Multivariate Data

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

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

EG. Loan application -
- customer observation vector - age, marital status, income, e.t.
- Typically, these variables are correlated
  - If not, great simplification.

Goals:
- simplify/summarize - explain data by few patterns.
- exploratory, - generate hypothesis about data.

Parameter Estimation

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

\[ \text{Variance of } X_i \ (i) = \sigma_i^2, \ \text{Covariance} \ \sigma_{ij} = \text{cov}(X_i, X_j) \]

\[ \Sigma = \text{a matrix, } \Sigma_{ii} = \sigma_i^2, \ \Sigma_{ij} = \sigma_{ij}. \]

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

Correlation

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

If variables are indep, then correlation and covariance is zero.
Parameter Estimation (cont.)

Given multivariate sample

sample mean \[ \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \], \[ m_i = \frac{1}{N} \sum_{i=1}^{N} x_i \]

Sample covariance \[ s_{ij} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - m_i)(x_j - m_j) \]

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 observed variables
- Regression imputation – predict the values from existing ones
- Data may not be missing at random
  - If so, need to model why it is missing
  - e.g., heights of Americans – Army data

Multivariate Normal

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

Mahalanobis distance \[ (x - \mu)^T \Sigma^{-1} (x - \mu) \]

Level sets \[ (x - \mu)^T \Sigma^{-1} (x - \mu) = c^2 \] are multidimensional hyperellipsoids centered on \( \mu \), shape and orientation determined by \( \Sigma \).
(3) **Multivariate Normal (Cont.)**

Bivariate case \((d=2)\)

\[ (X \sim N_d(\mu, \Sigma) ) \]

then each dimension of \(X\) is a univariate normal \((even \ not \ necessarily \ true)\).

Special case - if the components of \(X\) are independent then the correlation matrix \(\Sigma\) is diagonal.

\[ p(x) = \prod_i p(x_i) = \frac{1}{(2\pi)^{d/2}} \prod_i e^{-\frac{1}{2} (x_i - \mu_i)^2} \]

(Note: we can always pick a coordinate rep for which this is true \(\rightarrow\) by rotating space to diagonal. \(\Sigma = \Phi^T \Delta \Phi\) \(\Phi\) rotation \(\Phi\) can be obtained from eigenvalues/eigenvectors of \(\Sigma\))

**Important Property**

the projection of a d-dim normal onto a vector is also a Gaussian.

\[ \tilde{y} = \text{proj}_w \tilde{x} \]

\[ \tilde{y} \sim N(\Phi \tilde{\mu}, \Phi \Sigma \Phi^T) \]

\[ \tilde{w} \tilde{x} \sim N(\tilde{w} \tilde{\mu}, \tilde{w} \tilde{\Sigma} \tilde{w}) \]

\(\tilde{w}\) - std matrix.
Multivariate Classification

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

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

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

Given training samples \( X = \{x^1, \ldots, x^n \} \)

Estimate the distribution

\[ \hat{p}(x) = \frac{1}{n} \sum_i r_i \]

\[ \hat{m}_i = \frac{1}{\hat{r}_i} \sum x_i \]

\[ \hat{S}_i = \frac{1}{\hat{r}_i - 1} \sum (x_i - \hat{m}_i)(x_i - \hat{m}_i)^T \]

Plug into discriminant function

\[ g_i(x) = -\frac{1}{2} \log |\hat{S}_i| - \frac{1}{2} (x - \hat{m}_i)^T \hat{S}_i^{-1} (x - \hat{m}_i) + \log \hat{P}(C_i) \]

Quadratic Discriminant

(dropping constant term)

Can be written as

\[ g_i(x) = x^T \hat{W}_i x + \hat{w}_i^T x + \hat{w}_i \]

\[ \hat{W}_i = -\frac{1}{\hat{r}_i} \hat{S}_i^{-1}, \quad \hat{w}_i = \hat{S}_i^{-1} \hat{m}_i, \]

\[ \hat{w}_i = -\frac{1}{2} \hat{m}_i^T \hat{S}_i^{-1} \hat{m}_i - \frac{1}{2} \log |\hat{S}_i| + \log \hat{P}(C_i) \]

No. Parameters: \( K \cdot d + K \cdot d + K \cdot d + K \cdot d + 2K \cdot d + d + 1 \)
Multivariate Classification (Cont.)

For large \( d \), samples are small, \( \Sigma \) may be singular and \( \Sigma^{-1} \) won't exist. 

They need to decrease dimensionality (next tech) or pool data and estimate common covariance matrix:

\[
\Sigma = \Sigma_i = \frac{1}{N} \sum (x_i - \mu_i)(x_i - \mu_i)^T.
\]

Discriminant reduces to:

\[
g_i(x) = -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) + \log P(C_i).
\]

Further simplification:

- Assume off-diagonal elements covariance are zero.
- Naïve Bayes classifier:

\[
p(x | C_i) \text{ univariate Gaussian}\]

\[
g_i(x) = -\frac{1}{2} \sum (x_j - \mu_j)^2 + \log P(C_i).
\]

\( \mu_j \) is mean of \( j \)-th component of \( i \)-th class

\( \rightarrow \) reduces complexity of \( \Sigma \) from \( O(n^2) \) to \( O(n) \).

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

\[
g_i(x) = -\frac{1}{2} \sum (x_j - \mu_j)^2 + \log P(C_i).
\]

If priors equal, nearest mean classifier. \( \rightarrow \) prototype/template

\[
g_i(x) = \omega_i^T x + \omega_0 \quad \omega_i = \mu_i, \quad \omega_0 = -\frac{1}{2} \| \mu_i \|^2
\]
Tuning Complexity

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

- **Full Correlation**: $d + \frac{1}{2} d(d+1)$ parameters
- **Indep.-Same Varies**: $d + 1$

Same procedure as before:
(e.g. cross-validation, regularization, etc.)

**Discriminate Features**:

Discriminate attributes - color $\in \{\text{red, blue, green}\}$

Suppose $x_j$ are binary (Bernoulli)

- $p_{ij} = p(x_{j}=1 | c_i)$
- $p(x | c_i) = \prod_{j=1}^{d} p_{ij} x_{j} (1 - p_{ij})^{(1-x_{j})}$

Discriminant: $g_i(x) = \log p(x | c_i) + \log P(c_i)$

$= \sum_{j} (x_{j} \log p_{ij} + (1-x_{j}) \log (1-p_{ij})) + \log P(c_i)$

Estimator: $\hat{p}_{ij} = \frac{\sum x_{j} r_{c_i}^{t}}{\sum r_{c_i}^{t}}$

**General Case**: multinomial $x_j$ chosen from $(v_1 \ldots v_{n_j})$

- $p_{ijk} = \text{prob } (x_j \text{ from class } c_i \text{ takes value } v_k)$
- $p_{ijk} = p(z_{jk} = 1 | c_i) = p(x_j = v_k | c_i)$
Discrete Features (cont.)

\[ z_{jk}^t = \sum_{i=1}^{k_{ij}} q_{ji} x_{ij}^t = y_k \]

\[ p(x|C) = \prod_{j=1}^{K} \prod_{i=1}^{C_j} p(i|a) \]

\[ g_i(x) = \sum_{j=1}^{K} \sum_{k=1}^{C_j} z_{jk}^t \log \hat{p}(i|k) + \log \hat{p}(C_i) \]

Max like estimate for \( \hat{p}(i|k) \) is

\[ \hat{p}(i|k) = \frac{z_{jk}^t}{\sum_{j=1}^{K} z_{jk}^t} \]

Multivariate Regression

\[ r^t = \begin{bmatrix} C^t \end{bmatrix} \omega \rightarrow y = \omega_0 + \omega_1 x_1 + \cdots + \omega_d x_d + \epsilon \]

Least squares:

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

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

Minimize \( \omega \) s.t.

\[ X^T X \omega = X^T r \]

\[ \hat{\omega} = (X^T X)^{-1} X^T r \]