

**COEN 281 -- Introduction to Pattern Recognition and
Data Mining**

Lecture 6: Regularized Regression

Instructor: Dr. Giovanni Seni
GSeni@scu.edu

***Department of Computer Engineering
Santa Clara University***

Fall/18

Syllabus

Week 1	Introduction; R (Ch.1)
Week 2	Bayesian Decision Theory (Ch.2; DHS: 2.1-2.6, 2.9) Parameter Estimation (DHS: 3.1-3.4)
Week 3	Linear Discriminant Functions (Ch.3&4; DHS: 3.8.2, 5.1-5.8) Regularization (Ch.6; SE: Ch.3)
Week 4	Neural Networks (DHS: 6.1-6.6, 6.8); Deep Learning
Week 5	Support Vector Machines (Ch.9)
Week 6	Decision Trees (Ch. 8.1; DHS: 8.3; Ch 2 SE)
Week 7	Ensemble Methods (Ch. 8.2; SE: Ch 4, 5)
Week 8	Clustering (Ch. 10; DHS: 10.6, 10.7) Clustering (DHS: 10.9); How many clusters are there? (DHS: 10.10)
Week 9	Non-metric: Association Rules Collaborative Filtering
Week 10	Text Retrieval; Other topics

Overview

- In a Nutshell
- Predictive Learning Review
- Model Complexity & Regularization
- Regularized Linear Regression
- Example – 1M predictors

Regularization Methods in a Nutshell

- Methods intended to reduce “variance” and produce “sparse” linear models
- Methods add a “complexity penalty” term to the criterion being minimized
 - Complexity term penalizes for the increased variance associated with more complex model
 - Various penalizing functions possible
- A form of “biased” learning
- Leads to optimization problems with inequality constraints

Predictive Learning

Procedure Summary

- Given "training" data $D = \{y_i, x_{i1}, x_{i2}, \dots, x_{in}\}_1^N = \{y_i, \mathbf{x}_i\}_1^N$
 - y_i, x_{ij} are measured values of attributes (properties, characteristics) of an object
 - y_i is the "response" (or output) variable
 - x_{ij} are the "predictor" (or input) variables
 - D is a random sample from some unknown (joint) distribution
- Build a functional model $\hat{y} = \hat{F}(x_1, x_2, \dots, x_n) = \hat{F}(\mathbf{x})$
 - Offers adequate and interpretable description of how the inputs affect the outputs

Predictive Learning

Procedure Summary (2)

- **Model**: underlying functional form sought from data
$$\hat{F}(\mathbf{x}) = \hat{F}(\mathbf{x}; \mathbf{a}) \in \mathcal{F} \quad \text{family of functions indexed by } \mathbf{a}$$
- **Score criterion**: judges (lack of) quality of fitted model
 - Loss function $L(y, \hat{F})$: penalizes individual errors in prediction
 - Risk $R(\mathbf{a}) = E_{y, \mathbf{x}} L(y, \hat{F}(\mathbf{x}; \mathbf{a}))$: the expected loss over all predictions
- **Search Strategy**: minimization procedure of score criterion
$$\mathbf{a}^* = \arg \min_{\mathbf{a}} R(\mathbf{a})$$

Predictive Learning

Procedure Summary (3)

- “Surrogate” Score criterion:

- Training data: $\{y_i, \mathbf{x}_i\}_1^N \sim p(\mathbf{x}, y)$
- $p(\mathbf{x}, y)$ unknown $\Rightarrow \mathbf{a}^*$ unknown

\Rightarrow Use approximation: Empirical Risk

- $\hat{R}(\mathbf{a}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{F}(\mathbf{x}_i; \mathbf{a})) \quad \Rightarrow \quad \hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \hat{R}(\mathbf{a})$

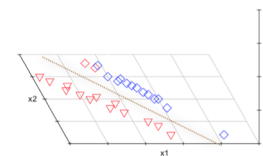
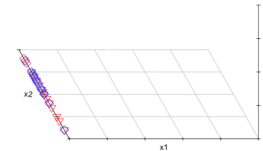
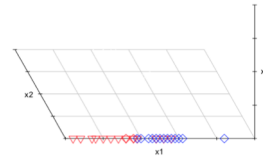
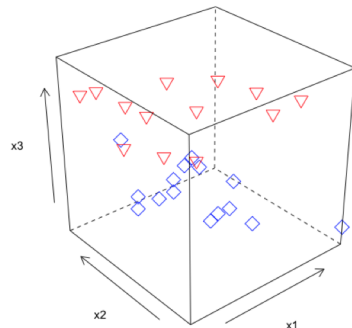
- If not $N \gg n$, $R(\hat{\mathbf{a}}) \gg R(\mathbf{a}^*)$

Overview

- In a Nutshell & Timeline
- Predictive Learning
- Model Complexity & Regularization
 - Right model “size”
 - Bias-Variance schematic
 - Regularization
- Regularized Linear Regression
- Path Finding by Generalized Gradient Descent
- Example – 1M predictors

Model Complexity

What is the “right” size of a linear model?



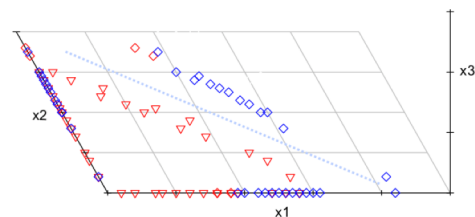
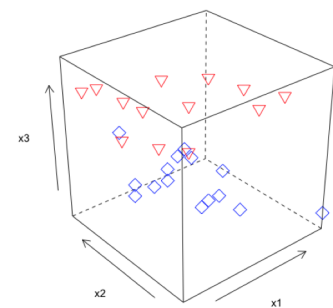
COEN281

GSeni@scu.edu

9

Model Complexity

What is the “right” size of a linear model?



- If model (# of variables) is too small, then approximation is too crude (**bias**) \Rightarrow increased errors
- If model is too large, then it fits the training data too closely (overfitting, increased **variance**) \Rightarrow increased errors

COEN281

GSeni@scu.edu

10

Model Complexity

Regularization In a Nutshell

- What is *regularization*?
 - “any part of model building which takes into account – implicitly or explicitly – the finiteness and imperfection of the data and the limited information in it, which we can term ‘variance’ in an abstract sense” [Rosset, 2003]
- Forms of regularization
 1. Explicit via [constraints on model complexity](#)
 2. Implicit through incremental building of the model
 - E.g., Stochastic Gradient Boosting
 3. Choice of robust loss functions

Overview

- In a Nutshell & Timeline
- Predictive Learning
- Model Complexity & Regularization
 - Regularized Linear Regression
 - Overview: “Constrained” vs. “Penalized” formulation
 - Coefficient Paths and Direct Path Seeking
 - Complexity Penalties
- Example – 1M predictors

Linear Regression

Overview

- Linear model: $F(\mathbf{x}) = a_0 + \sum_{j=1}^n a_j x_j$

- Standard coefficient estimation criterion (OLR):

$$\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \sum_{i=1}^N L(y_i, a_0 + \sum_{j=1}^n a_j x_{ij}) \quad \text{E.g., } \hat{\mathbf{a}} = (\mathbf{X}'\mathbf{X} + \epsilon \mathbf{I})^{-1} \mathbf{X}'\mathbf{y}$$

- OLR often unsatisfactory:
 - Prediction accuracy: high variance in coefficient estimates
 - Interpretation: desire for a smaller subset of predictors that exhibit the strongest effects
 - Subset Selection*: can be extremely variable because of its discrete process
 - Regularized Regression*: continuous process often preferred

COEN281

GSeni@scu.edu

14

Regularized Linear Regression

Overview

- Augmented coefficients estimation criterion:

$$\hat{\mathbf{a}} = \{\hat{a}_j\}_0^n = \arg \min_{\{a_j\}} \sum_{i=1}^N L(y_i, a_0 + \sum_{j=1}^n a_j x_{ij}) \quad \text{s.t. } P(\mathbf{a}) \leq t$$

- “Constraining” function $P(\mathbf{a})$:
 - Non-negative
 - $0 < t < P(\hat{\mathbf{a}})$: bias-variance tradeoff
 - Deterministic and independent of the particular random sample
 - \Rightarrow provides a stabilizing influence on the criterion being minimized
 - Best $P(\mathbf{a})$ requires knowledge of \mathbf{a}^*
 - Use $\mathbf{a} \approx \mathbf{a}^* \Rightarrow \text{sparcity}(\mathbf{a}) \approx \text{sparcity}(\mathbf{a}^*)$

COEN281

GSeni@scu.edu

15

Regularized Linear Regression

Penalized Formulation

- Equivalent penalized formulation:

$$\hat{\mathbf{a}} = \{\hat{a}_j\}_0^n = \arg \min_{\{a_j\}} \sum_{i=1}^N L(y_i, a_0 + \sum_{j=1}^n a_j x_{ij}) + \lambda \cdot P(\mathbf{a}) \quad (1)$$

- $P(\mathbf{a})$ penalizes for the increased variance associated with more complex model
- $\infty \geq \lambda \geq 0 \sim 0 < t < P(\hat{\mathbf{a}})$
- Coefficient “paths” $\hat{\mathbf{a}}(\lambda)$:
 - For each value of λ , we have a different solution to (1)
 - $\lambda = 0 \Rightarrow$ OLR solution
 - $\lambda = \infty \Rightarrow \{\hat{a}_j\}_1^n = 0; \hat{a}_0 = \arg \min_{\{a\}} \sum_{i=1}^N L(y_i, a)$

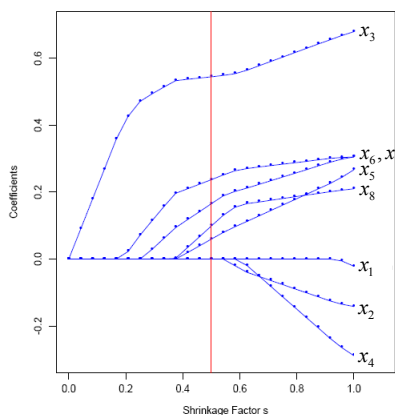
COEN281

GSeni@scu.edu

16

Regularized Linear Regression

Coefficient Paths



- Shrinkage factor: $s = 1/\lambda$

COEN281

GSeni@scu.edu

17

Regularized Linear Regression

Model Selection

- Given $L(y, \hat{y})$ and $P(\mathbf{a})$:

$$\hat{\lambda} = \arg \min_{0 \leq \lambda \leq \infty} \tilde{R}(\hat{\mathbf{a}}(\lambda)) = \arg \min_{\mathbf{a}} [\hat{R}(\mathbf{a}) + \lambda \cdot P(\mathbf{a})]$$

- Selected model: $\hat{\mathbf{a}}(\hat{\lambda})$
- Cross-validation often used on a predefined grid in $[\lambda_{\min}, \lambda_{\max}]$
- Challenge: rapidly produce paths without repeatedly optimizing

⇒ Direct Path Seeking algorithms

COEN281

GSeni@scu.edu

18

Regularized Linear Regression

Complexity Penalties

- Ridge: $P(\mathbf{a}) = \sum_{j=1}^n a_j^2$
 - Shrinks coefficients towards 0
 - “Dense” solutions
 - Best for *large number of small effects*
 - k identical predictors ⇒ each gets identical coefficient 1/kth the size



- Lasso: $P(\mathbf{a}) = \sum_{j=1}^n |a_j|$
 - “Sparse” solutions – i.e., does variable selection
 - Best for *small to moderate number of moderate-size effects*
 - Somewhat indifferent to very correlated predictors; will tend to pick one and ignore the rest

... up to a limit: extreme correlations cause instability

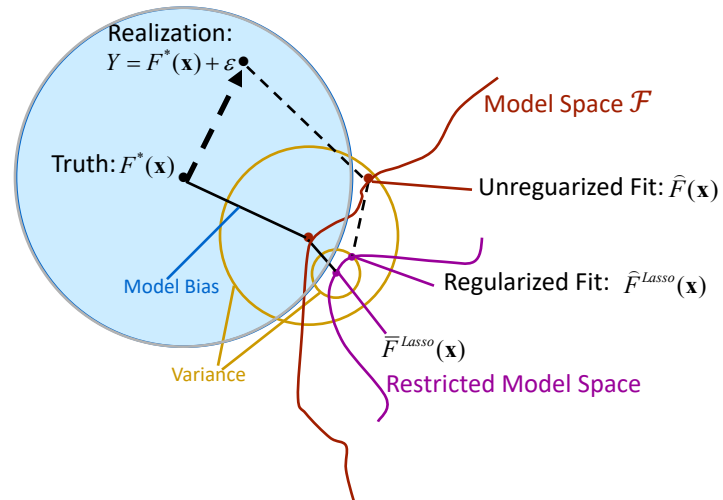
COEN281

GSeni@scu.edu

19

Regularized Regression

Regularized Bias-Variance Schematic



COEN281

GSeni@scu.edu

Adapted from ESL

20

Overview

- In a Nutshell & Timeline
 - Predictive Learning
 - Model Complexity & Regularization
 - Regularized Linear Regression
- Example – 1M predictors

COEN281

GSeni@scu.edu

21

Example

~1M Predictors

- Document classification task
 - “Bag of words” representation
 - Feature vector for each document is very sparse
 - R session:

```
load("Data/NewsGroup.RData")
attributes(NewsGroup)
[1] "x" "y"

x <- NewsGroup$x; dim(x)
[1] 11314 777811

y <- NewsGroup$y; length(y)
[1] 11314

summary(as.factor(y))
  -1    1
5420 5894
```

Example

~1M Predictors (2)

```
fit.news <- glmnet(x, y, family="binomial")

attributes(fit.news)
[1] "a0"      "beta"    "dev"     "nulldev" "df"      "lambda"
[7] "npasses" "jerr"    "dim"     "call"

length(fit.news$lambda)
[1] 100
dim(fit.news$beta)
[1] 777811 100

beta = coef(fit.news, s = 0.01)
class(beta); length(beta)
[1] "dgCMatrix"
[1] 777812

beta.all <- beta[,1]
length(which(abs(beta.all) > 0))
[1] 846
```

Example

~1M Predictors (3)

```
system.time(fit.news <- cv.glmnet(x, y, family = "binomial"))
      user system elapsed
52.137  11.623   64.564

attributes(fit.news)
[1] "lambda"      "cvm"          "cvstd"        "cvup"         "cvlo"
[6] "nzero"       "name"         "lambda.min"   "lambda.1se"

##... re-fit with best lambda
fit2.news <- glmnet(x, y, family = "binomial",
                    lambda = fit.news$lambda.1se)
beta = coef(fit2.news, s = fit.news$lambda.1se)
beta.all <- beta[,1]
length(which(abs(beta.all) > 0))
[1] 4986

yHat <- predict(fit2.news, x, s = fit.news$lambda.1se, type='class')[,1]
table(y, yHat)
      yHat
y      1      2
-1 5142  278
 1    61 5833
```