

732G12 Data Mining

Föreläsning 2

Johan Alenlöv

IDA, Linköping University, Sweden

- Modellval
- Generaliserade linjära modeller
- Modellval för linjär regression

- Vi söker en modell som **generaliserar** väl.
 - Med generalisering menas att den ska fungera bra på ny data.
- En komplex modell har lättare att överanpassa.
- Vad är "komplexitet"?
 - Linjär modell: Antal variabler, interaktioner, transformationer etc.
 - Neurala nätverk: Bredd och djup av modellen.
 - Trädmodeller: Djupet.

Regularisering är ett sätt att motverka överanpassning.

- Idé att hindra modellen att bli för komplex.
- Ger förhoppningsvis bättre generaliseringsfel.
- **Mycket viktigt tema inom maskininlärning**
- Görs på olika sätt för olika metoder.

Idé är att

Komplex modell + regularisering = en bra modell

Kommer prata mer om detta senare.

Två klassiska problem är regression och klassificering.

- Regression: Prediktera en variabel y , oftast y kontinuerlig. Bruset ε är:
 - Vanligast är normalfördelat.
 - Alternativt, t-fördelning, Gamma, Log-normal,...
 - Kan också vara diskret, Poisson, Negativ binomial,...
- Fördelningen ger felfunktionen.
- Klassificering: y är kategorisk med 2 eller flera utfall:
 - Binär: logistisk/probit regression
 - Fler klasser: Multinomial logistisk/probit regression
 - Kommer diskutera fler metoder senare
- Hur skapa felfunktion?

Förväxlingsmatris

		Predikterad klass	
		Class = 1	Class = 0
Sann klass	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

- Precision:

$$P = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

- Felkvot (error rate):

$$E = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Förväxlingsmatris

		Predikterad klass	
		Class = 1	Class = 0
Sann klass	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

- Sensitivitet (recall, hit rate, true positive rate):

$$\text{TPR} = \frac{f_{11}}{f_{11} + f_{01}}.$$

- Specifitet (selectivity, true negative rate):

$$\text{TNR} = \frac{f_{00}}{f_{10} + f_{00}}.$$

- F-score är ett mått av träffsäkerhet baserat på precision och sensitivitet.

$$F_{\beta} = (1 + \beta^2) \frac{P \cdot TPR}{\beta^2 \cdot P + TPR}$$

- Vanligt med $\beta = 1$ vilket ger (harmoniska medelvärde)

$$F_1 = 2 \frac{P \cdot TPR}{P + TPR}$$

- F-score är mellan 0 och 1, högre är bättre.
- β säger hur mycket du värderar precision och sensitivitet.
- Beräknas klassvis.

Generaliserade linjära modeller

Antag att data $\mathbf{y} = y_1, y_2, \dots, y_n$ är oberoende observationer från sannolikhetsfördelning från exponentialfamiljen.

Vi har en linjär prediktor $\mathbf{X}\beta$ samt en länkfunktion g som kopplar ihop prediktorn med medelvärdet μ genom

$$g(\mu) = \mathbf{X}\beta.$$

Tar "vanlig" linjär regression till andra typer av responsvariabler

- Kontinuerlig: Normal, t, gamma, log-normal
- Binär: Logistisk regression
- Nominell: Multinomiell logistisk regression
- Frekvensdata: Poisson regression

Linjär regression

- Likelihood: Normal
- Länkfunktion: identitetsfunktionen
- SKattas genom att minimera:

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

Logistisk regression

- Likelihood: Bernoulli (y kan vara 0 eller 1, $\mathbb{P}(y = 1) = p$)
- Länkfunktion: Logit,

$$g(\mu) = \log \left(\frac{\mu}{1 - \mu} \right)$$

- Skattas med att maximera MLE

Multinominell logistisk regression

- Likelihood: Kategorisk (y kan ta K olika värden, $\mathbb{P}(y = k) = p_k$)
- Länkfunktion: Logit,

$$g(\mu) = \log \left(\frac{\mu}{1 - \mu} \right)$$

- Skattas med att maximera MLE

Modellval för linjära regression

Utgå ifrån: y kontinuerlig med normal likelihood.

Vi har ett antal förklarande variabler $\mathbf{x} = (x_1, \dots, x_p)$. Vill hitta parametrar/modell som ger minst generaliseringsfel.

Två alternativ:

- Alternativ 1: Välj ut en delmängd av variablerna.
 - Best subset, forward selection, backward selection
- Alternativ 2: Behåll alla variabler men begränsa parametrarna.
 - Regularisering, Ridge och Lasso.

Om vi har flera modeller, hur jämför vi dessa?

Två alternativ:

- Indirekt skatta testfelet
 - Utgå från träningsmängden
 - Försök att minska den bias som uppstår när vi inte använder all data.
- Direkt skatta testfelet
 - Valideringsdata
 - Korsvalidering

Indirekt skatta testfelet

MSE på träningsdatan underskattar "riktiga" MSE värdet, kan inte användas för att välja modell.

Idé: justera träningsfelet för att ta hänsyn till detta.

$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2),$$

Litet C_p är bäst.

$$\text{adjusted } R^2 = 1 - \frac{\text{RSS} / (n - d - 1)}{\text{TSS} / (n - 1)},$$

där

$$\text{TSS} = \sum_{i=1}^n (y_i - \bar{y})^2.$$

Stort adjusted R^2 är bäst.

Tre andra mått AIC, BIC och HQIC är baserade på MLE skattning av modeller.

Låt $\log(\hat{L})$ vara log-likelihood för optimala parametervärden.

$$\text{AIC} = 2d - 2\log(\hat{L})$$

$$\text{BIC} = d \cdot \log(n) - 2\log(\hat{L})$$

$$\text{HQIC} = d \cdot \log(\log(n)) - 2\log(\hat{L})$$

Lågt värde är bättre.

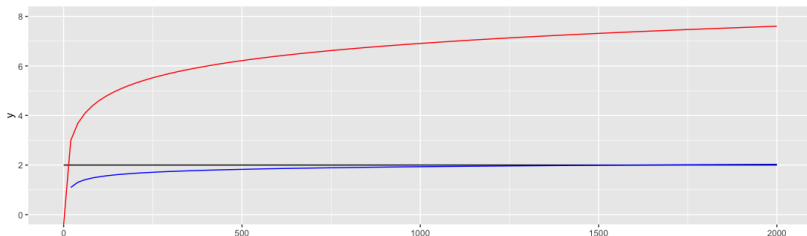
Linjär Regression

$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2)$$

$$\text{BIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + \log(n)d\hat{\sigma}^2)$$

$$\text{HQIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + \log(\log(n))d\hat{\sigma}^2)$$

För linjär regression är $C_p \propto \text{AIC}$.



Algorithm 6.1 *Best subset selection*

1. Let \mathcal{M}_0 denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
 2. For $k = 1, 2, \dots, p$:
 - (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
 - (b) Pick the best among these $\binom{p}{k}$ models, and call it \mathcal{M}_k . Here *best* is defined as having the smallest RSS, or equivalently largest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Bild från kursboken "An Introduction to Statistical Learning with Applications in R".

Algorithm 6.2 *Forward stepwise selection*

1. Let \mathcal{M}_0 denote the *null* model, which contains no predictors.
 2. For $k = 0, \dots, p - 1$:
 - (a) Consider all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - (b) Choose the *best* among these $p - k$ models, and call it \mathcal{M}_{k+1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Bild från kursboken "An Introduction to Statistical Learning with Applications in R".

Algorithm 6.3 *Backward stepwise selection*

1. Let \mathcal{M}_p denote the *full* model, which contains all p predictors.
 2. For $k = p, p - 1, \dots, 1$:
 - (a) Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k - 1$ predictors.
 - (b) Choose the *best* among these k models, and call it \mathcal{M}_{k-1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Bild från kursboken "An Introduction to Statistical Learning with Applications in R".

Krympning (Shrinkage)

Idé: begränsa hur stora parametrarna får vara.

- Straffa stora parametrar
- Ändra deras värdemängd

Vanligaste metoderna:

- Ridge: l^2 -norm
- Lasso: l^1 -norm

Kom ihåg att standardisera era förklarande variabler först innan krympning!

Ridge Regression

I vanliga regression minimerar vi

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

I Ridge lägger vi till l^2 -norm på β vilket ger

$$f_{\text{Ridge}}(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2, \quad \lambda \geq 0.$$

λ är en **hyperparameter** som vi behöver sätta.

Notera att β_0 inte påverkas.

Ridge Regression

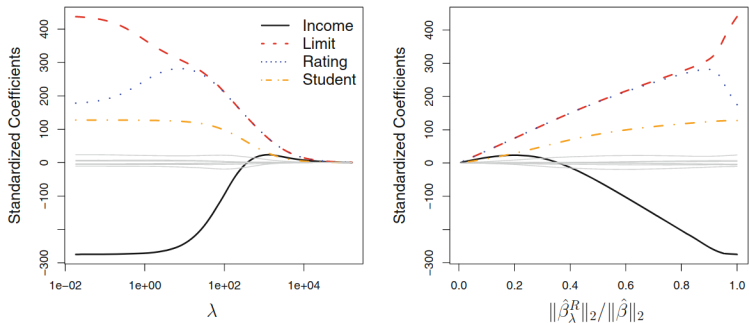


FIGURE 6.4. The standardized ridge regression coefficients are displayed for the **Credit** data set, as a function of λ and $\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$.

I vanliga regression minimerar vi

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

I Ridge lägger vi till l^1 -norm på β vilket ger

$$f_{\text{Ridge}}(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|, \quad \lambda \geq 0.$$

λ är en **hyperparameter** som vi behöver sätta.

Notera att β_0 inte påverkas.

Lasso Regression

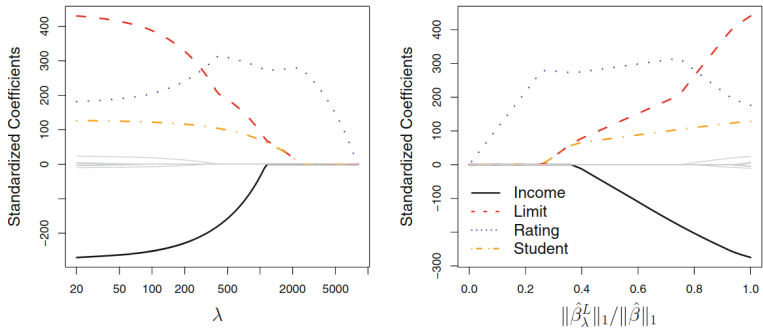


FIGURE 6.6. The standardized lasso coefficients on the **Credit** data set are shown as a function of λ and $\|\hat{\beta}_\lambda^L\|_1 / \|\hat{\beta}\|_1$.

Ridge vs Lasso

	Ridge	Lasso
$\lambda \rightarrow 0$	$\hat{\beta}_{\text{Ridge}} \rightarrow \hat{\beta}_{\text{OLS}}$	$\hat{\beta}_{\text{Lasso}} \rightarrow \hat{\beta}_{\text{OLS}}$
$\lambda \rightarrow \infty$	$\hat{\beta}_{\text{Ridge}} \rightarrow 0$	$\hat{\beta}_{\text{Lasso}} \rightarrow 0$
Norm	l^2 -norm	l^1 -norm
Område	Hypersfär	Polytop
Parametrar	Krymper och ger många små	Sätter många till 0
Korrelerade variable	Krymper alla lika mycket	Sätter en till 0

Ridge vs Lasso

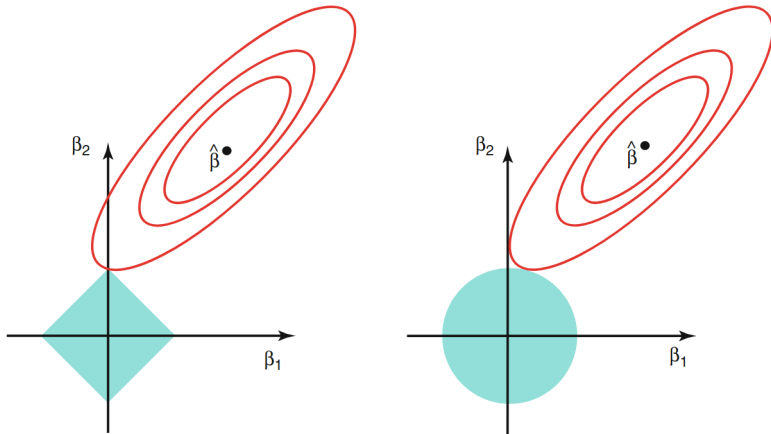


FIGURE 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Ridge vs Lasso

- I R, skattas enkelt med `glmnet()` eller `cv.glmnet()`.
- Vilken som är bäst beror på kontext.
- Ridge passar bra när y beror på de flesta variablerna.
 - Många variabler och ungefär samma effektstorlek.
- Lasso passar bra när y beror på bara några få variabler.
 - Fåtal variabler med hög effektstorlek, resten nära 0.
- Lasso kan vara lättare att tolka.
- Lättare att göra inference på parameterarna i Ridge.
- Vilken man ska välja får ofta avgöras empiriskt för specifika dataset.
- Finns många utökningar.

- Låt varje parameter β_j ha sin egen vikt w_j

$$f(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p w_j |\beta_j|, \quad \lambda \geq 0.$$

- Välj vikter

$$w_j = |\hat{\beta}_{j,\text{start}}|^{-\gamma}$$

- Ofta väljs $\gamma = 1$.
- Behöver startvärde $\hat{\beta}_{j,\text{start}}$, kan använda OLS eller Ridge skattning.
- Stora startvärden ger små vikter vilket gör att parametern krymper mindre.
- Finns vissa teoretiska fördelar, men kräver extra skattning.

Elasticnet regression

- Kombinerar Ridge och Lasso.
- Ny hyperparameter α för att mixa dessa,

$$f(\beta) = \text{RSS} + \lambda \left((1 - \alpha) \sum_{j=1}^p \beta_j^2 + \alpha \sum_{j=1}^p |\beta_j| \right), \quad \lambda \geq 0, 0 \leq \alpha \leq 1.$$

- λ är likt tidigare hur mycket regularisering vi gör.
- α väljer hur mycket vikt vid Ridge respektive Lasso.
- Kan tvinga vissa β till 0, men inte lika många som Lasso.
- Klarar av korrelerade/grupper av variabler bättre än Lasso.
- Nackdel är att vi har två hyperparametrar.