic intelligence and different tests just reflect that one underlying attribute in different ways!!

5. The EM Algorithm

Start with a model

$$p(y \mid \theta)$$

Elaborate the model to include latent variables:

$$p(y, z \mid \theta)$$

is such a way that the marginal model (margin out z) is our orginal model.

Note: in our mixture mode "θ" $= (\theta, p)$ and $Z = \Delta$.

Let θ' be a current value.

Iterate as follows:

E Step:

$$Q(\theta, \theta') = E(log(p(y, z \mid \theta))$$

where the expectation is taken over

$$Z \mid y, \theta'$$

M Step:

Get the next θ by maximizing $Q(\theta, \theta')$.

6. Multivariate Mixtures of Normals

The mixture of normals model gets more exciting when we use the multivariate normal distribution.

y is now a vector and $\theta_j = (\mu_j, \Sigma_j)$ where now μ is a vector and Σ is a variance matrix.

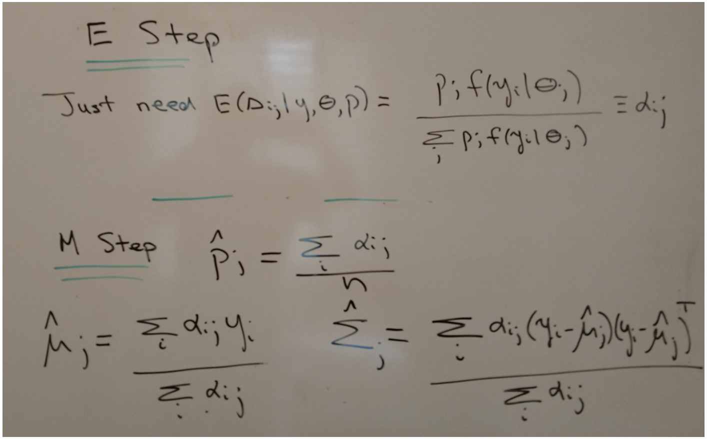
$$f(y \mid \theta_j) \sim N(\mu_j, \Sigma_j)$$

and

$$p(y \mid \theta, p) = \sum_{j=1}^{J} p_j \, f(y \mid \theta_j)$$

as in the univariate case.

EM algorithm for mixture of multivariate normals.



E Step

Just need $E(\Delta_{ij} | y, \theta, p) = \dfrac{p_j f(y_i | \theta_j)}{\sum_j p_j f(y_i | \theta_j)} \equiv \alpha_{ij}$

M Step $\hat{p}_j = \dfrac{\sum_i \alpha_{ij}}{n}$

$$\hat{\mu}_j = \frac{\sum_i \alpha_{ij} y_i}{\sum_i \alpha_{ij}} \qquad \hat{\Sigma}_j = \frac{\sum_i \alpha_{ij} (y_i - \hat{\mu}_j)(y_i - \hat{\mu}_j)^T}{\sum_i \alpha_{ij}}$$

See for example section 11.4.2 of "Machine Learning, a Probabilistic Approach" by Kevin Murphy.

Simplifying Σ_j

In the univariate case, the `mclust` R-package considered two models

- unequal variances: $\theta_j = (\mu_j, \sigma_j)$.
- equal variances: $\theta_j = (\mu_j, \sigma)$.

And then BIC was used to choose both the number of components and the model.

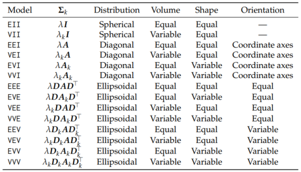
In the multivariate case, `mclust` considers a large number of simplifying assumptions about the Σ_j expressed in terms of the decomposition

$$\Sigma_j = \lambda_j \, D_k A_k D_k'$$

where λ_j is a scalar, D_k is an orthogonal matrix, and A_k is diagonal.

mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models

by Luca Scrucca, Michael Fop, T. Brendan Murphy and Adrian E. Raftery



Model	$\boldsymbol{\Sigma}_k$	Distribution	Volume	Shape	Orientation
EII	$\lambda \boldsymbol{I}$	Spherical	Equal	Equal	—
VII	$\lambda_k \boldsymbol{I}$	Spherical	Variable	Equal	—
EEI	$\lambda \boldsymbol{A}$	Diagonal	Equal	Equal	Coordinate axes
VEI	$\lambda_k \boldsymbol{A}$	Diagonal	Variable	Equal	Coordinate axes
EVI	$\lambda \boldsymbol{A}_k$	Diagonal	Equal	Variable	Coordinate axes
VVI	$\lambda_k \boldsymbol{A}_k$	Diagonal	Variable	Variable	Coordinate axes
EEE	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{\top}$	Ellipsoidal	Equal	Equal	Equal
EVE	$\lambda \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^{\top}$	Ellipsoidal	Equal	Variable	Equal
VEE	$\lambda_k \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{\top}$	Ellipsoidal	Variable	Equal	Equal
VVE	$\lambda_k \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^{\top}$	Ellipsoidal	Variable	Variable	Equal
EEV	$\lambda \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^{\top}$	Ellipsoidal	Equal	Equal	Variable
VEV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^{\top}$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^{\top}$	Ellipsoidal	Equal	Variable	Variable
VVV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^{\top}$	Ellipsoidal	Variable	Variable	Variable

Table 3: Parameterisations of the within-group covariance matrix $\boldsymbol{\Sigma}_k$ for multidimensional data available in the **mclust** package, and the corresponding geometric characteristics.

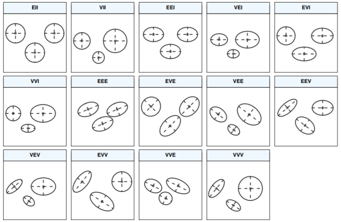
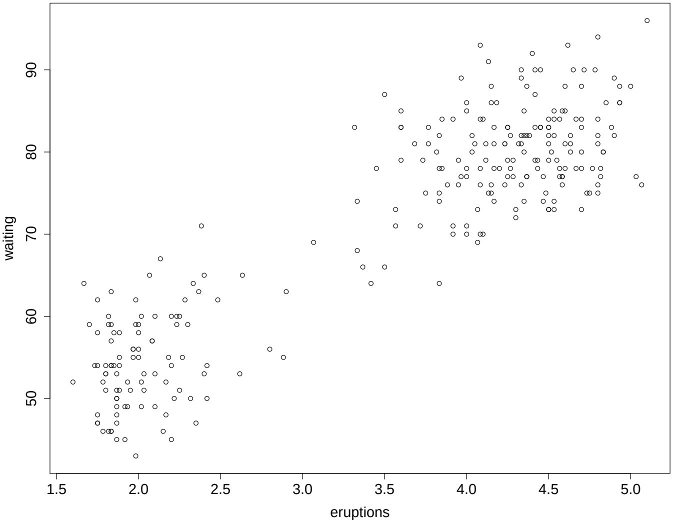


Figure 2: Ellipses of isodensity for each of the 14 Gaussian models obtained by eigen-decomposition in case of three groups in two dimensions.

Eruptions of old faithful, Bivariate Normal Mixtures

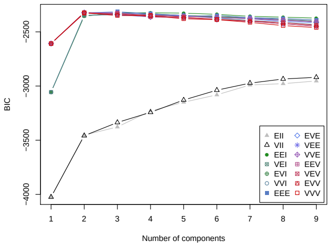
```
##faithful data
#A data frame with 272 observations on 2 variables.
#
#       [,1]  eruptions  numeric  Eruption time in mins
#       [,2]  waiting    numeric  Waiting time to next
#                                 eruption (in mins)

> head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
```

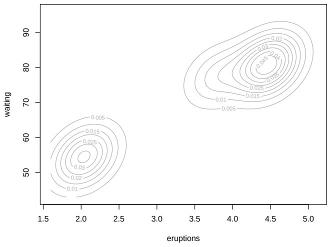


Obviously not bivariate normal.

BIC selects model EEE with just three components !!!!

```
-------------------------------------------------------
Density estimation via Gaussian finite mixture modeling
-------------------------------------------------------

Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3
components:

 log-likelihood   n df       BIC       ICL
      -1126.326 272 11 -2314.316 -2357.824
```


BIC
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
EVE
VEE
VVE
EEV
VEV
EVV
VVV

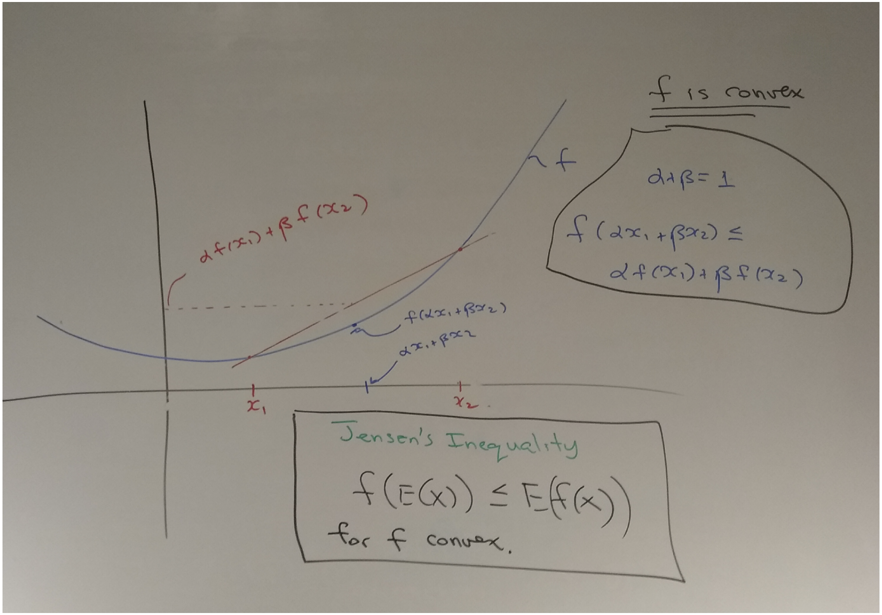

waiting
eruptions

7. More on EM

We can actually get a handle on how different maximizing the expected log likelihood is from maximizing the likelihood.

And we get to use the Kullback-Leibler divergence!!

First, recall Jensen's inequality.



f is convex

$\alpha + \beta = 1$

$f(\alpha x_1 + \beta x_2) \leq \alpha f(x_1) + \beta f(x_2)$

Jensen's Inequality

$f(E(x)) \leq E(f(x))$

for f convex.

Kullback-Leibler Divergence

f, g densities.

Want a "distance" between f and g.

$$K(f,g) = \int \log(f/g)\, f$$

Note

(i) $K(f,g) = \int [-\log](g/f)\, f$

$\geq (-\log)\left(\int \frac{g}{f} f\right)$ [$-\log$ is convex]

$= (-\log)(1) = 0$

(ii) $K(f,f) = 0$

KL example, the exponential distribution

$f(x) = e^{-x} \qquad E[x] = 1$

$Y = \frac{X}{\lambda} \qquad X = \lambda Y \qquad \frac{dx}{dy} = \lambda$

$f_Y(y) = \lambda e^{-\lambda y} \qquad E\{Y\} = \frac{1}{\lambda}$

$$\log \frac{f(y|\lambda_1)}{f(y|\lambda_2)} = \{\log \lambda_1 - \lambda_1 Y] - [\log \lambda_2 - \lambda_2 Y]$$

$$= [\log \lambda_1 - \log \lambda_2\} + Y[\lambda_2 - \lambda_1\}$$

$$K(f(y|\lambda_1), f(y|\lambda_2)) = \log\left(\frac{\lambda_1}{\lambda_2}\right) + \frac{\lambda_2 - \lambda_1}{\lambda_1}$$

EM:

Model: $f(z, x \mid \theta)$

- x observed
- z latent

Iterates of θ: $\{\theta^t\}$.

$$Q(\theta \mid \theta^t) = E(log(f(z, x \mid \theta))$$

where E is over $Z \mid x, \theta^t$.

$$\theta^{t+1} = \underset{\theta}{\text{argmax}}\ Q(\theta \mid \theta^t)$$

$$f(z,x|\theta) = f(x|\theta)\, f(z|x,\theta)$$

$$\log f(x|\theta) = E \log f(z,x|\theta) - E \log f(z|x,\theta)$$

E is wrt $z|x,\theta^t$

$$\ell(\theta) = Q(\theta|\theta^t) - E \log f(z|x,\theta)$$

Let $k(\theta|\theta^t) \equiv k\left(f(z|x,\theta^t), f(z|x,\theta)\right)$

$$= E\, f(z|x,\theta^t) - E\, f(z|x,\theta))$$

$$\ell(\theta) = Q(\theta|\theta^t) + k(\theta|\theta^t) - C \quad \left[\begin{array}{l} C = \\ E f(z|x,\theta^t) \end{array} \right.$$

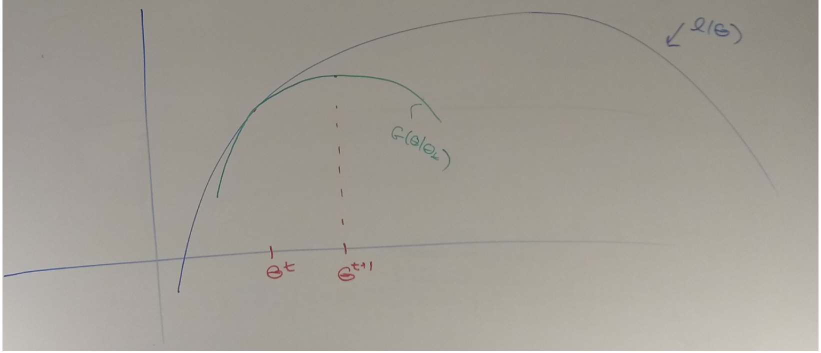
$$G(\theta|\theta^t) \equiv Q(\theta|\theta^t) - C$$

$$= \ell(\theta) - k(\theta|\theta^t)$$

$$\theta^{t+1} = \arg\max_{\theta} G(\theta|\theta^t)$$

$$k(\theta^t|\theta^t) = 0 \qquad k(\theta|\theta^t) \geq 0$$

$$G(\theta|\theta^t) = \ell(\theta) - k(\theta|\theta^t)$$



So, for example, we know that if

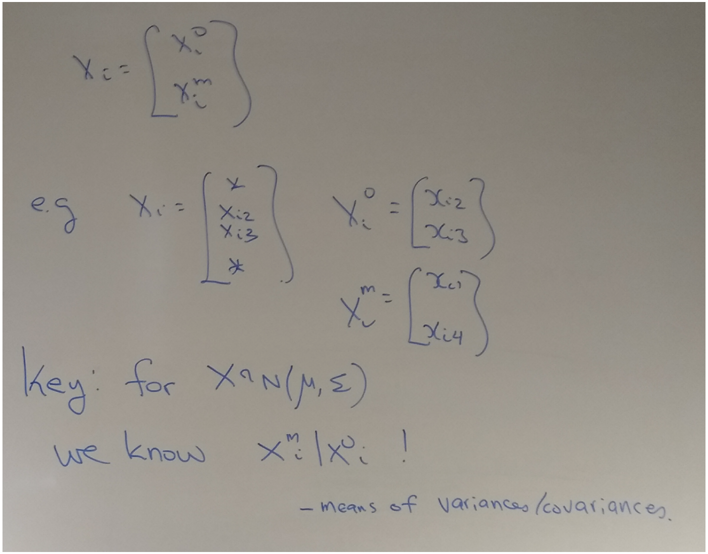
θ^{t+1} is different from θ^{t},

we actually did increase the likelihood.

8. Missing Data with the IID Multivariate Normal

Suppose we have our IID $X_i \sim N_p(\mu, \Sigma)$ model and we want MLEs for μ and Σ.

But, in some of the X_i some of the components of X_i are missing.



$$X_i = \begin{bmatrix} X_i^o \\ X_i^m \end{bmatrix}$$

e.g $X_i = \begin{bmatrix} * \\ X_{i2} \\ X_{i3} \\ * \end{bmatrix}$ $X_i^o = \begin{bmatrix} x_{i2} \\ x_{i3} \end{bmatrix}$

$X_i^m = \begin{bmatrix} x_{i1} \\ x_{i4} \end{bmatrix}$

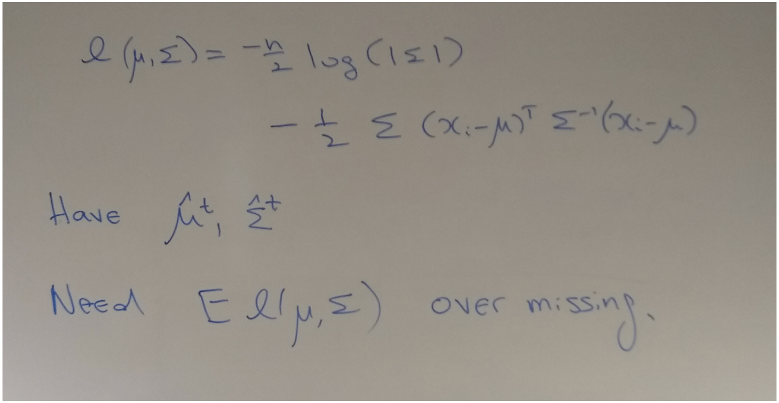
Key: for $X \sim N(\mu, \Sigma)$

we know $X_i^m \mid X_i^o$!

– means of variances/covariances.

We assume we have: MAR, missing at random.

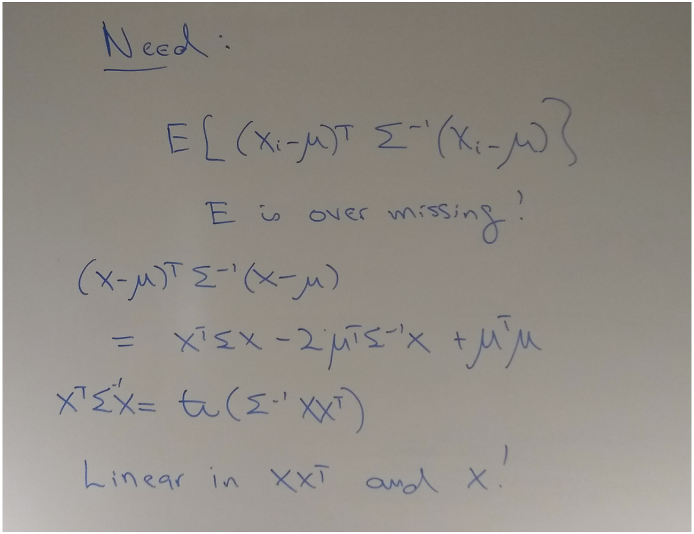
For example, we don't tend to drop the biggest or smallest, it is just random which is missing.



$$\ell(\mu, \Sigma) = -\frac{n}{2} \log(|\Sigma|) - \frac{1}{2} \sum (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)$$

Have $\hat{\mu}^t, \hat{\Sigma}^t$

Need $E\,\ell(\mu, \Sigma)$ over missing.



Need:

$$E\{(x_i-\mu)^T \Sigma^{-1} (x_i-\mu)\}$$

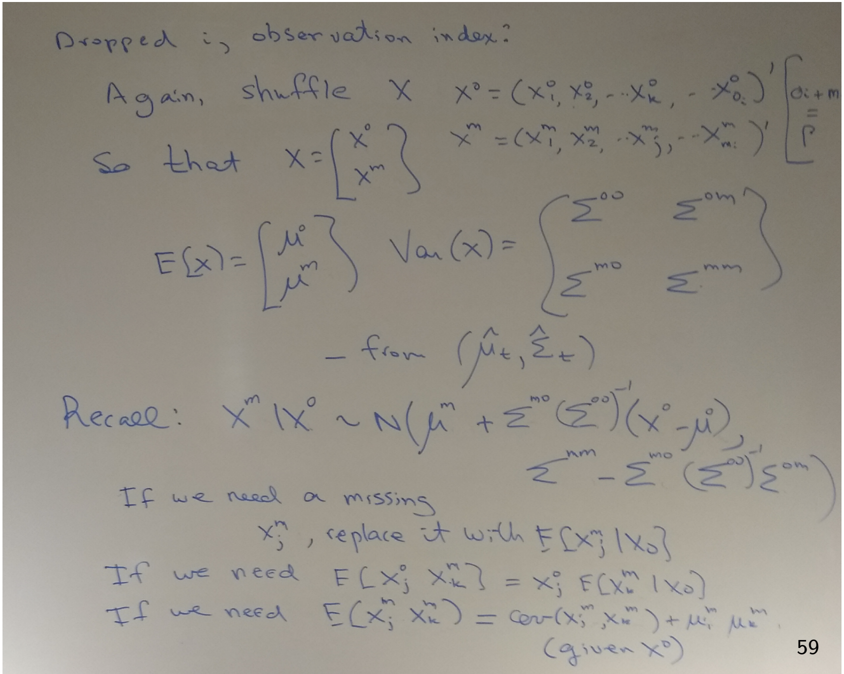
E is over missing!

$$(x-\mu)^T \Sigma^{-1} (x-\mu)$$

$$= x^T \Sigma x - 2\mu^T \Sigma^{-1} x + \mu^T \mu$$

$$x^T \Sigma^{-1} x = tr(\Sigma^{-1} x x^T)$$

Linear in xx^T and x!



Dropped i, observation index:

Again, shuffle X $\quad x^o = (x_1^o, x_2^o, \dots x_k^o, \dots x_{o_i}^o)' \quad x^m = (x_1^m, x_2^m, \dots x_j^m, \dots x_{m_i}^m)' \quad \left[\begin{matrix} o_i + m_i \\ = \\ p \end{matrix}\right.$

So that $X = \begin{bmatrix} x^o \\ x^m \end{bmatrix}$

$$E\{x\} = \begin{bmatrix} \mu^o \\ \mu^m \end{bmatrix} \quad Var(x) = \begin{pmatrix} \Sigma^{oo} & \Sigma^{om} \\ \Sigma^{mo} & \Sigma^{mm} \end{pmatrix}$$

— from $(\hat{\mu}_t, \hat{\Sigma}_t)$

Recall: $X^m \mid X^o \sim N\left(\mu^m + \Sigma^{mo}(\Sigma^{oo})^{-1}(x^o - \mu^o),\ \Sigma^{mm} - \Sigma^{mo}(\Sigma^{oo})^{-1}\Sigma^{om}\right)$

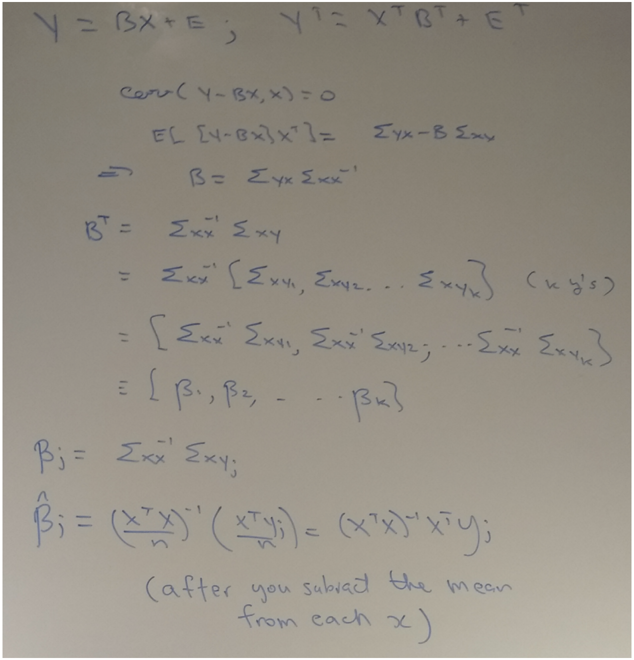
If we need a missing x_j^m, replace it with $E\{x_j^m \mid x_o\}$

If we need $E\{x_j^o x_k^m\} = x_j^o E(x_k^m \mid x_o)$

If we need $E(x_j^m x_k^m) = cov(x_j^m, x_k^m) + \mu_j^m \mu_k^m$
(given x^o)

Note that this is very close to regression imputation where we impute missing values by regressing the missing on the non-missing.

Note the our formula for the conditional mean of a multivariate normal subvector Y says you should run a regression of each element of the subvector on X.



$Y = BX + E$; $Y^T = X^T B^T + E^T$

$\operatorname{cov}(Y - BX, X) = 0$

$E[\{Y - BX\} X^T] = \Sigma_{yx} - B\,\Sigma_{xx}$

$\Rightarrow \quad B = \Sigma_{yx} \Sigma_{xx}^{-1}$

$B^T = \Sigma_{xx}^{-1} \Sigma_{xy}$

$= \Sigma_{xx}^{-1} \{\Sigma_{xy_1}, \Sigma_{xy_2} \ldots \Sigma_{xy_k}\}$ (k y's)

$= \{\Sigma_{xx}^{-1} \Sigma_{xy_1}, \Sigma_{xx}^{-1} \Sigma_{xy_2}, \ldots \Sigma_{xx}^{-1} \Sigma_{xy_k}\}$

$= \{\beta_1, \beta_2, \ldots \beta_k\}$

$\beta_j = \Sigma_{xx}^{-1} \Sigma_{xy_j}$

$\hat{\beta}_j = \left(\frac{X^T X}{n}\right)^{-1} \left(\frac{X^T y_j}{n}\right) = (X^T X)^{-1} X^T y_j$

(after you subtract the mean from each x)