



# BIOST 546: Machine Learning for Biomedical Big Data

Ali Shojaie

Lecture 7: Penalized Regression for Other Data Types  
Spring 2017

# Recap

- Tree-based methods
- Bootstrap
- Bagging, boosting and random forests

# Today's Class

- Penalized regression in general form
  - ▶ other data types
  - ▶ other penalties

# Data of Different Types

- Simple continuous response
- Binary response
- Count data
- Survival outcome

# Different Data Need Different Models

- Simple continuous response: **squared error**
- Binary response: **logistic loss** (0-1/hinge loss for other methods)
- Count data: **Poisson loss**
- Survival outcome: **Cox loss**

# Log-Likelihood Loss

Data generating mechanisms  $\rightarrow$  (log)likelihood  $\rightarrow$  Loss function

# Log-Likelihood Loss

Our usual Gaussian model

$$y_i = \beta_0 + x_i^\top \beta + \varepsilon_i$$

with  $\varepsilon_i$  iid  $N(0, \sigma^2)$

The likelihood:

$$\mathcal{L}(\beta \mid x, y) = (2\pi\sigma^2)^{n/2} \exp - \frac{1}{2\sigma^2} \sum (y_i - x_i^\top \beta)^2$$

# Log-Likelihood Loss

Our usual Gaussian model

$$y_i = \beta_0 + x_i^\top \beta + \varepsilon_i$$

with  $\varepsilon_i$  iid  $N(0, \sigma^2)$

The likelihood:

$$\mathcal{L}(\beta \mid x, y) = (2\pi\sigma^2)^{n/2} \exp - \frac{1}{2\sigma^2} \sum (y_i - x_i^\top \beta)^2$$

Equivalent to:

$$\min \sum (y_i - x_i^\top \beta)^2$$



# Log-Likelihood Loss

Our usual Gaussian model

$$y_i = \beta_0 + x_i^\top \beta + \varepsilon_i$$

with  $\varepsilon_i$  iid  $N(0, \sigma^2)$

The likelihood:

$$\mathcal{L}(\beta \mid x, y) = (2\pi\sigma^2)^{n/2} \exp - \frac{1}{2\sigma^2} \sum (y_i - x_i^\top \beta)^2$$

Equivalent to:

$$\min \sum (y_i - x_i^\top \beta)^2$$

our usual **least squares** criterion!

# Log-Likelihood Loss

Logistic model

$$\log \left( \frac{p_i}{1 - p_i} \right) = \beta_0 + x_i^\top \beta$$

with  $p_i = P(Y_i = 1 \mid x_i)$

The likelihood:

$$\begin{aligned} \mathcal{L}(\beta \mid x, y) &= \prod_i p_i^{y_i} (1 - p_i)^{(1 - y_i)} \\ &= \prod_i \text{expit}(\beta_0 + x_i^\top \beta)^{y_i} \left( 1 - \text{expit}(\beta_0 + x_i^\top \beta) \right)^{(1 - y_i)} \end{aligned}$$

# Log-Likelihood Loss

Logistic model

$$\log \left( \frac{p_i}{1 - p_i} \right) = \beta_0 + x_i^\top \beta$$

with  $p_i = P(Y_i = 1 \mid x_i)$

The likelihood:

$$\begin{aligned} \mathcal{L}(\beta \mid x, y) &= \prod_i p_i^{y_i} (1 - p_i)^{(1 - y_i)} \\ &= \prod_i \text{expit}(\beta_0 + x_i^\top \beta)^{y_i} \left( 1 - \text{expit}(\beta_0 + x_i^\top \beta) \right)^{(1 - y_i)} \end{aligned}$$

Equivalent to:

$$\min \sum \left( -y_i x_i^\top \beta + \log \left( 1 + e^{x_i^\top \beta} \right) \right)$$

# Log-Likelihood Loss

Logistic model

$$\log \left( \frac{p_i}{1 - p_i} \right) = \beta_0 + x_i^\top \beta$$

with  $p_i = P(Y_i = 1 \mid x_i)$

The likelihood:

$$\begin{aligned} \mathcal{L}(\beta \mid x, y) &= \prod_i p_i^{y_i} (1 - p_i)^{(1 - y_i)} \\ &= \prod_i \text{expit}(\beta_0 + x_i^\top \beta)^{y_i} \left( 1 - \text{expit}(\beta_0 + x_i^\top \beta) \right)^{(1 - y_i)} \end{aligned}$$

Equivalent to:

$$\min \sum \left( -y_i x_i^\top \beta + \log \left( 1 + e^{x_i^\top \beta} \right) \right)$$

which is solved in **logistic regression**.

# Log-Likelihood Loss

Other examples:

# Log-Likelihood Loss

Other examples:

- **Poisson Model:**

$$\log(E[y_i | x_i]) = \beta_0 + \beta^\top x_i$$

is used to model **rare events**

- ▶ deaths from TB each year in the US
- ▶ counts from sequencing data for gene expression
- ▶ limit of Binomial likelihood for a large number of trials with a really biased coin (e.g.  $\pi = 3/1000$ )

give rise to **Poisson regression**

# Penalized likelihood

**Q:** What do we do if  $p > n$  in e.g. Poisson regression?

# Penalized likelihood

**Q:** What do we do if  $p > n$  in e.g. Poisson regression?

**A:** The idea is the same – need to **control the model complexity!!**



# Penalized likelihood

**Q:** What do we do if  $p > n$  in e.g. Poisson regression?

**A:** The idea is the same – need to **control the model complexity!!**

- Can use e.g. **penalties**, as in penalized logistic regression!
- The general formulation is:

$$\min \ell(\beta) \rightarrow \min \ell(\beta) + \lambda \|\beta\|_1$$

# Penalized likelihood

**Q:** What do we do if  $p > n$  in e.g. Poisson regression?

**A:** The idea is the same – need to **control the model complexity**!!

- Can use e.g. **penalties**, as in penalized logistic regression!
- The general formulation is:

$$\min \ell(\beta) \rightarrow \min \ell(\beta) + \lambda \|\beta\|_1$$

- For Poisson regression: `glmnet(x, y, family = "Poisson")`

# Log-Likelihood Loss

Other examples:

# Log-Likelihood Loss

Other examples:

- **Cox Model** (nested multinomials):

- ▶  $x_i$ : features
- ▶  $y_i$ : time on study
- ▶  $z_i$ : indicator of fail/censoring

Consider likelihood conditional on failure times:

$$P(\text{person } j \text{ fails at time } t \mid \text{a failure at time } t) = \frac{e^{x_j^\top \beta}}{\sum_{k \text{ at risk at } t} e^{x_k^\top \beta}}$$

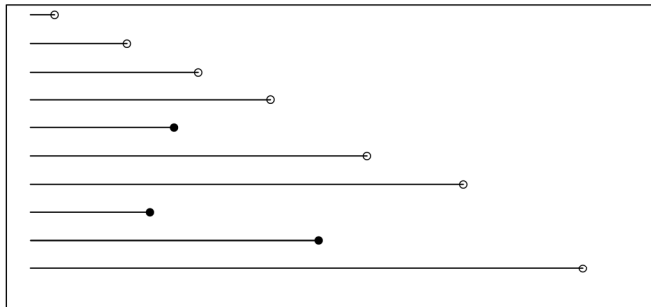
# Survival Outcome

# Survival Outcome

We're interested in **length** of survival time...

# Survival Outcome

We're interested in **length** of survival time... but not everyone dies;

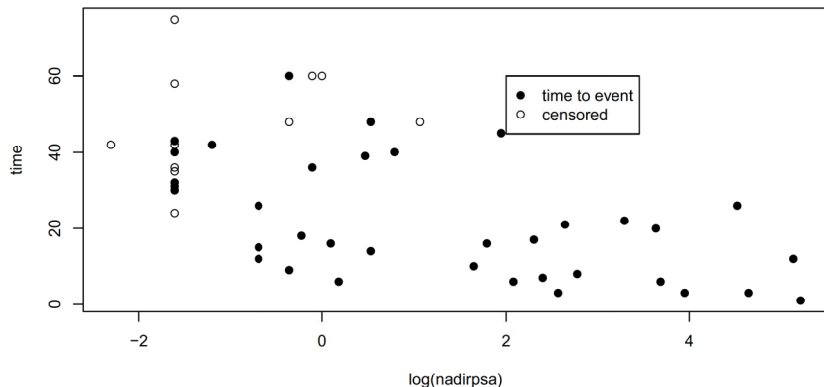


Survival time (all start at zero)

At random, we see survival time  $T$  **or** just know that  $T > C$

# Survival Outcome

Results are *somewhat* intuitive...



What do you think the effect of `nadirpsa` is?



# Surv objects

The 'outcome' in survival analysis involves both an observed time and a censoring status. These are packaged together in a `Surv` object

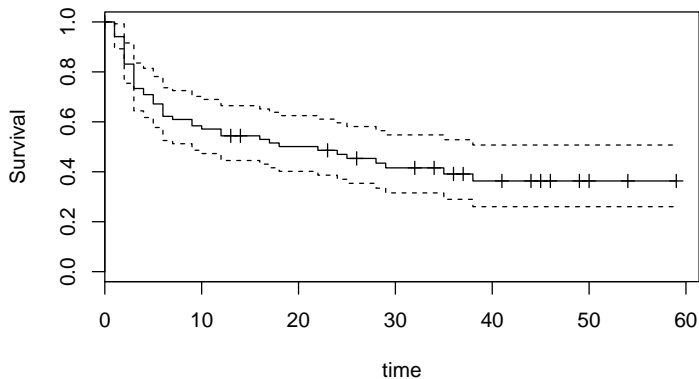
- `library(survival)` has many features for low-dimensional data
- `Surv(time, event)` is the simplest form (for simple right-censoring data)
- `event` tells R whether we saw  $T$  or just  $T > C$
- Full  $T, C$  terminology a bit cumbersome, censoring is instead shown with a  $+$

```
> library(survival)
Loading required package: splines
> tumor.surv <- with(tumor, Surv(time, event) )
> tumor.surv[1:10]
[1] 0+ 1+ 4+ 7+ 10+ 6 14+ 18+ 5 12
```

**Always check this!** Is your censoring setup correctly?

# Survival Curves

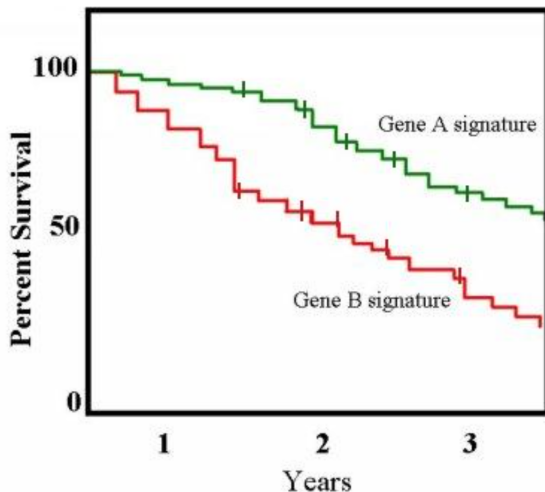
The most common, 'intuitive' summary, also known as **Kaplan-Meier** curves



```
plot(survfit(tumor.surv ~ 1))
```

# Survival Curves

Can e.g. compare different groups



Gene B signature < Gene A signature

# Building prognostic survival classifiers

- Generally want a classification of **high** vs **low** risk:
- Given a Cox-model
  - ▶ with  $p$  genomic features
  - ▶ and coefficient vector  $\beta$

We know observations with larger  $x_i^\top \beta$  are higher risk!

# Building prognostic survival classifiers

- Generally want a classification of **high** vs **low** risk:
- Given a Cox-model
  - ▶ with  $p$  genomic features
  - ▶ and coefficient vector  $\beta$

We know observations with larger  $x_i^\top \beta$  are higher risk!

Can choose a cutoff ( $c$ ), and classify observations with  $x_i^\top \beta \geq c$  high risk, otherwise low risk.

# Building prognostic survival classifiers

- Generally want a classification of **high** vs **low** risk:
- Given a Cox-model
  - ▶ with  $p$  genomic features
  - ▶ and coefficient vector  $\beta$

We know observations with larger  $x_i^\top \beta$  are higher risk!

Can choose a cutoff ( $c$ ), and classify observations with  $x_i^\top \beta \geq c$  high risk, otherwise low risk.

How do we choose  $c$ ?

# Building prognostic survival classifiers

- Generally want a classification of **high** vs **low** risk:
- Given a Cox-model
  - ▶ with  $p$  genomic features
  - ▶ and coefficient vector  $\beta$

We know observations with larger  $x_i^\top \beta$  are higher risk!

Can choose a cutoff ( $c$ ), and classify observations with  $x_i^\top \beta \geq c$  high risk, otherwise low risk.

How do we choose  $c$ ? cross-validation! (CV survival curves)

# Cox Regression in HD

**Q:** What if  $p > n$  in survival settings?



# Cox Regression in HD

**Q:** What if  $p > n$  in survival settings?

**A:** The answer is the same...

# Cox Regression in HD

**Q:** What if  $p > n$  in survival settings?

**A:** The answer is the same...

Can use regularization:

$$\min \ell(\beta) + \lambda \|\beta\|_1$$

# Example: Gene Expression Example

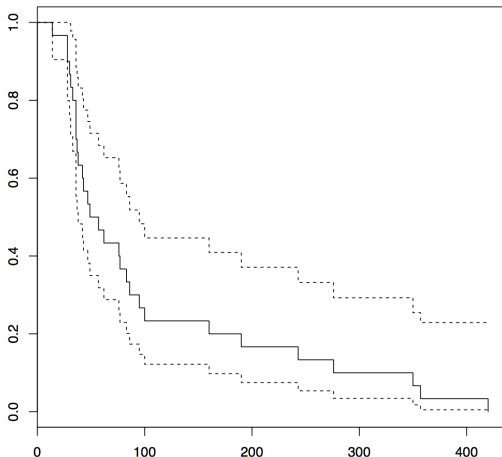
Building a prognostic classifier for patients with advanced bladder cancer receiving chemotherapy:

- GEO-GSE5287
- 30 patients
- 22283 gene expressions

# Example: Gene Expression Example

Building a prognostic classifier for patients with advanced bladder cancer receiving chemotherapy:

- GEO-GSE5287
- 30 patients
- 22283 gene expressions

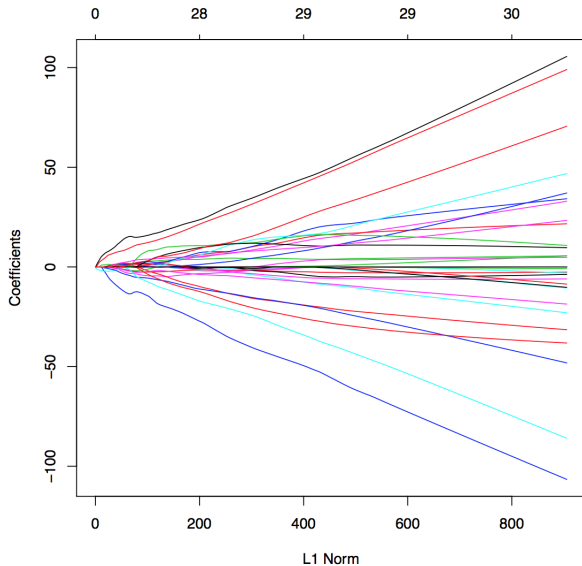


# Cox Regression in HD

Very easy to run lasso:

```
fit <- glmnet(X, Surv(time,status), family = "cox")  
plot(fit)
```

# Cox Regression in HD



# Cross validation for KM in HD

This is a bit more tricky, as we may need to cross validate for each pair of candidate  $c$  and  $\lambda$ :

# Cross validation for KM in HD

This is a bit more tricky, as we may need to cross validate for each pair of candidate  $c$  and  $\lambda$ :

1. Break data into folds
2. For each fold  $k$ 
  - 2.1. Train on all data except  $k$ th fold to find  $\hat{\beta}$
  - 2.2. Calculate **score**  $\eta_i = x_i^\top \hat{\beta}$  for all  $i$  in the left-out fold
3. Split the data into  $i$  with  $\eta_i \leq c$  and  $\eta_i > c$
4. Plot KM curves!

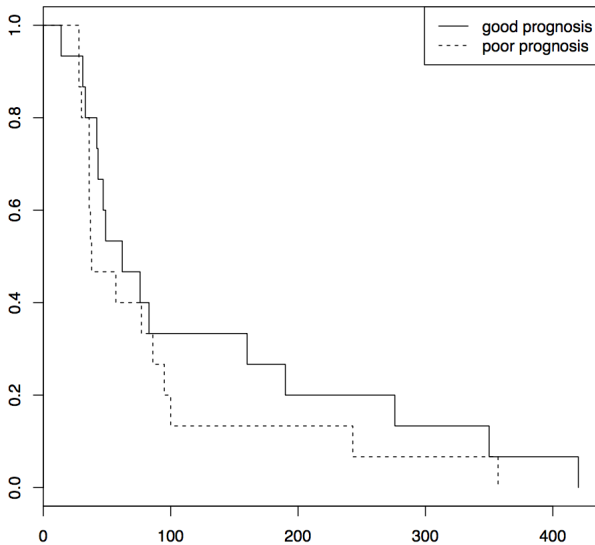
Choose the best KM plot!



```

> unord <- match(1:30,obs.ord)
> test.pred <- matrix(0,ncol = 100, nrow = 30)
> for(fold in 1:3){
+ ind.train <- obs.ord[((fold-1)*10 + 1):(fold*10)]
+ fit.train <- glmnet(X[ind.train,], Surv(time[ind.train],status[ind.train]), family="cox")
+ test.pred[-ind.train,] <- predict(fit.train, X[-ind.train,])
+}
> k <- 80
> plot(survfit(Surv(time,status)~(test.pred[,k] > median(test.pred[,k]))),
+ lty = c(1,2))
> legend("topright", c("good prognosis","poor prognosis"), lty = c(1,2))

```



Not so great!

# Log-Likelihood Recap

- Losses are often based on generative model or error structure ??
- Minimize Negative Log Likelihood
- Can add sparsity/ridge/other penalties

# Other Penalties

# Other Penalties

- Lasso/ridge are not the only sensible penalties

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- **Group sparsity:**
  - ▶ Categorical variables
  - ▶ Genes in the same pathway
  - ▶ Other groupings among variables

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- **Group sparsity:**
  - ▶ Categorical variables
  - ▶ Genes in the same pathway
  - ▶ Other groupings among variables

$$\min \ell(\beta) + \lambda \sum_k \|\beta^{(k)}\|_2$$

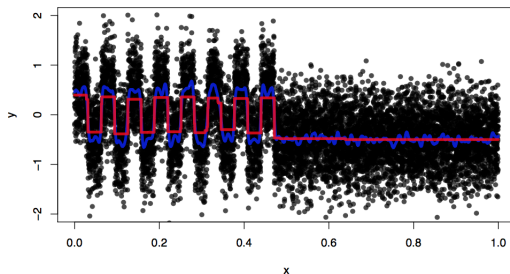


# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:

# Other Penalties

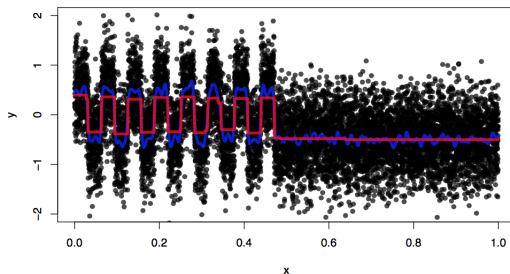
- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- **Fused sparsity:**
  - ▶ To encourage similarity among consecutive covariates



e.g. in the setting of DNA methylation data, or copy number variation data (CNV)

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- **Fused sparsity:**
  - ▶ To encourage similarity among consecutive covariates



e.g. in the setting of DNA methylation data, or copy number variation data (CNV)

$$\min \ell(\beta) + \lambda \sum_j |\beta_j - \beta_{j-1}|$$

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- There are various other types of penalties
  - ▶ Hierarchical sparsity
  - ▶ Smoothness
  - ▶ ...

# Other Penalties

- Lasso/ridge are not the only sensible penalties
- In some settings it makes sense to use other types of penalties:
- There are various other types of penalties
  - ▶ Hierarchical sparsity
  - ▶ Smoothness
  - ▶ ...
- This is a very active area of research!!