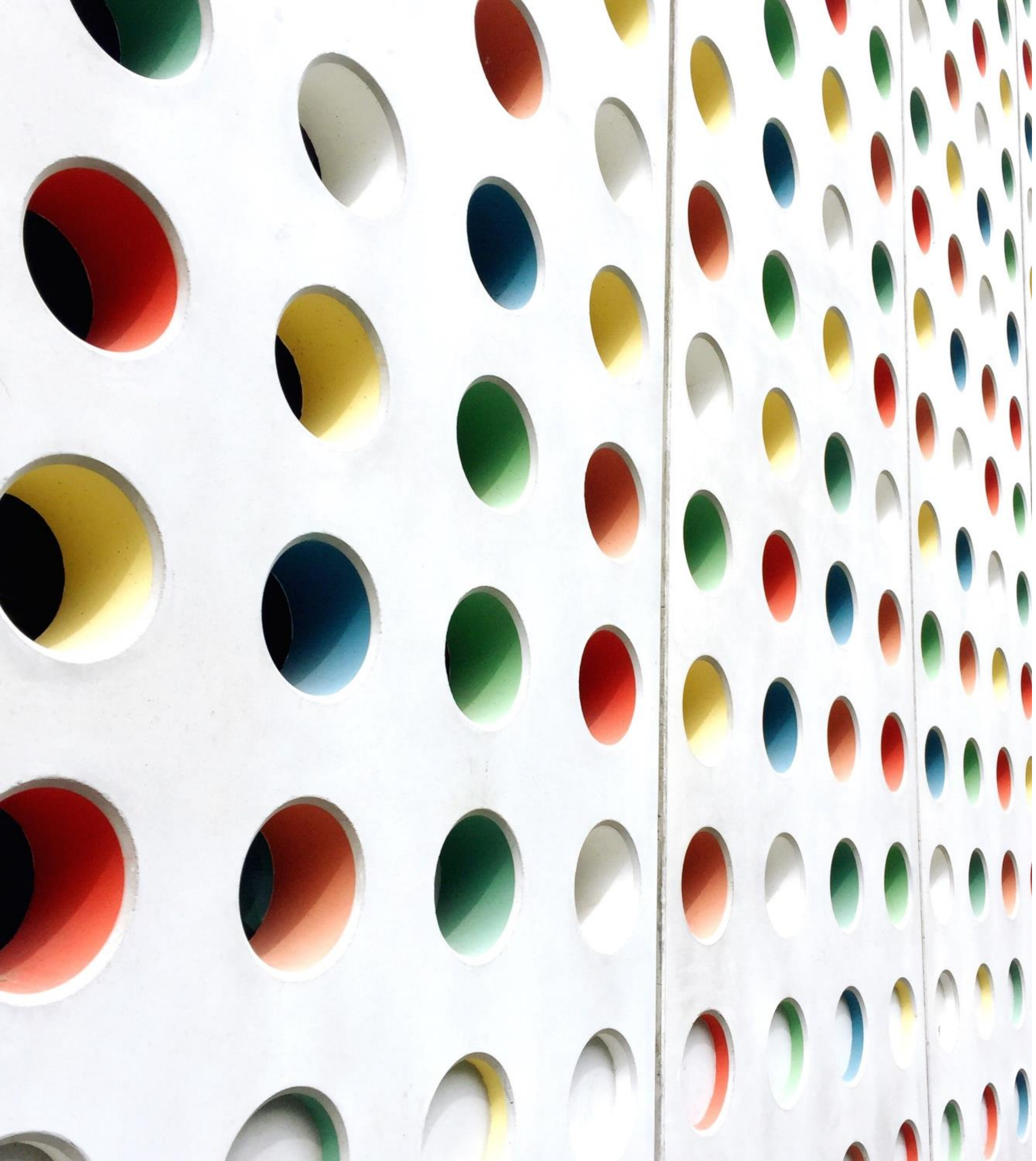


Гаусови модели

Basics

- One-dimensional Gaussian pdf

$$\mathcal{N}(x|\mu,\sigma^2) \quad \triangleq \quad \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

- Multivariate Gaussian pdf

$$\mathcal{N} = (\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right]$$

Eigenvalues and Eigenvectors*

- Eigenvectors and eigenvalues

$$Av = v\lambda$$

  - $\mathbf{A}$ – a $\mathbf{N}$x$\mathbf{N}$ matrix
  - v – a $\mathbf{N}$x$\mathbf{1}$ vector (eigenvector)
  - λ – a scalar (eigenvalue)

- Eigendecomposition of matrix

$$AV = V\Lambda$$

$$A = V\Lambda V^{-1} = V\Lambda V^{T}$$

Mahalanobis distance

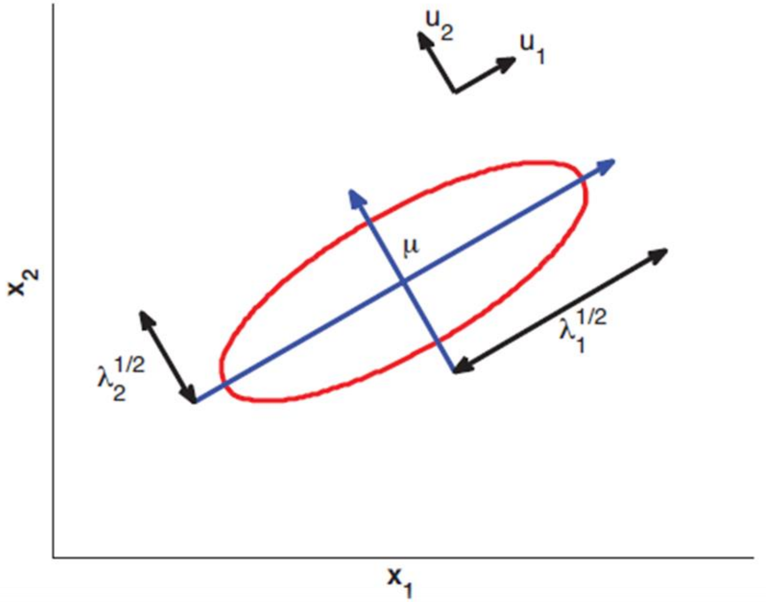
- Euclidean distance
- Mahalanobis distance $(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$
  - Eigendecomposition of Σ as $\Sigma = \mathrm{U}\Lambda\mathrm{U}^T$, where U is an orthonormal matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues

$$\boldsymbol{\Sigma}^{-1} = \mathbf{U}^{-T}\boldsymbol{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U}^T = \sum_{i=1}^{D} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T$$

$$\begin{aligned}
(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) &= (\mathbf{x}-\boldsymbol{\mu})^T \left(\sum_{i=1}^{D} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T \right) (\mathbf{x}-\boldsymbol{\mu}) \\
&= \sum_{i=1}^{D} \frac{1}{\lambda_i}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{u}_i\mathbf{u}_i^T (\mathbf{x}-\boldsymbol{\mu}) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i} \quad \text{where } y_i \triangleq \mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu})
\end{aligned}$$

Gaussian as an ellipsoid

Formula for ellipse in 2D is $\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} = 1$



The contours of the Gaussian lie along ellipses, where the **eigenvectors determine its orientation**, and the **eigenvalues determine its elongation**

In general, the Mahalanobis distance corresponds to Euclidean distance in a transformed coordinate system, where we shift by μ and rotate by $\mathbf{U}$

It is a multi-dimensional generalization of the idea of measuring how many standard deviations σ away x is from the mean μ

MLE for a MVN

- If we have ***N iid*** samples $x_i \sim N(\mu, \sigma^2)$, then the MLE for the parameters is given by *the empirical mean* and *empirical covariance*

$$\hat{\boldsymbol{\mu}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \triangleq \overline{\mathbf{x}}$$

$$\hat{\boldsymbol{\Sigma}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T = \frac{1}{N}(\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T) - \overline{\mathbf{x}}\,\overline{\mathbf{x}}^T$$

- In the univariate case, we get the following results:

$$\hat{\mu} = \frac{1}{N}\sum_i x_i = \overline{x}$$

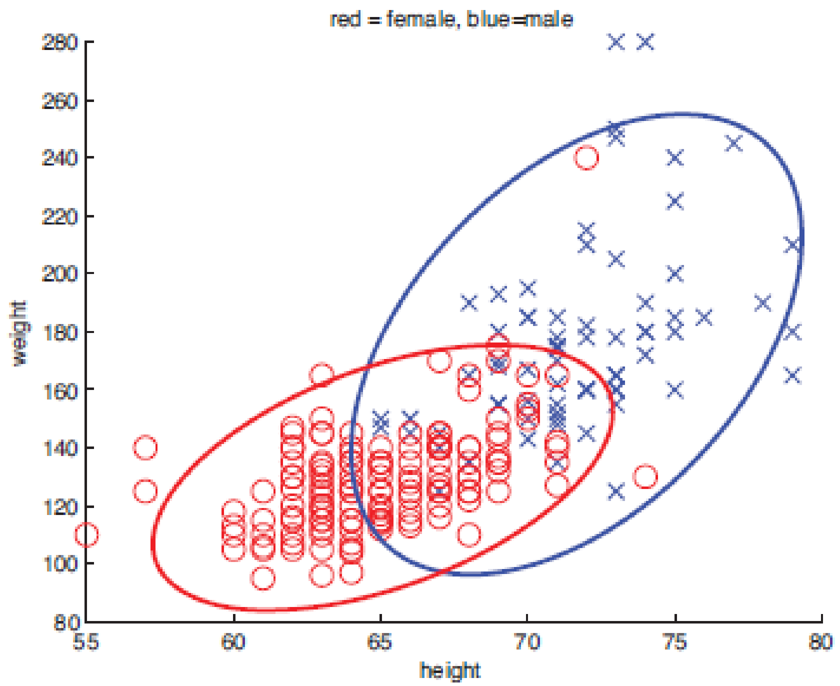
$$\hat{\sigma}^2 = \frac{1}{N}\sum_i (x_i - \overline{x})^2 = \left(\frac{1}{N}\sum_i x_i^2\right) - (\overline{x})^2$$

Gaussian discriminant analysis

- One application of MVNs is to define the class conditional densi in a generative classifier

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) \propto p(\mathbf{x}|y = c, \boldsymbol{\theta})p(y = c|\boldsymbol{\theta})$$

where $p(\mathbf{x}|y = c, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$



Two Gaussian class-conditional densities of the height and weight of men and women. The ellipses contain 95% of the probability mass.

- We can classify a feature vector using the following decision rule:

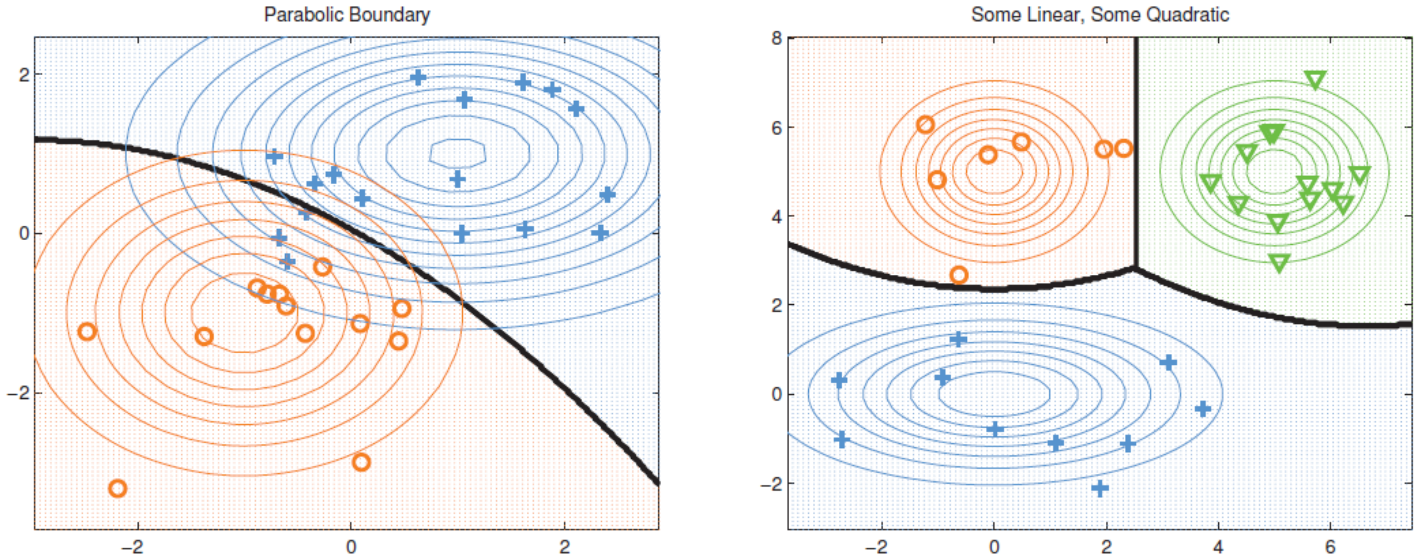
$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \left[\log p(y = c|\boldsymbol{\pi}) + \log p(\mathbf{x}|\boldsymbol{\theta}_c)\right]$$

- When we compute the probability of $\mathbf{x}$ under each class conditional density, we are measuring the distance from $\mathbf{x}$ to the center of each class, μ_c, using Mahalanobis distance (**nearest centroids classifier**)

Quadratic discriminant analysis (QDA)

If we plug the Gaussian density in the standard Bayes rule for classification we get

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_c |2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right]}{\sum_{c'} \pi_{c'} |2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{c'})^T\boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{c'})\right]}$$



Linear discriminant analysis (LDA)

- We consider a special case where the covariance matrices are **tied/shared** across classes, $\Sigma_c = \Sigma$

$$\begin{aligned} p(y=c|\mathbf{x},\boldsymbol{\theta}) &\propto \pi_c \exp\left[\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right] \\ &= \exp\left[\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}] \end{aligned}$$

- Since the quadratic term $x^T\Sigma^{-1}x$ is independent of c, it will cancel out in the numerator and denominator.

Linear discriminant analysis (LDA)

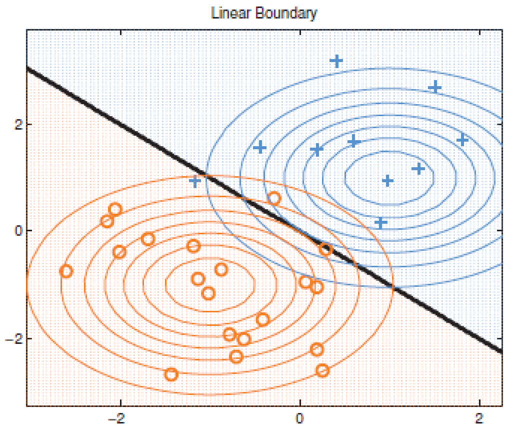
- If we introduce a change of variables and plug them in the previous equation:

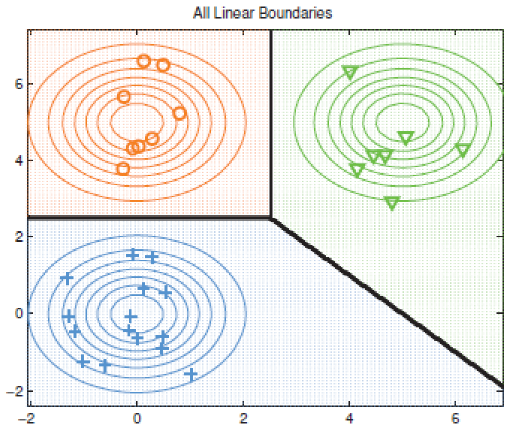
$$\gamma_c = -\frac{1}{2}\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c + \log \pi_c$$
$$\boldsymbol{\beta}_c = \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c$$

we get $p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c$

and that is the **softmax** function $\mathcal{S}(\boldsymbol{\eta})_c = \frac{e^{\eta_c}}{\sum_{c'=1}^{C} e^{\eta_{c'}}}$

which for two classes becomes a sigmoid function





- If we take logs of the softmax function, we end up with a linear function of x. Thus, the decision boundary between any two classes will be a straight line. Hence, this technique is called **linear discriminant analysis** (LDA)

Two class LDA

- In the binary case, the posterior is
$$p(y=1|\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1}}{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1}+e^{\boldsymbol{\beta}_0^T\mathbf{x}+\gamma_0}} = \frac{1}{1+e^{(\boldsymbol{\beta}_0-\boldsymbol{\beta}_1)^T\mathbf{x}+(\gamma_0-\gamma_1)}} = \mathrm{sigm}\left((\boldsymbol{\beta}_1-\boldsymbol{\beta}_0)^T\mathbf{x}+(\gamma_1-\gamma_0)\right)$$

- Now
$$\begin{aligned}\gamma_1-\gamma_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1+\frac{1}{2}\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0+\log(\pi_1/\pi_0)\\ &= -\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)+\log(\pi_1/\pi_0)\end{aligned}$$

if we define
$$\begin{aligned}\mathbf{w} &= \boldsymbol{\beta}_1-\boldsymbol{\beta}_0=\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\\ \mathbf{x}_0 &= \frac{1}{2}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_0)-(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\frac{\log(\pi_1/\pi_0)}{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)}\end{aligned}$$

then we have $\mathbf{w}^T\mathbf{x}_0=-(\gamma_1-\gamma_0)$, and hence $p(y=1|\mathbf{x},\boldsymbol{\theta}) = \mathrm{sigm}(\mathbf{w}^T(\mathbf{x}-\mathbf{x}_0))$

Two class LDA

- How can we interpret $p(y = 1|\mathbf{x}, \boldsymbol{\theta}) \quad = \quad \text{sigm}(\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0))$?
- The final decision rule is as follows: shift $\mathbf{x}$ by $\mathbf{x}_0$, project onto the line $\mathbf{w}$, and see if the result is positive or negative.
- The class prior, π_c, just changes the decision threshold, and not the overall geometry
- The magnitude of $\mathbf{w}$ determines the steepness of the logistic function, and depends on how well-separated the means are, relative to the variance.
  - One can define the **discriminability** of a signal from the background noise using a quantity called **d-prime**, in which μ_1 is the mean of the signal and μ_0 is the mean of the noise

$$d' \triangleq \frac{\mu_1 - \mu_0}{\sigma}$$

MLE for discriminant analysis

The simplest way to fit a discriminant analysis model is to use maximum likelihood. The log-likelihood function is

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \left[\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_i = c)\log \pi_c\right] + \sum_{c=1}^{C}\left[\sum_{i:y_i=c}\log \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\right]$$

- The class prior terms π_c are calculated as the empirical percentage of each class $\hat{\pi}_c = \frac{N_c}{N}$
- For the class-conditional densities, we just partition the data based on its class label, and compute the MLE for each Gaussian:

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i:y_i=c}\mathbf{x}_i, \quad \hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c}\sum_{i:y_i=c}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T$$

Strategies for preventing overfitting

- The MLE can badly overfit in high dimensions.
  - In particular, the MLE for a full covariance matrix is singular if $N_c < D$. And even when $N_c > D$, the MLE can be ill-conditioned, meaning it is close to singular (doesn't have an inverse matrix)

- Solutions:
  - Use a diagonal covariance matrix for each class, which assumes the features are conditionally independent; this is equivalent to using a naive Bayes classifier.
  - Use a full covariance matrix, but force it to be the same for all classes, $\Sigma_c = \Sigma$. This is an example of **parameter tying** or **parameter sharing,** and is equivalent to LDA.
  - Use a diagonal covariance matrix *and* forced it to be shared. This is called diagonal covariance LDA.
  - Use a full covariance matrix, but impose a prior and then integrate it out. If we use a conjugate prior, this can be done in closed form. Analogous to *Bayesian naive Bayes.*
  - Project the data into a low dimensional subspace and fit the Gaussians there.

Diagonal LDA

- A simple alternative to LDA is to tie the covariance matrices, so $\Sigma_c = \Sigma$ as in LDA, and then to use a diagonal covariance matrix for each class.
- The corresponding discriminant function is as follows

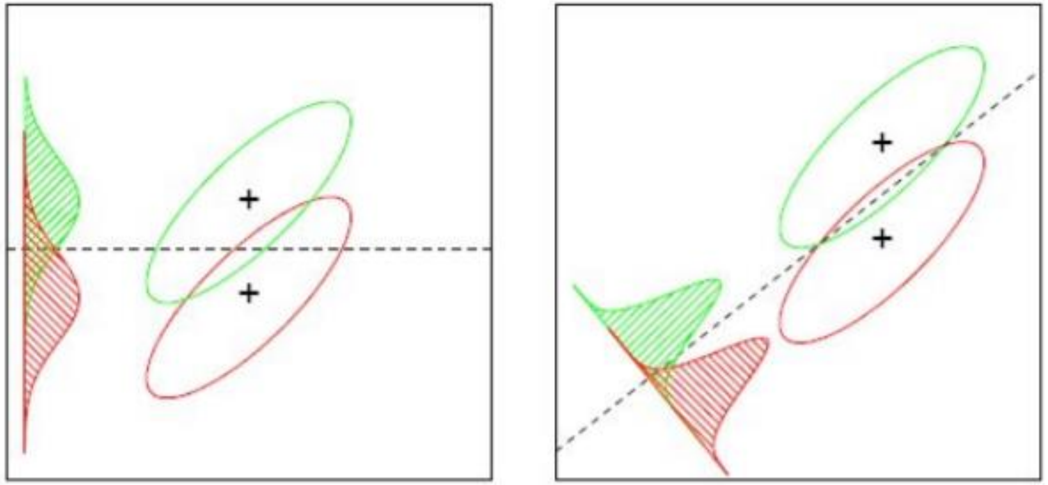
$$\delta_c(\mathbf{x}) = \log p(\mathbf{x}, y = c|\boldsymbol{\theta}) = -\sum_{j=1}^{D} \frac{(x_j - \mu_{cj})^2}{2\sigma_j^2} + \log \pi_c$$

- Typically, we set $\hat{\mu}_{cj} = \overline{x}_{cj}$ and $\hat{\sigma}_j^2 = s_j^2 = \frac{\sum_{c=1}^{C} \sum_{i:y_i=c} (x_{ij} - \overline{x}_{cj})^2}{N - C}$

- In high dimensional settings, this model can work much better than LDA .

LDA for dimensionality reduction

- LDA can be used to perform supervised dimensionality reduction, by projecting the input data to a linear subspace consisting of the directions which maximize the separation between classes
- The dimension of the output is necessarily less than the number of classes (C-1), so this is (in general) a rather strong dimensionality reduction, and only makes sense in a multiclass setting.
- An example of an LDA dimensionality reduction with two classes from 2D to 1D space



Inference in jointly Gaussian distributions

Given a joint distribution, $p(x_1, x_2)$, it is useful to compute marginals $p(x_1)$ and conditionals $p(x_1|x_2)$.

Suppose $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ is jointly Gaussian with $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix}$

The **marginals** are

$$\begin{aligned} p(\mathbf{x}_1) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ p(\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \end{aligned}$$

The **posterior conditionals** are

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}$$

The conditional mean is a linear function of $\mathbf{x}_2$, and the conditional covariance is a constant matrix independent of $\mathbf{x}_2$.

Marginals and conditionals of a 2d Gaussian

- Let us consider a 2d example where the covariance matrix is $\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$

- The marginal $p(x_1)$ is a 1D Gaussian, obtained by projecting the joint distribution onto the x_1 line

$$p(x_1) = \mathcal{N}(x_1|\mu_1, \sigma_1^2)$$

- Suppose we observe $X_2 = x_2$; the conditional $p(x_1|x_2)$ is obtained by "slicing" the joint distribution through the $X_2 = x_2$ line

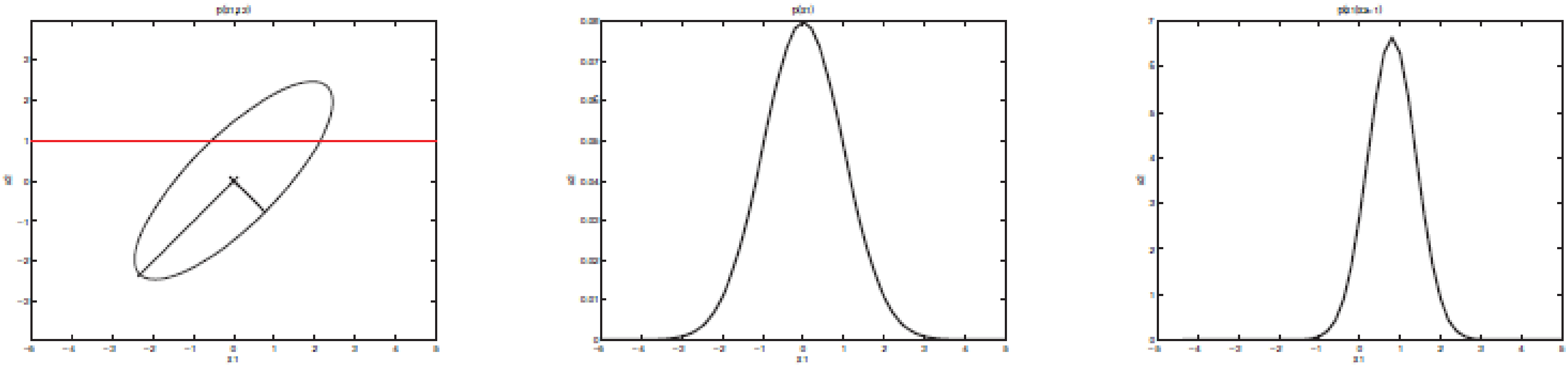
$$p(x_1|x_2) \quad = \quad \mathcal{N}\left(x_1|\mu_1 + \frac{\rho\sigma_1\sigma_2}{\sigma_2^2}(x_2 - \mu_2),\ \sigma_1^2 - \frac{(\rho\sigma_1\sigma_2)^2}{\sigma_2^2}\right)$$

- If $\sigma_1 = \sigma_2 = \sigma$

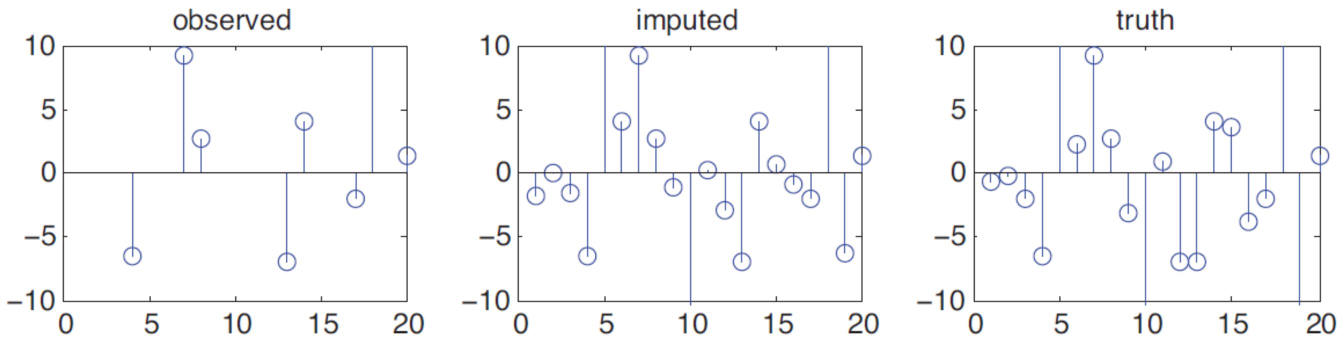
$$p(x_1|x_2) \quad = \quad \mathcal{N}\left(x_1|\mu_1 + \rho(x_2 - \mu_2),\ \sigma^2(1 - \rho^2)\right)$$

Marginals and conditionals of a 2d Gaussian

- An example where $\rho = 0.8$, $\sigma_1 = \sigma_2 = 1$, $\mu = 0$ and $x_2 = 1$.
  - $E[x_1|x_2=1] = 0.8$
  - $\mathrm{var}[x_1|x_2=1] = 1 - 0.8^2 = 0.36$
  - If $\rho = 0$, we get $p(x_1|x_2) = p(x_1)$



Data imputation



- Suppose we are missing some entries in a design matrix. If the columns are correlated, we can use the observed entries to predict the missing entries.
- We sampled some data from a 20-dimensional Gaussian, and then deliberately "hid" 50% of the data in each row.
- We then inferred the missing entries given the observed entries, using the true (generating) model.
- More precisely, for each row $\boldsymbol{i}$, we compute $p(\mathbf{x}_{\mathbf{h}_i}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta})$, where $\mathbf{h}_i$ and $\mathbf{v}_i$ are the indices of the hidden and visible entries in case $\boldsymbol{i}$.
- From this, we compute the marginal distribution of each missing variable, $p(x_{h_{ij}}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta})$.
- We then plot the mean of this distribution, $\hat{x}_{ij} = E\left[x_j \middle| \mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta}\right]$, which represents our "best guess" about the true value of that entry
- We can use $\mathrm{var}[x_{h_{ij}}|\mathbf{x}_{\mathbf{v}_i}, \boldsymbol{\theta}]$ as a measure of confidence in this guess

Information form

- Suppose $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix are the **moment parameters**
- It is sometimes useful to use the **canonical parameters** $\boldsymbol{\Lambda} \triangleq \boldsymbol{\Sigma}^{-1}, \ \boldsymbol{\xi} \triangleq \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$

and the MVN in **information form** $\mathcal{N}_c(\mathbf{x}|\boldsymbol{\xi}, \boldsymbol{\Lambda}) = (2\pi)^{-D/2}|\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \boldsymbol{\xi}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi} - 2\mathbf{x}^T\boldsymbol{\xi})\right]$

- The **marginalization** and **conditioning** formulas in information form are

$$
\begin{aligned}
p(\mathbf{x}_2) &= \mathcal{N}_c(\mathbf{x}_2|\boldsymbol{\xi}_2 - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\xi}_1, \boldsymbol{\Lambda}_{22} - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}) \\
p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}_c(\mathbf{x}_1|\boldsymbol{\xi}_1 - \boldsymbol{\Lambda}_{12}\mathbf{x}_2, \boldsymbol{\Lambda}_{11})
\end{aligned}
$$

- marginalization is easier in moment form
- conditioning is easier in information form

- **Multiplying** two Gaussians is simply expressed as $\mathcal{N}_c(\xi_f, \lambda_f)\mathcal{N}_c(\xi_g, \lambda_g) = \mathcal{N}_c(\xi_f + \xi_g, \lambda_f + \lambda_g)$

while in moment form it is much messier $\mathcal{N}(\mu_f, \sigma_f^2)\mathcal{N}(\mu_g, \sigma_g^2) = \mathcal{N}\left(\frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\sigma_g^2 + \sigma_g^2}, \frac{\sigma_f^2\sigma_g^2}{\sigma_g^2 + \sigma_g^2}\right)$

Linear Gaussian systems

- Suppose we have two variables, $\mathbf{x}$ and $\mathbf{y}$. Let $\mathbf{x} \in R^{D_x}$ be a hidden variable, and $\mathbf{y} \in R^{D_x}$ be a noisy observation of $\mathbf{x}$. Let us assume we have the following prior and likelihood:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Sigma}_y)$$

where $\mathbf{A}$ is a matrix of size $D_y \times Dx$. This is an example of a linear Gaussian system.

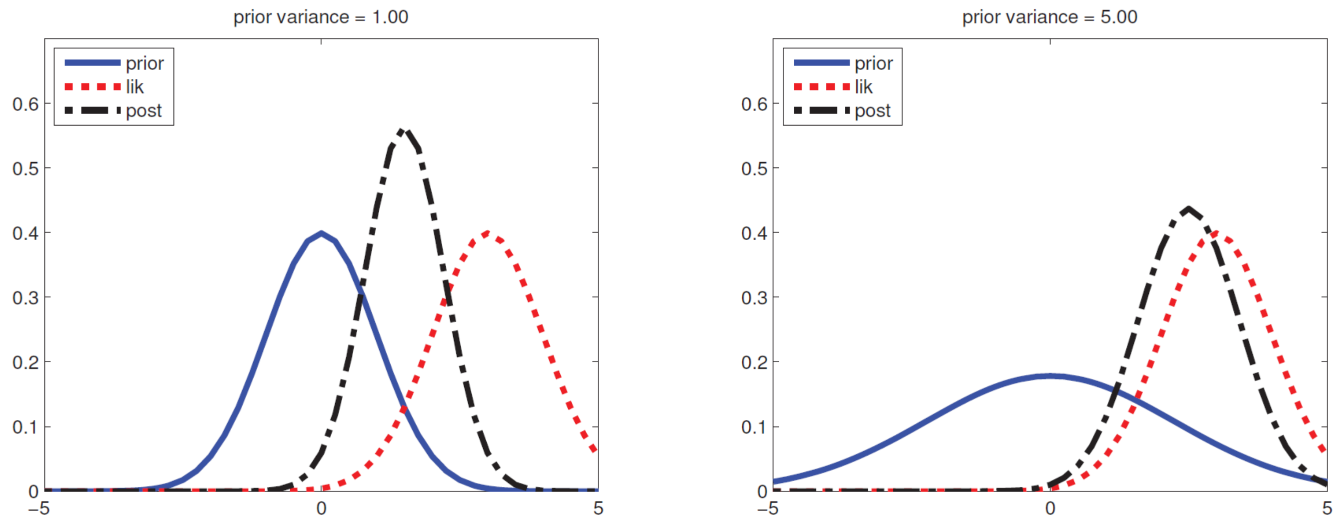
- We can then infer x from y by using the Bayes rule

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y})$$
$$\boldsymbol{\Sigma}_{x|y}^{-1} = \boldsymbol{\Sigma}_x^{-1} + \mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A}$$
$$\boldsymbol{\mu}_{x|y} = \boldsymbol{\Sigma}_{x|y}[\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x]$$
$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}_x + \mathbf{b}, \boldsymbol{\Sigma}_y + \mathbf{A}\boldsymbol{\Sigma}_x\mathbf{A}^T)$$

Inferring an unknown scalar from noisy measurements

- Suppose we make N noisy measurements $\mathbf{y}_i$ of some underlying quantity $\mathbf{x}$; let us assume the measurement noise has fixed precision $\lambda_y = 1/\sigma^2$, so the likelihood is $p(y_i|x) = \mathcal{N}(y_i|x, \lambda_y^{-1})$
- Let us use a Gaussian prior for the value of the unknown source $p(x) = \mathcal{N}(x|\mu_0, \lambda_0^{-1})$
- The resulting posterior is

$$
\begin{aligned}
p(x|\mathbf{y}) &= \mathcal{N}(x|\mu_N, \lambda_N^{-1}) \\
\lambda_N &= \lambda_0 + N\lambda_y \\
\mu_N &= \frac{N\lambda_y}{N\lambda_y + \lambda_0}\bar{y} + \frac{\lambda_0}{N\lambda_y + \lambda_0}\mu_0
\end{aligned}
$$

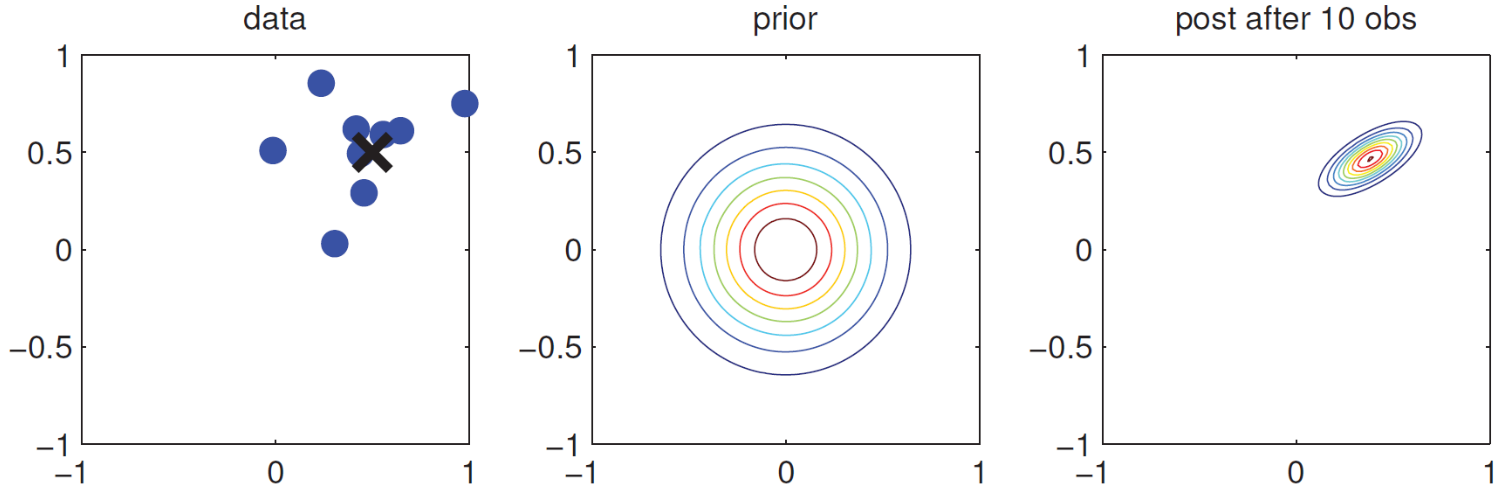


Inferring an unknown vector from noisy measurements

- Now consider N vector-valued observations, $\mathbf{y}_i \sim N(\mathbf{x}, \boldsymbol{\Sigma}_y)$, and a Gaussian prior, $x \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
- If we set $\mathbf{A} = \mathbf{I}$ and $\mathbf{b} = 0$

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{y}_1, \ldots, \mathbf{y}_N) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \\
\boldsymbol{\Sigma}_N^{-1} &= \boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\Sigma}_y^{-1} \\
\boldsymbol{\mu}_N &= \boldsymbol{\Sigma}_N(\boldsymbol{\Sigma}_y^{-1}(N\overline{\mathbf{y}}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0)
\end{aligned}
$$

- A 2D example:



Combining measurements from different devices

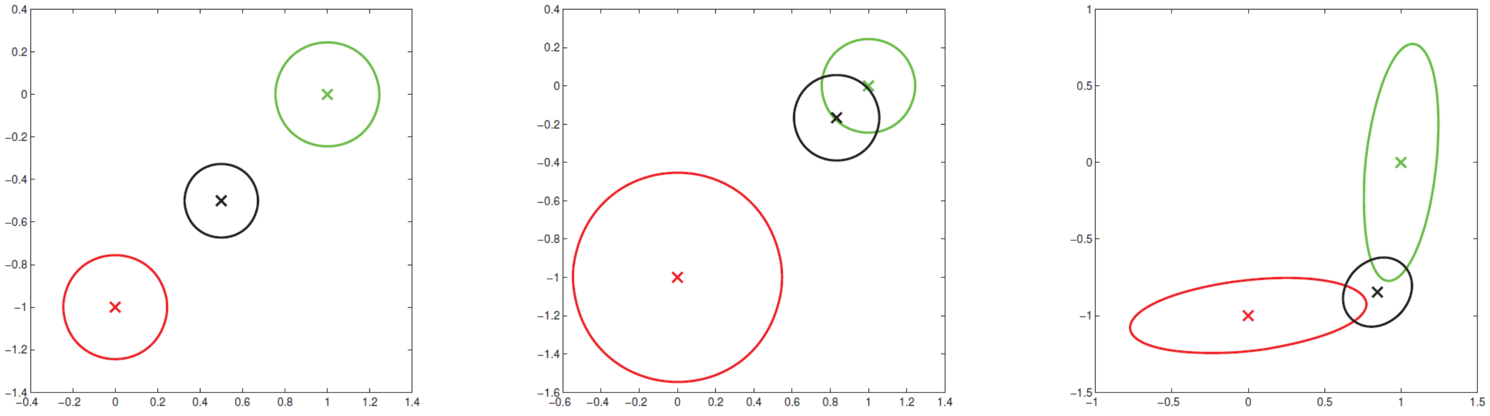
If we have multiple measuring devices, we can combine them together (**sensor fusion**).

If we have multiple observations with different covariances (sensors with different reliabilities), the posterior will be an appropriate weighted average of the data

(a) Equally reliable sensors, $\boldsymbol{\Sigma}_{y,1} = \boldsymbol{\Sigma}_{y,2} = 0.01\mathbf{I}_2$

(b) Sensor 2 is more reliable, $\boldsymbol{\Sigma}_{y,1} = 0.05\mathbf{I}_2$ and $\boldsymbol{\Sigma}_{y,2} = 0.01\mathbf{I}_2$

(c) Sensor 1 is more reliable in the vertical direction, and Sensor 2 in the horizontal direction. $\boldsymbol{\Sigma}_{y,1} = 0.01\begin{pmatrix} 10 & 1 \\ 1 & 1 \end{pmatrix}$, $\boldsymbol{\Sigma}_{y,2} = 0.01\begin{pmatrix} 1 & 1 \\ 1 & 10 \end{pmatrix}$



(a) (b) (c)

Inferring the parameters of an MVN

As in previous models, to infer the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ of the Gaussian we can

- find a point estimate (as with MLE) $p(\boldsymbol{\mu}|D, \boldsymbol{\Sigma}) = N(\bar{\mathbf{x}}, \frac{1}{N}\boldsymbol{\Sigma})$
- calculate the posterior distribution of each parameter
  - add a prior for each parameter (Gaussian or Normal-Inverse-Wischart)
  - find the product of the prior and likelihood

The posterior distribution are usually a convex combination of the prior and the MLE, like:

$$
\begin{aligned}
p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\Sigma}) &= \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_N, \mathbf{V}_N) \\
\mathbf{V}_N^{-1} &= \mathbf{V}_0^{-1} + N\boldsymbol{\Sigma}^{-1} \\
\mathbf{m}_N &= \mathbf{V}_N(\boldsymbol{\Sigma}^{-1}(N\overline{\mathbf{x}}) + \mathbf{V}_0^{-1}\mathbf{m}_0)
\end{aligned}
$$

$$
\begin{aligned}
p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) &= \mathrm{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{m}_N, \kappa_N, \nu_N, \mathbf{S}_N) \\
\mathbf{m}_N &= \frac{\kappa_0\mathbf{m}_0 + N\overline{\mathbf{x}}}{\kappa_N} = \frac{\kappa_0}{\kappa_0 + N}\mathbf{m}_0 + \frac{N}{\kappa_0 + N}\overline{\mathbf{x}} \\
\kappa_N &= \kappa_0 + N \\
\nu_N &= \nu_0 + N \\
\mathbf{S}_N &= \mathbf{S}_0 + \mathbf{S}_{\overline{x}} + \frac{\kappa_0 N}{\kappa_0 + N}(\overline{\mathbf{x}} - \mathbf{m}_0)(\overline{\mathbf{x}} - \mathbf{m}_0)^T
\end{aligned}
\qquad \mathbf{S} \triangleq \textstyle\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^T
$$

Conclusion

The analysis is quite sensitive to outliers and the size of the smallest group must be larger than the number of features.

Assumptions:

- Multivariate normality: features are normal for each class
- Homogeneity of variance/covariance (homoscedasticity): Variances among group variables are the same across levels of predictors. (If we use LDA)
- Multicollinearity: Predictive power can decrease with an increased correlation between features
- The between class decision function is linear or quadratic

Materials

Kevin P. Murphy - Machine Learning A Probabilistic Perspective, Chapter 4

Scikit-learn: Linear and Quadratic Discriminant Analysis