LIST OF FORMULASTOPICS FOR STK2100

(Version May 2022)

1 General issues

(a) The Bias-variance trade-off

(b) Training, test and validation sets

(c) Complexity/degrees of freedom

(i) Assuming $\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{y}$, degrees of freedom is defined as $\text{df} = \text{trace}(\boldsymbol{S})$.

(ii) One-to-one correspondance between degrees of freedom and penalty parameter.

(iii) Selection of df/penalty usually through cross-validation

(d) Loss functions

(i) For regression, one usually uses quadratic loss: $L(y, \hat{y}) = (y - \hat{y})^2$. The optimal predictor based in input variable(s) $\mathbf{x}$ is then $\widehat{Y} = E[Y|\mathbf{x}]$.

(ii) For classification we usually use 0-1 loss: $L(y, \hat{y}) = I(y = \hat{y})$ where $I(\cdot)$ is the indicator function. The optimal prediction based on input variable(s) $\mathbf{x}$ is then $\widehat{Y} = \text{argmax}_k \Pr(Y = k|\mathbf{x})$.

(e) Model selection criteria

(i) AIC defined by $\text{AIC} = -2l(\hat{\boldsymbol{\theta}}) + 2|\boldsymbol{\theta}|$ where $|\boldsymbol{\theta}|$ is the number of *free* parameters in the model.

(ii) BIC defined by $\text{BIC} = -2l(\hat{\boldsymbol{\theta}}) + \log(n)|\boldsymbol{\theta}|$.

(iii) K-folded cross-validation

i. Divide the N data points into K groups by randomization

ii. For $k = 1, ..., K$

A. Fit the model on all data *except* data from group k.

B. Predict $\hat{y}_i$ for all data in group k based on the fitted model

C. Calculate $\text{RSS}_i = (y_i - \hat{y}_i)^2$ for i in group k

iii. Calculate $\text{CV}_{(K)} = \frac{1}{N}\sum_{i=1}^{N} \text{RSS}_i$

(f) Principal components: 1. component defined by $\boldsymbol{z}_{i1} = \boldsymbol{a}_1^T \boldsymbol{x}_i$ with

$$\boldsymbol{a}_1 = \underset{\boldsymbol{a}}{\text{argmax}}\, \boldsymbol{a}^T \boldsymbol{S} \boldsymbol{a} \quad \text{subject to} \quad \boldsymbol{a}^T \boldsymbol{a} = 1$$

where $\boldsymbol{S}$ is the sample covariance matrix. Next components defined similarly.

2 Multiple linear regression

(a) Model:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i \,;\; i = 1, 2, \ldots, N \,;$$

where the x_{ij}'s are known numbers and the ϵ_i's are independent and $N(0, \sigma^2)$-distributed.

(b) Matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{Y} = (Y_1, \ldots, Y_N)^T$ and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^T$ are N- and $(p+1)$-dimenional vectors, respectively. Further, $\mathbf{X} = \{x_{ij}\}$ (with $x_{i1} = 1$) is an $N \times (p+1)$-dimensional matrix.

(c) The least squares estimator for $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

(d) Let $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^T$. Then the $\hat{\beta}_j$'s are normal distributed and unbiased, and

$$\mathrm{Var}(\hat{\beta}_j) = \sigma^2 c_{jj} \quad \text{and} \quad \mathrm{Cov}(\hat{\beta}_j, \hat{\beta}_l) = \sigma^2 c_{jl}$$

where c_{jl} is element (j, l) in the $(p+1) \times (p+1)$ matrix $\mathbf{C} = (\mathbf{X}^T\mathbf{X})^{-1}$.

(e) Let $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ik}$, and $\mathrm{RSS} = \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2$. Then $\hat{\sigma}^2 = \frac{\mathrm{RSS}}{n-(p+1)}$ is an unbiased estimator for σ^2, and $[N - (p+1)]\hat{\sigma}^2/\sigma^2 \sim \chi^2_{N-(p+1)}$. Further, $\hat{\sigma}^2$ and $\hat{\boldsymbol{\beta}}$ are independent.

(f) Let $\mathrm{SE}(\hat{\beta}_j)^2$ be the variance estimator for $\hat{\beta}_j$ that we get by replacing σ^2 with $\hat{\sigma}^2$ in the formulae for $\mathrm{Var}(\hat{\beta}_j)$ in point (d). Then $(\hat{\beta}_j - \beta_j)/\mathrm{SE}(\hat{\beta}_j) \sim t_{N-(p+1)}$.

(g) We can test the hypothesis

$$H_0 : \beta_{i_1} = \beta_{i_2} = \cdots = \beta_{i_q} = 0$$

by using the test observator

$$\mathrm{F} = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/q}{\mathrm{RSS}_1/(N-p-1)} \overset{H_0}{\sim} F_{q,N-p-1}$$

where $\mathrm{RSS}_0 = \sum_{i=1}^{N}(y - \hat{y}_i)^2$ when $\hat{y}_i$ is calculated *under* H_0 while RSS_1 is similarly under the full model.

(h) Ridge/Lasso regression: Minimize with respect to $\boldsymbol{\beta}$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q$$

with $q = 2$ for Ridge, $q = 1$ for Lasso.

(*i*) Best subset selection: Minimize with respect to $\boldsymbol{\beta}$

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

subject to that at most k of the β_j's are non-zero.

Usually some sub-optimal solutions (e.g. forward/backward selection) is applied.

3 General regression methods

(*a*) General setting: Assume

$$Y_i = f(\boldsymbol{x}_i)+\varepsilon_i$$

where $f(\cdot)$ is some general function while $\varepsilon_i, i=1,...,N$ are noise terms assumed to have zero expectation and variance σ^2.

(*b*) The K-nearest neighbor regression method is defined by

$$\hat{f}(\mathbf{x}_0)=\frac{1}{K}\sum_{\mathbf{x}_i\in\mathcal{N}_k(\boldsymbol{x}_0)} y_i$$

where $\mathcal{N}_k(\boldsymbol{x}_0)\subset\{\mathbf{x}_1,...,\mathbf{x}_N\}$ contain the K nearest points to $\mathbf{x}_0$ in the training set.

(*c*) Basis expansions: $f(\boldsymbol{x})=\sum_{m=1}^{M}\beta_m h_m(\boldsymbol{x})$

(*i*) Cubic spline: Piecewise polyomial with basis functions

$$\begin{aligned} h_1(x) =& 1, \quad h_2(x)=x, \quad h_3(x)=x^2, \quad h_4(x)=x^3,\\ h_{3+k}(x) =& (x-c_k)_+^3, \quad k=1,...,K. \end{aligned}$$

(*ii*) Natural cubic splines, smoothing splines.

(*iii*) Additive models: $f(\boldsymbol{x})=\sum_{j=1}^{p} f_j(x_j)$.

(*d*) Kernel methods/Local polynomial regression

$$\min_{\alpha(x_0),\beta_j(x_0)}\sum_{i=1}^{N}K_\lambda(x_0,x_i)[y_i-\alpha(x_0)-\sum_{j=1}^{d}\beta_j(x_0)x_i^j]^2.$$

(*e*) Neural network (one hidden layer):

$$f(\boldsymbol{x}_i)=f_1(\beta_0+\sum_{j=1}^{J}\beta_j f_0(\sum_{h=1}^{p}\alpha_{hj}x_{ih}))$$

with f_0, f_1 some chosen (nonlinear) activation function.

(*f*) Tree-based methods: $f(\boldsymbol{x}) = \sum_{m=1}^{M} c_m I(\boldsymbol{x} \in R_m)$ where $\mathcal{R}^p = R_1 \cup R_2 \cup \cdots \cup R_M$ and regions are defined through sequential splitting based on one variable at a time.

(*i*) Bagging, random forrest, Boosting:

$$\hat{f}_{\text{avg}}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(\mathbf{x})$$

where $\hat{f}^1(\mathbf{x}), \hat{f}^2(\mathbf{x}), ..., \hat{f}^B(\mathbf{x})$ are B different predictors based on ordinary bootstrapping (bagging) or where splitting are only considered by a subset of explanatory variables (random forrest). For boosting, the $\hat{f}^b$'s are estimated sequentially.

4 Some methods for Classification

(*a*) Logistic regression for binary responses:

$$\Pr(G = 1|\boldsymbol{x}) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)} \Leftrightarrow \log \frac{\Pr(G = 1|\boldsymbol{x})}{\Pr(G = 0|\boldsymbol{x})} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j.$$

Can be combined with Ridge, Lasso, subset selection procedures.

(*b*) Several classes:

$$\log \frac{\Pr(G = k|\boldsymbol{x})}{\Pr(G = K|\boldsymbol{x})} = \beta_{k0} + \sum_{j=1}^{p} \beta_{kj} x_j.$$

(*c*) In general: Want to estimate

$$\log \frac{\Pr(G = k|\boldsymbol{x})}{\Pr(G = K|\boldsymbol{x})} = f_k(\boldsymbol{x})$$

or more directly

$$\Pr(G = k|\boldsymbol{x}) = E[I(G = k)|\boldsymbol{x}]$$

Can be constructed with same techniques as for regression.

(*i*) K-nearest neighbor classification:

$$\Pr(G = g|\mathbf{X} = \mathbf{x}_0) \approx \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_0(\boldsymbol{x}_0)} I(g_i = g).$$

(*ii*) Generalized additive models:

$$\log \frac{\Pr(G=k|\boldsymbol{x})}{\Pr(G=K|\boldsymbol{x})} = \beta_0 + \sum_{j=1}^{p} f_j(x_j).$$

Criterion:

$$\text{PRSS}(\alpha, f_1, f_2, ..., f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j$$

(*d*) Alternative methods:

(*i*) Use Bayes theorem and model $f(\boldsymbol{x}|G=k) = f_k(\boldsymbol{x})$:

$$\Pr(G=k|X=x) = \frac{\pi_k f_k(\boldsymbol{x})}{\sum_{l=1}^{K} \pi_l f_l(\boldsymbol{x})}.$$

i. LDA: $f_k(\mathbf{x}) = p(x|G=k) = N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$.

ii. QDA: $f_k(\mathbf{x}) = p(x|G=k) = N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

(*ii*) Separating hyperplanes (2 classes): Boundary defined by $\{\boldsymbol{x} : \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x} = 0\}$

i. Optimal separating hyperplanes: Define output $y_i \in \{-1, 1\}$,

$$\max_{\beta, \beta_0, \|\beta\|=1} M : \text{s. t. } y_i \left(\boldsymbol{x}_i^T \beta + \beta_0 \right) \geq M, \quad i = 1, \ldots, N$$

ii. Rosenblatt's perceptron learning algorithm

5 The maximum likelihood method

(*a*) Maximum likelihood principle:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\text{argmax}}\, \ell(\boldsymbol{\theta}), \quad \ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}).$$

(*b*) Typically found as the solution of $\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \mathbf{0}$.

(*c*) Newton-Raphson

$$\boldsymbol{\theta}^{(s+1)} = \boldsymbol{\theta}^{(s)} - \left[\frac{\partial^2 \ell(\boldsymbol{\theta}^{(s)})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \right]^{-1} \frac{\partial \ell(\boldsymbol{\theta}^{(s)})}{\partial \boldsymbol{\theta}}$$

(*d*) Under certain regularity conditions, $\hat{\boldsymbol{\theta}} \approx N(\boldsymbol{\theta}, J(\hat{\boldsymbol{\theta}})^{-1})$ with

$$J(\hat{\boldsymbol{\theta}}) = - \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$$