6- numerical estimation and regression

regression

input variables = regressors, independent / predictor / explantory / exposure variables

output variables = regressands, dependent / response / explained / outcome variables

☆ if the model is non-linear, utilize the expectation maximization algorithm

linear regression = $y = mx + b \xrightarrow{\text{for noice}}$ random variable ← $Y_j = \alpha + \beta x_j + E_j$ ← error variable with mean 0 (for unbiased estimator)

↳ unknown parameters to be estimated

the least-squares estimator = $\sum_{j=1}^{N} (E_j)^2 = \sum_{j=1}^{N} (Y_j - (\alpha + \beta x_j))^2$ minimize

↳ $\hat{\beta} \rightarrow \frac{\partial \sum (E_j)^2}{\partial \hat{\beta}} = \sum -2x_j(y_j - \hat{\alpha} - \hat{\beta} x_j) = 0$

↳ $\hat{\alpha} \rightarrow \frac{\partial \sum (E_j)^2}{\partial \hat{\alpha}} = \sum -2(y_j - \hat{\alpha} - \hat{\beta} x_j) = 0$

$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$

$\hat{\beta} = \sum (y_j - \bar{y})(x_j - \bar{x}) / \sum (x_j - \bar{x})^2$

weighted least squares = normally error of each point has different var → $\min \left(\sum w_j (Y_j - \alpha - \beta x_j)^2 \right)$

multiple regression = $Y_j = \alpha + \beta_1 x_{j1} + \beta_2 x_{j2} + \ldots + \beta_k x_{jk} + E_j$

↳ matrix form: $Y = C\beta + E$

$Y = (Y_1, Y_2 \ldots Y_n)^T$

$\beta = (\alpha, \beta_1, \beta_2 \ldots \beta_k)^T$

$E = (E_1, E_2 \ldots, E_n)^T$

$$C = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix} \longrightarrow \underline{\text{estimate}}\ \hat{\beta} = (C^T C)^{-1} C^T Y$$

polynomial regression = $Y_j = \alpha + \beta_1 x_j + \beta_2 x_j^2 + \ldots + \beta_k x_j^k + E_j$

↳ change it like ($x = x_j$, $y = x_j^2$, $z = x_j^3$) → predict $t = y$, like it is linear

multivariate normal (multinormal) = generalization of the one-dimensional normal/gaussian distribution to higher dimensions

↳ every combinations of dimensions is distributed normally

↳ $\Sigma_{ij} = \text{cov}(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)]$

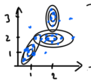
↳ symmetric and positive semi-definite

$$\Sigma_{ij} = \begin{bmatrix} E[(x_1-\mu_1)(x_1-\mu_2)] & E[(x_1-\mu_1)(x_2-\mu_2)] & \cdots & E[(x_1-\mu_1)(x_n-\mu_n)] \\ E[(x_2-\mu_1)(x_2-\mu_2)] & E[(x_2-\mu_1)(x_2-\mu_2)] & \cdots & E[(x_2-\mu_2)(x_n-\mu_n)] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & \cdots & E[(x_n-\mu_n)(x_n-\mu_n)] \end{bmatrix}$$

↳ $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ → no correlation

↳ $\Sigma = \begin{pmatrix} 1 & 0.75 \\ 0.75 & 1 \end{pmatrix}$ → positive correlation between x and y

mixtures of multinormals = when only a mixture of gaussians can represents data



$0.7\, N((2,2), \Sigma_1) + 0.15\, N((1,1), \Sigma_2) + 0.15\, N((2,3), \Sigma_3)$

↳ density function = $\sum_{i=1}^{k} w_i N(\mu_i, \Sigma_i)$ $\quad \left(\sum_{i=1}^{k} w_i = 1 \right)$

☆ if we know how many gaussians in the mixture:

① if which point belonged to which gaussian is given, the parameters can be found using maximum likelihood estimation

② if the parameters is given, we can find the assignments using expectation max algorithm ↗ uses max likelihood estimation again

expectation maximization algorithm = guess the assignments → use them to estimate parameters → use them to generate better assignments → repeat until no improvement (like k-means)

↳ expectation step = given parameters θ, find expected value on the latent (gizli) variables Z

$z = E[Z \mid X, \theta]$

↳ $(n \times k)$ matrix

$z_{ij} = \begin{cases} 1, & \text{if point } i \text{ in } j\text{th gaussian (assigning members to clusters)} \\ 0, & \text{otherwise} \end{cases}$

soft assignments = $0 \leq z_{ij} \leq 1$ (used here)

hard assignment = $z_{ij} = 1$, rest $z_{i*} = 0$

↳ maximization step = estimate new parameters based on max likelihood

$\theta' = \hat{\theta}_{MLE} = \arg\max_{\theta} \log L(\theta \mid x, z)$ $\quad \theta = \{ (\mu_1, \Sigma_1), \ldots (\mu_k, \Sigma_k) \}$ (calculating new means and covariance matrices)

③ if number of gaussians is unknown:

↳ $k = n$, each point gets its own gaussian, max possible likelihood → overfitting

↳ $k = 1$, simple low likelihood → underfitting

bayesian information criterion (BIC) = $\underbrace{-2 \log L(\theta \mid x, z)}_{\text{accuracy}} + \underbrace{k \log(n)}_{\text{penalize for more parameters}}$

must be lowest

k = # of regressors

n = # of points

linear model in R:

- lm(<response var> ~ <input var>, <data frame>)

simple regression:

my_linear_model <- lm(y ~ x, my_data_frame)

predict(my_linear_model , data.frame(x=1:5))

multivariable regression:

y = β_1x + β_2 + α

my_multi_regression_model <- lm(y ~ x + z, my_data_frame)

polynomial regression:
h(t) = k + v_0t + gt^2
lm(h~t+t2,data.frame(h=ball$h, t=ball$t, t2= ball$t^2))

- <<- global assignment in R