Multivariate Data

\[ X = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & \cdots & \cdots & X_d^2 \\ \vdots & \ddots & \ddots & \vdots \\ X_1^N & \cdots & \cdots & X_d^N \end{bmatrix} \]

d-columns - d variables
inputs (features/attributes)
N-rows - i.i.d.
observations (examples/instances)

E.G. Loan application -
customer observation vector - age, marital status, income, etc.
Typically, these variables are correlated
If not, great simplification.
Goals: simplifying / summarizing - explore data by few parameters,
exploring, generate hypotheses about data.

Parameter Estimation

\[ E[X] = \mu = [\mu_1, \ldots, \mu_d] \]

Variance of \( X_i \) = \( \sigma_i^2 \)
Covariance \( \Sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)] \)

\( \Sigma \) is a matrix, \( \Sigma_{ii} = \sigma_i^2 \)

\[ \Sigma = E[(X - \mu)(X - \mu)^T] = E[XX^T] - \mu \mu^T \]

Correlation \( \text{Corr} (X_i, X_j) = \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \)

If variables are independent, then correlation and covariance is zero.
Given multivariate sample

\[ \mathbf{m} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad \text{and} \quad \mathbf{m}_c = \frac{1}{N} \sum_{i=1}^{N} x_i \]

Sample covariance

\[ S_{ij} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mathbf{m}) (x_j - \mathbf{m}) \]

Sample correlation

\[ r_{ij} = \frac{S_{ij}}{S_{i}S_{j}} \]

Missing values - some variables may be missing in observations.

Imputation - estimate the missing variable

- Mean imputation - substitute the mean of other observations
- Regression imputation - predict these values from existing ones
  - If so, need to model why it is missing at random
  - E.g., heights of Americans - Army data

Multivariate Normal

\[ p(x) = \frac{1}{(2\pi)^{d/2} \left| \Sigma \right|^{1/2}} e^{-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)} \]

Mahalanobis distance

\[ (x - \mu)^T \Sigma^{-1} (x - \mu) = c^2 \]

Level sets of hyperellipsoids centered on \( \mu \), shape & orientation determined by \( \Sigma \).
Multivariate Normal (Cont.)

Bivariate case ($d=2$)

Level sets.

$y \sim N_d(\mu, \Sigma)$

then each dimension of $y$ is a univariate normal (converge not necessarily true).

Special case - if the components of $y$ are independent then the correlation covariance are 0.

$p(y) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi \sigma_i^2}} \exp \left( -\frac{1}{2} \frac{(y_i - \mu_i)^2}{\sigma_i^2} \right) \cdot (-2\pi \Sigma)^k / \prod_{i=1}^{d} \sigma_i^2$.

(Note: we can always pick a coordinate rep for which this is true by rotating space to diagonal.

$\Sigma \rightarrow \Phi^T \Sigma \Phi \rightarrow \Phi$ rotation.

$\Phi$ can be obtained from eigenvectors/eigenvectors of $\Sigma$).

Important Property

the projection of a d-dimensional normal on a vector $v$ is also a Gaussian.

$w^T x \sim N(w^T \mu, w^T \Sigma w)$

$w^T x \sim N((w^T \mu, w^T \Sigma w)$

$w^T x \sim N(0, w^T \Sigma w)$

$w^T x \sim N_k(w^T \mu, w^T \Sigma w)$

$w$ dkx matrix.

$\text{rank} k < d$.
Multivariate Classifier

\[ P(x|c_i) = \frac{1}{(2\pi)^{d/2} \Sigma_i^{1/2}} e^{-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)} \]

\text{(analytic solution)}

\text{Discriminant function}

\[ g_i(x) = \log P(x|c_i) + \log P(c_i) \]

\[ g_i(x) = -\frac{1}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_i| - \frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \]

\[ + \log P(c_i) \]

Given training samples \( \mathcal{X} = \left\{ x^1, x^2, \ldots, x^N \right\} \)

Estimate the distribution \( \hat{P}(c_i) = \frac{\# x^j \in \mathcal{X} \cap c_i}{\# \mathcal{X}} \)

\[ m_i = \frac{\sum_{x^j \in \mathcal{X} \cap c_i} x^j}{\# x^j \in \mathcal{X} \cap c_i} \]

\[ S_i = \sum_{x^j \in \mathcal{X} \cap c_i} (x^j - m_i)(x^j - m_i)^T \frac{1}{\# x^j \in \mathcal{X} \cap c_i} \]

Plug into discriminant function:

\[ g_i(x) = -\frac{1}{2} \log |S_i| - \frac{1}{2} (x - m_i)^T S_i^{-1} (x - m_i) \]

\text{(dropping constant term)}

\text{Quadratic Discriminant (dropping constant term)}

\[ g_i(x) = x^T \omega_i x + \omega_i^T x + \omega_i \]

\[ \omega_i = -\frac{1}{2} S_i^{-1} \mu_i \]

\[ \omega_i = -\frac{1}{2} S_i^{-1} \mu_i \]

\text{No. parameters} \( K \cdot d + K \cdot c + K \) common
For large $d$, samples are small, $\Sigma_i$ may be singular and $\Sigma^{-1}$ won't exist. They need to decrease dimensionality (next lecture) or pool data and estimate common covariance matrix: $\hat{\Sigma} = \frac{1}{N} \sum_i P(C_i) \Sigma_i$. 

**Discriminant reduces to** 

$$g_i(x) = -\frac{1}{2} (x-m_i)^T \Sigma_i^{-1} (x-m_i) + \log \hat{P}(C_i)$$

**Further simplification**

assuming off-diagonal elements covariances are zero.

**Naive Bayes classifier** $P(x|C_i)$ univariate Gaussian

$g_i(x) = -\frac{1}{2} \sum_{j=1}^{p} \left( x_j - \mu_{ij} \right)^2 + \log \hat{P}(C_i)$

$\mu_{ij}$ is mean of $j^{th}$ component of $i^{th}$ class.

$\Rightarrow$ reduces complexity of $\Sigma$ from $O(p^2)$ to $O(p)$.

**Extreme simplification**: assume all variances are the same. Then Mahalanobis distance $= $ Euclidean distance

$$g_i(x) = -\frac{1}{2} \sum_{j=1}^{p} (x_j - \mu_{ij})^2 + \log \hat{P}(C_i)$$

If priors are equal, $g_i(x) = -\frac{1}{2} ||x-m_i||^2$ nearest mean classifier. $\Rightarrow$ prototype/template

$$g_i(x) = w_i^TX + w_{i0} \quad w_i = m_i, \quad w_{i0} = -\frac{1}{2} ||m_i||^2$$
Tuning Complexity

As before, for polynomial regression, we can choose between models with different complexity:

- Full Gaussianness: \( d + \frac{1}{2} d(d+1) \) parameters
- Independent Variables: \( d + 1 \)

Same procedure as before:

(e.g. cross-validation, regularization, etc.)

Discrete Features:

Discrete attributes - color \( \in \{ \text{red, blue, green} \} \), pixel \( \in \{ \text{on, off} \} \)

Suppose \( x_j \) are binary (Bernoulli):

\[
\begin{align*}
\hat{p}_{ij} &= p(x_j=1|c_i) \\
p(x|c_i) &= \prod_{j=1}^{d} p_{ij} x_j (1 - p_{ij})^{(1-x_j)}
\end{align*}
\]

Discriminant:

\[
q_i(x) = \log p(x|c_i) + \log p(c_i)
\]

\[
= \sum_j x_j \log p_{ij} + (1 - x_j) \log (1 - p_{ij}) + \log p(c_i)
\]

Estimate:

\[
\hat{p}_{ij} = \frac{1}{d} x_j \frac{r_{ij}}{\sum_k r_{ik}}
\]

General Case: multinomial \( x_j \) chosen from \( (v_1, ..., v_{k_j}) \)

\[
\begin{align*}
\hat{p}_{ij} &= \text{prob} (x_j \text{ from class } c_i \text{ takes value } v_k) \\
\hat{p}_{jk} &= p(z_{jk}=1|c_i) = p(x_j=v_k|c_i)
\end{align*}
\]
Discrete Features (cont.)

If attributes are independent, then

\[ p(x|C_i) = \prod_{j=1}^{n} \prod_{k=1}^{z_{jk}} \pi_{jk} \]

The discriminant function is

\[ g_i(x) = \frac{1}{z_i} \sum_{j=1}^{n} z_{jk} \log \pi_{jk} + \log \hat{p}(C_i) \]

Max likelihood estimate for \( \pi_{jk} \) is

\[ \hat{\pi}_{jk} = \frac{1}{z_i} \frac{Z_{jk}^t r^t}{Z r^t} \]

Multivariate Regression

\[ r^t = g(x^t|\omega_0, \ldots, \omega_d) = \omega_0 + \omega_1 x_1^t + \ldots + \omega_d x_d^t + \epsilon \]

Least squares:

\[ E(\omega_0, \ldots, \omega_d | X) = \frac{1}{2} \sum_{t} (r^t - \omega_0 - \omega_1 x_1^t - \ldots - \omega_d x_d^t)^2 \]

\[ X = \begin{bmatrix} 1 & x_1^t & \cdots & x_d^t \end{bmatrix}, \quad \omega = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_d \end{bmatrix}, \quad r = \begin{bmatrix} r_1 \\ \vdots \\ r_d \end{bmatrix} \]

Minimize \( \omega \cdot \cdot \cdot \cdot r \)

\[ \bar{\omega} = (X^t X)^{-1} X^t r \]