Machine Learning for Signal Processing
Linear Gaussian Models

Class 14. 3 Nov 2015

Instructor: Bhiksha Raj
Recap: MAP Estimators

- MAP (Maximum A Posteriori): Find most probable value of $y$ given $x$

$$y = \arg\max_y P(Y|x)$$
MAP estimation

- $x$ and $y$ are jointly Gaussian

$$z = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$E[z] = \mu_z = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$$

$$\text{Var}(z) = C_{zz} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix}$$

$$C_{xy} = E[(x - \mu_x)(y - \mu_y)^T]$$

$$P(z) = N(\mu_z, C_{zz}) = \frac{1}{\sqrt{2\pi |C_{zz}|}} \exp\left(-0.5(z - \mu_z)^T C_{zz}^{-1} (z - \mu_z) \right)$$

- $z$ is Gaussian
MAP estimation: Gaussian PDF
MAP estimation: The Gaussian at a particular value of X
Conditional Probability of $y \mid x$

$$P(y \mid x) = N(\mu_y + C_{yx} C_{xx}^{-1} (x - \mu_x), C_{yy} - C_{yx} C_{xx}^{-1} C_{xy})$$

$$E_{y\mid x}[y] = \mu_{y\mid x} = \mu_y + C_{yx} C_{xx}^{-1} (x - \mu_x)$$

$$Var(y \mid x) = C_{yy} - C_{yx} C_{xx}^{-1} C_{xy}$$

- The conditional probability of $y$ given $x$ is also Gaussian
  - The slice in the figure is Gaussian
- The mean of this Gaussian is a function of $x$
- The variance of $y$ reduces if $x$ is known
  - Uncertainty is reduced
MAP estimation: The Gaussian at a particular value of $X$
MAP Estimation of a Gaussian RV

\[ \hat{y} = \arg \max_y P(y \mid x) = E_{y \mid x}[y] \]
Its also a minimum-mean-squared error estimate

• Minimize error:

\[ \text{Err} = E[\|y - \hat{y}\|^2 | x] = E[(y - \hat{y})^T (y - \hat{y}) | x] \]

\[ \text{Err} = E[y^T y + \hat{y}^T \hat{y} - 2\hat{y}^T y | x] = E[y^T y | x] + \hat{y}^T \hat{y} - 2\hat{y}^T E[y | x] \]

• Differentiating and equating to 0:

\[ d.\text{Err} = 2\hat{y}^T d\hat{y} - 2E[y | x]^T d\hat{y} = 0 \]

\[ \hat{y} = E[y | x] \]

The MMSE estimate is the mean of the distribution
For the Gaussian: MAP = MMSE

Most likely value is also the MEAN value.

- Would be true of any symmetric distribution
A Likelihood Perspective

- \( y \) is a noisy reading of \( a^T x \)
  \[
y = a^T x + e
\]
- Error \( e \) is Gaussian
  \[
e \sim N(0, \sigma^2 I)
\]
- Estimate \( A \) from
  \[
  Y = [y_1 \ y_2 \ldots y_N] \quad X = [x_1 \ x_2 \ldots x_N]
  \]
The *Likelihood* of the data

\[ y = a^T x + e \quad \text{e} \sim N(0, \sigma^2 I) \]

• Probability of observing a specific \( y \), given \( x \), for a particular matrix \( a \)

\[ P(y \mid x; a) = N(y; a^T x, \sigma^2 I) \]

• Probability of collection: \( \mathbf{X} = [\mathbf{x}^1 \mathbf{x}^3 \ldots \mathbf{x}^N] \quad \mathbf{Y} = [y^1 y^3 \ldots y^N] \)

\[ P(Y \mid X; a) = \prod_i N(y_i; a^T x_i, \sigma^2 I) \]

• Assuming IID for convenience (not necessary)
A Maximum Likelihood Estimate

\[ y = a^T x + e \quad e \sim N(0, \sigma^2 I) \quad Y = [y_1 \ y_2 \ldots y_N] \quad X = [x_1 \ x_2 \ldots x_N] \]

\[ P(Y | X) = \prod_i \frac{1}{\sqrt{(2\pi\sigma^2)^D}} \exp\left( \frac{-1}{2\sigma^2} \| y_i - a^T x_i \|^2 \right) \]

\[ \log P(Y | X; a) = C - \sum_i \frac{1}{2\sigma^2} \| y_i - a^T x_i \|^2 \]

\[ \log P(Y | X, a) = C - \frac{1}{2\sigma^2} \text{trace} \left( (Y - a^T X)^T (Y - a^T X) \right) \]

- Maximizing the log probability is identical to minimizing the least squared error
A problem with regressions

• ML fit is sensitive
  – Error is squared
  – Small variations in data $\rightarrow$ large variations in weights
  – Outliers affect it adversely

• Unstable
  – If dimension of $\mathbf{X} \geq$ no. of instances
    • $(\mathbf{XX}^T)$ is not invertible

\[
A = (\mathbf{XX}^T)^{-1} \mathbf{XY}^T
\]
MAP estimation of weights

- Assume weights drawn from a Gaussian
  \[ P(a) = N(0, \sigma^2 I) \]
- Max. Likelihood estimate
  \[ \hat{a} = \arg \max_a \log P(Y \mid X; a) \]
- Maximum \textit{a posteriori} estimate
  \[ \hat{a} = \arg \max_a \log P(a \mid Y, X) = \arg \max_a \log P(Y \mid X, a) P(a) \]
MAP estimation of weights

\[ \hat{a} = \arg \max_A \log P(a | Y, X) = \arg \max_A \log P(Y | X, a) P(a) \]

- \( P(a) = N(0, \sigma^2 I) \)
- \( \log P(a) = C - \log \sigma - 0.5\sigma^{-2} \|a\|^2 \)

\[ \log P(Y | X, a) = C - \frac{1}{2\sigma^2} \text{trace}\left((Y - a^T X)^T (Y - a^T X)\right) \]

\[ \hat{a} = \arg \max_A C' - \log \sigma - \frac{1}{2\sigma^2} \text{trace}\left((Y - a^T X)^T (Y - a^T X)\right) - 0.5\sigma^2 a^T a \]

- Similar to ML estimate with an additional term
MAP estimate of weights

\[ dL = \left( 2a^T XX^T + 2yX^T + 2\sigma I \right) da = 0 \]

\[ a = (XX^T + \sigma I)^{-1} XY^T \]

• Equivalent to *diagonal loading* of correlation matrix
  – Improves condition number of correlation matrix
    • Can be inverted with greater stability
  – Will not affect the estimation from well-conditioned data
  – Also called Tikhonov Regularization
    • Dual form: Ridge regression

• **MAP estimate of weights**
  – Not to be confused with MAP estimate of Y
MAP estimate priors

- Left: Gaussian Prior on W
- Right: Laplacian Prior

\[
\frac{1}{2b} \exp \left( -\frac{|x - \mu|}{b} \right)
\]
MAP estimation of weights with laplacian prior

- Assume weights drawn from a Laplacian
  \[ P(a) = \lambda^{-1} \exp(-\lambda^{-1}|a|_1) \]
- Maximum a posteriori estimate

\[
\hat{a} = \arg\max_a C' - \text{trace}\left( (Y - a^T X)^T (Y - a^T X) \right) - \lambda^{-1}|a|_1
\]

- No closed form solution
  - Quadratic programming solution required
    - Non-trivial
MAP estimation of weights with laplacian prior

• Assume weights drawn from a Laplacian
  \[ P(a) = \lambda^{-1} \exp(-\lambda^{-1}|a|_1) \]

• Maximum a posteriori estimate
  \[
  \hat{a} = \arg \max_a \text{trace}\left( (Y - a^T X)^T (Y - a^T X) \right) - \lambda^{-1}|a|_1
  \]

• Identical to L_1 regularized least-squares estimation
**L₁-regularized LSE**

\[ \hat{a} = \arg \max_a C' - \text{trace} \left( (Y - a^T X)^T (Y - a^T X)^T \right) - \lambda^{-1} |a|_1 \]

- No closed form solution
  - Quadratic programming solutions required

- Dual formulation

\[ \hat{a} = \arg \max_a C' - \text{trace} \left( (Y - a^T X)^T (Y - a^T X)^T \right) \quad \text{subject to} \quad |a|_1 \leq t \]

- “LASSO” – Least absolute shrinkage and selection operator
LASSO Algorithms

- Various convex optimization algorithms
- LARS: Least angle regression
- Pathwise coordinate descent
- Matlab code available from web
Regularized least squares

- Regularization results in selection of suboptimal (in least-squares sense) solution
  - One of the loci outside center
- Tikhonov regularization selects *shortest* solution
- $L_1$ regularization selects *sparsest* solution

Image Credit: Tibshirani
The different formalisms in $L_2$

\[ \hat{a} = \arg \max_a C' - \text{trace} \left( (Y - a^T X)^T (Y - a^T X) \right) - \lambda^{-1} \|a\|^2 \]

\[ \hat{a} = \arg \max_a C' - \text{trace} \left( (Y - a^T X)^T (Y - a^T X) \right) \quad \text{subject to} \quad \|a\|^2 \leq t \]

- Expand both the ball and the ellipses till the both just meet
- Fix the ball, expand the ellipse till it meets the ball
The different formalisms in $L_1$

\[ \hat{a} = \arg \max_a C' - \text{trace}\left( (Y - a^T X)^T (Y - a^T X)^T \right) - \lambda^{-1} |a|_1 \]

\[ \hat{a} = \arg \max_a C' - \text{trace}\left( (Y - a^T X)^T (Y - a^T X)^T \right) \text{ subject to } |a|_1 \leq t \]

- Expand both the diamond and the ellipses till the both just meet
- Fix the diamond, expand the ellipse till it meets the ball
• General statistical estimators
• All used to predict a variable, based on other parameters related to it.

• Most common assumption: Data are Gaussian, all RVs are Gaussian
  – Other probability densities may also be used.

• For Gaussians relationships are linear as we saw.
Gaussians and more Gaussians..

• Linear Gaussian Models..

• But first a recap
A Brief Recap

- Principal component analysis: Find the $K$ bases that best explain the given data
- Find $B$ and $C$ such that the difference between $D$ and $BC$ is minimum
  - While constraining that the columns of $B$ are orthonormal
Remember Eigenfaces

• Approximate every face $f$ as

$$f = w_{f,1} V_1 + w_{f,2} V_2 + w_{f,3} V_3 + \ldots + w_{f,k} V_k$$

• Estimate $V$ to minimize the squared error

• **Error is unexplained by $V_1 \ldots V_k$**

• **Error is orthogonal to Eigenfaces**
**Eigen Representation**

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

Illustration assuming 3D space

\[ w_{11} = w_{11} + \varepsilon_1 \]
Representation

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

\[ w_{12} + \varepsilon_2 \]

Illustration assuming 3D space

Error is at 90° to the eigenface
 Representation

- K-dimensional representation
  - Error is orthogonal to representation

All data with the same representation $wV_1$ lie a plane orthogonal to $wV_1$
With 2 bases

\[ \text{Error is at } 90^\circ \text{ to the eigenfaces} \]

\[ w_{11} + w_{21} + \varepsilon_1 \]

Illustration assuming 3D space

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance
• K-dimensional representation
  – Error is orthogonal to representation
  – Weight and error are specific to data instance

\[ w_{12} + w_{22} + \epsilon_2 \]
In Vector Form

\[ X_i = w_{1i} V_1 + w_{2i} V_2 + \varepsilon_i \]

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

Error is at 90° to the eigenfaces
In Vector Form

\[ X_i = w_{1i} V_1 + w_{2i} V_2 + \varepsilon_i \]

\[ x = Vw + e \]

- \( K \)-dimensional representation
- \( x \) is a \( D \) dimensional vector
- \( V \) is a \( D \times K \) matrix
- \( w \) is a \( K \) dimensional vector
- \( e \) is a \( D \) dimensional vector

Error is at 90° to the eigenface
• For the given data: find the K-dimensional subspace such that it captures most of the variance in the data
  – Variance in remaining subspace is minimal
Constraints

\[ x = Vw + e \]

- \( V^TV = I \) : Eigen vectors are orthogonal to each other
- For every vector, error is orthogonal to Eigen vectors
  - \( e^TV = 0 \)
- Over the collection of data
  - Average \( w^Tw = \text{Diagonal} \) : Eigen representations are uncorrelated
  - \( e^Te = \) minimum: Error variance is minimum
    - Mean of error is 0

Error is at 90° to the eigenface
A Statistical Formulation of PCA

\[ x = Vw + e \]

\[ w \sim N(0, B) \]

\[ e \sim N(0, E) \]

- \( x \) is a random variable generated according to a linear relation
- \( w \) is drawn from an \( K \)-dimensional Gaussian with diagonal covariance
- \( e \) is drawn from a 0-mean (\( D-K \))-rank \( D \)-dimensional Gaussian
- Estimate \( V \) (and \( B \)) given examples of \( x \)
Linear Gaussian Models!!

$$x = Vw + e$$

$$w \sim N(0, B)$$

$$e \sim N(0, E)$$

- $x$ is a random variable generated according to a linear relation
- $w$ is drawn from a Gaussian
- $e$ is drawn from a 0-mean Gaussian
- Estimate $V$ given examples of $x$
  - In the process also estimate $B$ and $E$
Linear Gaussian Models!!

- $x$ is a random variable generated according to a linear relation
- $w$ is drawn from a Gaussian
- $e$ is drawn from a $0$-mean Gaussian

Estimate $V$ given examples of $x$
- In the process also estimate $B$ and $E$

PCA is a specific instance of a linear Gaussian model with particular constraints
- $B = \text{Diagonal}$
- $\mathbf{v}^T \mathbf{v} = 1$
- $E$ is low rank
Linear Gaussian Models

\[ x = \mu + Vw + e \quad w \sim N(0, B) \]
\[ e \sim N(0, E) \]

• Observations are linear functions of two uncorrelated Gaussian random variables
  – A “weight” variable \( w \)
  – An “error” variable \( e \)
  – Error not correlated to weight: \( E[e^Tw] = 0 \)

• Learning LGMs: Estimate parameters of the model given instances of \( x \)
  – The problem of learning the distribution of a Gaussian RV
LGMs: Probability Density

\[ \mathbf{x} = \mu + \mathbf{Vw} + \mathbf{e} \]
\[ \mathbf{w} \sim N(0, B) \]
\[ \mathbf{e} \sim N(0, E) \]

• The mean of \( \mathbf{x} \):

\[ \mathbb{E}[\mathbf{x}] = \mu + \mathbf{V}E[\mathbf{w}] + \mathbb{E}[\mathbf{e}] = \mu \]

• The Covariance of \( \mathbf{x} \):

\[ \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^T] = \mathbf{V}B\mathbf{V}^T + \mathbf{E} \]
The probability of $\mathbf{x}$

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{Vw} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, \mathbf{B})$$

$$\mathbf{e} \sim N(0, \mathbf{E})$$

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{VBV}^T + \mathbf{E})$$

$$P(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^D | \mathbf{VBV}^T + \mathbf{E}|}} \exp\left(-0.5(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{VBV}^T + \mathbf{E})^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

- $\mathbf{x}$ is a linear function of Gaussians: $\mathbf{x}$ is also Gaussian
- Its mean and variance are as given
Estimating the variables of the model

\[ x = \mu + Vw + e \]

\[ w \sim N(0, B) \]
\[ e \sim N(0, E) \]

\[ x \sim N(\mu, VBV^T + E) \]

- Estimating the variables of the LGM is equivalent to estimating \( P(x) \)
  - The variables are \( \mu, V, B \) and \( E \)
Estimating the model

\[ x = \mu + Vw + e \]

\[ w \sim N(0, B) \]
\[ e \sim N(0, E) \]

\[ x \sim N(\mu, VBV^T + E) \]

• The model is indeterminate:
  – \( Vw = VCC^{-1}w = (VC)(C^{-1}w) \)
  – We need extra constraints to make the solution unique

• Usual constraint: \( B = I \)
  – Variance of \( w \) is an identity matrix
Estimating the variables of the model

\[ x = \mu + Vw + e \]

\[ w \sim N(0, I) \]
\[ e \sim N(0, E) \]

\[ x \sim N(\mu, VV^T + E) \]

- Estimating the variables of the LGM is equivalent to estimating \( P(x) \)
  - The variables are \( \mu, V, \) and \( E \)
The Maximum Likelihood Estimate

\[ \mathbf{x} \sim \mathcal{N}(\mathbf{\mu}, \mathbf{VV}^T + \mathbf{E}) \]

- Given training set \( \mathbf{x}_1, \mathbf{x}_2, .. \mathbf{x}_N \), find \( \mathbf{\mu}, \mathbf{V}, \mathbf{E} \)

- The ML estimate of \( \mathbf{\mu} \) does not depend on the covariance of the Gaussian

\[ \mathbf{\mu} = \frac{1}{N} \sum_i \mathbf{x}_i \]
We can safely assume “centered” data
- $\mu = 0$

If the data are not centered, “center” it
- Estimate mean of data
  - Which is the maximum likelihood estimate
- Subtract it from the data
Simplified Model

\[ x = Vw + e \]

\[ w \sim N(0, I) \]
\[ e \sim N(0, E) \]

\[ x \sim N(0, VV^T + E) \]

- Estimating the variables of the LGM is equivalent to estimating \( P(x) \)
  - The variables are \( V \), and \( E \)
Estimating the model

\[ x = Vw + e \]

\[ x \sim N(0, VV^T + E) \]

- Given a collection of \( x_i \) terms
  - \( x_1, x_2, \ldots, x_N \)
- Estimate \( V \) and \( E \)
- \( w \) is unknown for each \( x \)
- But if assume we know \( w \) for each \( x \), then what do we get:
Estimating the Parameters

\[ x_i = Vw_i + e \]

\[ P(e) = N(0, E) \]

\[ P(x \mid w) = N(Vw, E) \]
Reminder: $x$ and $w$ are jointly Gaussian

$x = Vw + e$

\[ P(x) = N(0, VV^T + E) \]

\[ P(w) = N(0, I) \]

\[ C_{xz} = E[(x - \mu_x)(w - \mu_w)^T] = V \]

\[
C_{zz} = \begin{bmatrix} C_{xx} & C_{xw} \\ C_{wx} & C_{ww} \end{bmatrix} = \begin{bmatrix} VV^T + E & V \\ V^T & I \end{bmatrix}
\]

\[ \mu_z = \begin{bmatrix} \mu_x \\ \mu_w \end{bmatrix} = 0 \]

$\mathbf{z} = \begin{bmatrix} x \\ w \end{bmatrix}$

• $x$ and $w$ are jointly Gaussian!
MAP estimation: Gaussian PDF
MAP estimation: The Gaussian at a particular value of $X$
Conditional Probability of $x \mid w$

$$P(x \mid w) = N(\mu_x + C_{xw} C_{ww}^{-1} (w - \mu_w), C_{xx} - C_{xw} C_{ww}^{-1} C_{wx})$$

$$= N(C_{xw} C_{ww}^{-1} w, C_{xx} - C_{xw} C_{ww}^{-1} C_{wx})$$

$$E_{x \mid w}[x] = C_{xw} C_{ww}^{-1} w$$

$$Var(x \mid w) = C_{xx} - C_{xw} C_{ww}^{-1} C_{wx}$$

• Comparing to

$$P(x \mid w) = N(Vw, E)$$

• We get:

$$V = C_{xw} C_{ww}^{-1}$$

$$E = C_{xx} - C_{xw} C_{ww}^{-1} C_{wx}$$
Or more explicitly

\[ C_{ww} = \frac{1}{N} \sum_{i} w_{i} w_{i}^{T} \]

\[ C_{xw} = \frac{1}{N} \sum_{i} x_{i} w_{i}^{T} \]

\[ V = C_{xw} C_{ww}^{-1} \]

\[ E = C_{xx} - C_{xw} C_{ww}^{-1} C_{wx} \]

\[ V = \left( \sum_{i} x_{i} w_{i}^{T} \right) \left( \sum_{i} w_{i} w_{i}^{T} \right)^{-1} \]

\[ E = \frac{1}{N} \left( \sum_{i} x_{i} x_{i}^{T} - V \sum_{i} w_{i} x_{i}^{T} \right) \]
Estimating LGMs: If we know $w$

$$x_i = Vw_i + e$$

$$P(e) = N(0, E)$$

$$V = \left( \sum_i x_i w_i^T \right) \left( \sum_i w_i w_i^T \right)^{-1}$$

$$E = \frac{1}{N} \left( \sum_i x_i x_i^T - V \sum_i w_i x_i^T \right)$$

- But in reality we don’t know the $w$ for each $x$
  - So how to deal with this?

- EM..
Recall EM

• We figured out how to compute parameters if we knew the missing information
• Then we “fragmented” the observations according to the posterior probability $P(z|x)$ and counted as usual
• In effect we took the expectation with respect to the a posteriori probability of the missing data: $P(z|x)$
EM for LGMs

\[ x_i = Vw_i + e \]

\[ P(e) = N(0, E) \]

\[ V = \left( \sum_i x_i w_i^T \right) \left( \sum_i w_i w_i^T \right)^{-1} \]

\[ E = \frac{1}{N} \left( \sum_i x_i x_i^T - V \sum_i w_i w_i^T \right) \]

\[ V = \left( \sum_i x_i E_{w|x_i} [w^T] \right) \left( \sum_i E_{w|x_i} [ww^T] \right)^{-1} \]

\[ E = \frac{1}{N} \sum_i x_i x_i^T - \frac{1}{N} V \sum_i E_{w|x_i} [w] x_i^T \]

• Replace unseen data terms with expectations taken w.r.t. \( P(w|x_i) \)
EM for LGMs

\[
x_i = Vw_i + e
\]

\[
P(e) = N(0, E)
\]

\[
V = \left( \sum_i x_iw_i^T \right) \left( \sum_i w_iw_i^T \right)^{-1}
\]

\[
E = \frac{1}{N} \left( \sum_i x_ix_i^T - V \sum_i w_ix_i^T \right)
\]

- Replace unseen data terms with expectations taken w.r.t. \( P(w|x_i) \)
Flipping the problem

- How do we estimate the above terms?
- MAP to the rescue!!

\[ E_{w|x_i}[w] \]
\[ E_{w|x_i}[ww^T] \]
Expected Value of \( w \) given \( x \)

\[ x = Vw + e \]

\[ P(e) = N(0, E) \quad P(w) = N(0, I) \]

\[ P(x) = N(0, VV^T + E) \]

• \( x \) and \( w \) are jointly Gaussian!
  – \( x \) is Gaussian
  – \( w \) is Gaussian
  – They are linearly related

\[ z = \begin{bmatrix} x \\ w \end{bmatrix} \quad P(z) = N(\mu_z, C_{zz}) \]
Recall: \( w \) and \( x \) are jointly Gaussian

\[
x = Vw + e
\]

\[
e \sim N(0, E) \quad P(w) = N(0, I)
\]

\[
P(x) = N(0, VV^T + E)
\]

\[
C_{xx} = VV^T + E \quad C_{ww} = I
\]

\[
C_{xw} = E[(x - \mu_x)(w - \mu_w)^T] = V
\]

\[
P(z) = N(\mu_z, C_{zz})
\]

\[
\mu_z = \begin{bmatrix} \mu_x \\ \mu_w \end{bmatrix} = 0
\]

\[
C_{zz} = \begin{bmatrix} C_{xx} & C_{xw} \\ C_{wx} & C_{ww} \end{bmatrix}
\]

\* \( x \) and \( w \) are jointly Gaussian! \*
$P(w \mid z)$

- $P(w \mid z)$ is a Gaussian

\[
P(w \mid x) = N(\mu_w + C_{wx} C_{xx}^{-1} (x - \mu_x), C_{ww} - C_{wx} C_{xx}^{-1} C_{xw})
\]

\[
= N(C_{wx} C_{xx}^{-1} x, C_{ww} - C_{wx} C_{xx}^{-1} C_{xw})
\]

\[
= N(V^T (VV^T + E)^{-1} x, I - V^T (VV^T + E)^{-1} V)
\]

$\text{Var}(w \mid x) = I - V^T (VV^T + E)^{-1} V$

$E_{w \mid x_i}[w] = V^T (VV^T + E)^{-1} x_i$

$E_{w \mid x_i}[ww^T] = \text{Var}(w \mid x) + E_{w \mid x_i}[w] E_{w \mid x_i}[w]^T$

$E_{w \mid x_i}[ww^T] = I - V^T (VV^T + E)^{-1} V + E_{w \mid x_i}[w] E_{w \mid x_i}[w]^T$
LGM: The complete EM algorithm

\[ x = Vw + e \]

\[ e \sim N(0, E) \quad P(w) = N(0, I) \]

\[ P(x) = N(0, VV^T + E) \]

- Initialize \( V \) and \( E \)

- E step:
  \[ E_{w|x_i}[w] = V^T (VV^T + E)^{-1} x_i \]

  \[ E_{w|x_i}[ww^T] = I - V^T (VV^T + E)^{-1} V + E_{w|x_i}[w]E_{w|x_i}[w]^T \]

- M step:
  \[ V = \left( \sum_i x_i E_{w|x_i}[w^T] \right) \left( \sum_i E_{w|x_i}[ww^T] \right)^{-1} \]

  \[ E = \frac{1}{N} \sum_i x_i x_i^T - \frac{1}{N} V \sum_i E_{w|x_i}[w]x_i^T \]
So what have we achieved

• Employed a complicated EM algorithm to learn a \textit{Gaussian} PDF for a variable $x$

• What have we gained???

• Next class:
  – PCA
    • Sensible PCA
    • EM algorithms for PCA
  – Factor Analysis
    • FA for feature extraction
LGMs : Application 1

Learning principal components

\[ x = Vw + e \]
\[ w \sim N(0, I) \]
\[ e \sim N(0, E) \]

• Find directions that capture most of the variation in the data
• Error is orthogonal to these variations
The full covariance matrix of a Gaussian has $D^2$ terms.

- Fully captures the relationships between variables.
- Problem: **Needs a lot of data to estimate robustly**
To be continued..

• Other applications..
• Next class