BITTIGER

DS 501 Data scientist express bootcamp

Week 3 [Ella]

BITTIGER

Summary

- Collinearity
  - Definition and impact
  - Regularization
    - Ridge (L2)
    - Lasso (L1)
    - Mixed
- Cross validation

Summary

- Logistic regression
  - Definition, sigmoid function
  - Relationship with linear regression
  - Decision boundary
  - Coefficients estimation
  - Gradient descent
  - Interpreting Coefficients
  - Regularization
  - Evaluation model performance
    - Confusion matrix
    - ROC curve

Collinearity

- What is collinearity, multicollinearity?
  - Highly correlated predictors, example.
  - Involve more than 2 predictors, multicollinearity
- Why is it a problem?
  - Having two predictors, both are parent's height, β_1, β_2 and $\beta_1 + \gamma, \beta_2 - \gamma$ gives same prediction
  - Increases the variance of β, deflate t score and…?
- Always a problem?
  - Depends on your goal
- How to identify it?
- How to resolve it?

How to identify (multi)collinearity?

- Variance inflation factor (VIF)
  - Quantifies the multicollinearity issue
- Calculation
  - Build linear regression model on each feature
  - $X_1 = \beta_0 + \beta_2 X_2 + \ldots + \beta_n X_n$

    $\mathrm{VIF}_i = \frac{1}{1 - R_i^2}$
  - Rule of thumb: VIF > 5 sometimes 10

https://onlinecourses.science.psu.edu/stat501/node/347

Resolve multicollinearity

- No unique solution
  - Select/delete correlated variables manually
  - Add penalization to select variables ‘automatically’
- Penalization
  - Original optimization function (LSE)

    $$min(Y - X\beta)^T (Y - X\beta)$$
  - Optimization function with penalty, λ is penalty factor

    $$min(Y - X\beta)^T (Y - X\beta) + \lambda |\beta|^p$$
  - λ controls amount of regularization
    - λ approaches 0, identical model to least squares solution
    - λ approaches inf, intercept-only model

Regularization

- Regularization
  - Ridge
    - Original constraint is
    - Lagrange multiplier
    - Matrix form
    - LSE solution

$$\min \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 \ s.t. \ \sum_{j=1}^{p} \beta_j^2 \le t$$

$$\min \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \boxed{\lambda \sum_{j=1}^{p} \beta_j^2}$$

Not on intercept

$$\min (Y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta$$

$$\hat{\beta}_{ridge} = (X^T X + \boxed{\lambda I_p})^{-1} X^T Y$$

http://statweb.stanford.edu/~tibs/sta305files/Rudyregularization.pdf
https://en.wikipedia.org/wiki/Lagrange_multiplier

Coefficient estimator

- Is $\hat{\beta}_{ridge}$ still unbiased?
  - Remember LSE (no regularization) $\hat{\beta} = (X^T X)^{-1} X^T Y$ is unbiased
  - Now $\hat{\beta}_{ridge} = (X^T X + \lambda I_p)^{-1} X^T Y$

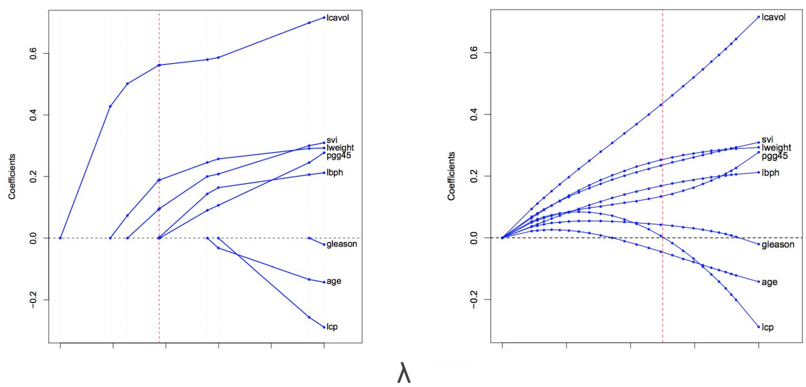
    $E(\hat{\beta}_{ridge}) = E[(I_p + \lambda (X^T X)^{-1})\hat{\beta}] = (I_p + \lambda (X^T X)^{-1})\beta$
  - $\hat{\beta}_{ridge}$ Is biased now, is that concerning??
- $$\underbrace{E[(\hat{\beta}_{ridge} - \beta)^2]}_{\text{Mean squared error}} = \underbrace{(E(\hat{\beta}_{ridge} - \beta))^2}_{\text{Bias^2}} + \underbrace{E[(\hat{\beta}_{ridge} - E(\hat{\beta}_{ridge}))^2]}_{\text{Variance}}$$

LASSO

- LASSO (least absolute shrinkage and selection operator)
  - $$min\,(Y - X\beta)^T (Y - X\beta) \;\; s.t. \; \sum_{i=1}^{p} |\beta_j| \le t$$
    $$min \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$
  - Still λ controls amount of regularization
- Why LASSO
  - Large enough λ sets some coefficients to be **0**
  - LASSO performs model selection for us

http://stats.stackexchange.com/questions/78694/how-to-interpret-the-lasso-selection-plot

LASSO vs. ridge



Elastic net

- Combination of both ridge and lasso
    - $$\lambda \sum_{j=1}^{p}((1 - \alpha)\,\beta_j^2 + \alpha\,|\beta_j|)$$
    - Advantage of ridge to shrink the magnitude of coefficients fast
    - Advantage of lasso to perform feature selection
    - alpha = 0 Ridge; alpha = 1 LASSO; (0, 1) mix;
    - library(glmnet)

Choosing lamda

- Recap role of λ
  - $\lambda \downarrow 0$, no regularization, identical solution as least square solution
  - $\lambda \uparrow \infty$, intercept only model
- How to choose λ
  - Traditional way: plot all coefficients against multiple value of λ , and choose λ when coefficients are not rapidly changing. (issue??)
    - Check function **lm.ridge()** in R, library(MASS) and **lars()**
  - Current standard practice is **cross validation**.

Cross validation

- Objective: find λ to minimize MSE
  - Bigger picture: find the optimal model (respective to λ)
- What's cross validation?
  - Partition training data T to K separate sets with equal size
    - K = 5, 10
  - For each k = 1, …, K, fit model to data excluding kth-fold T_k
  - Use fitted model on T_k to compute cross validation error
    - For example, use MSE, cv_error_k = $|T\_k|^{-1}$ Sum{(y - f(x))^2}
  - Sum over k folds, compute overall cv error
    - cv_error = K^{-1} Sum(cv_error_k)
- λ is chosen to minimize cv_error

Cross validation

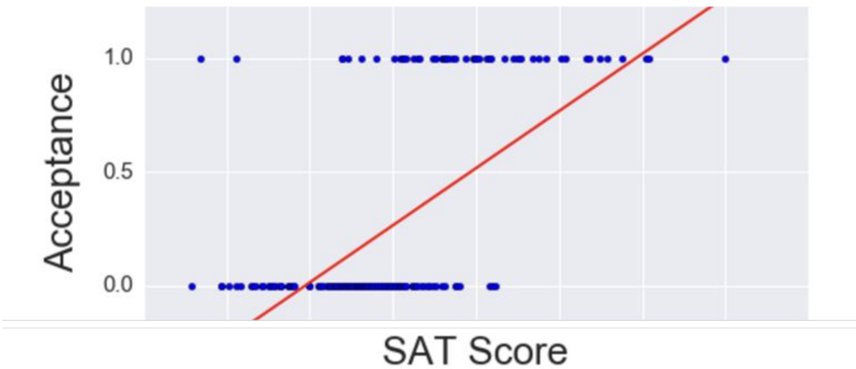
- Common type of cross validation
  - K fold
  - Leave one out

Logistic regression

- Problem to solve
  - Response Y is categorical, feature vector X to predict Y
- Examples
  - Identifying spam emails to prevent people from receiving spam
  - Predicting if borrowers will default on their loans
  - Determining whether someone has a disease to guide treatment decisions
  - Determining whether customers will churn
  - Predicting if a potential buyer will make a purchase

    …

Why not linear regression?



- Issues?

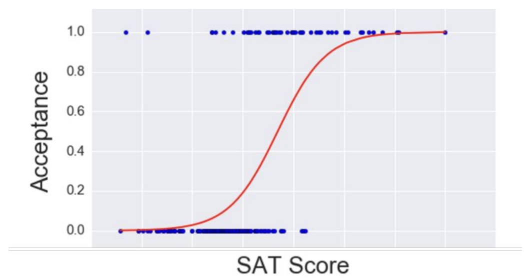
What do we need?

- Takes continuous input (e.g. -infinity to infinity)
- Produces output [0, 1]
- Has an intuitive transition
- Has interpretable coefficients (like linear regression)

Mapping feature space onto probabilities

- Modeling probabilities requires a functional form that maps onto interval [0,1]
  - Typical choice is the logistic function*

$$\hat{p} = h_\theta(x) = \frac{1}{(1 + e^{-\theta^T x})}$$



* Other less common choices include the inverse Gaussian ("probit") and the hyperbolic tangent functions.

Logistic regression - basics

- Very popular binary classifier
  - Recall Bernoulli random variable
    $$f(k;p) = p^k(1-p)^{1-k} \text{ for } k \in \{0,1\}$$
  - Logistic regression estimates parameter p of the Bernoulli
- Estimates probability that an observation is in a given category based on the observation’s features
- Regression step estimates the probability
- Classification step rounds the probability to 0 or 1

Relationship with linear regression

- Logistic model of probability is equivalent to a linear model of the log-odds ratio

$$h_\theta(x) = \frac{1}{(1 + e^{-\theta^T x})} \rightarrow \ln\left(\frac{p}{1-p}\right) = \theta^T x$$

Decision boundary

- The predicted result flips from 0 to 1 in a certain region of the feature space

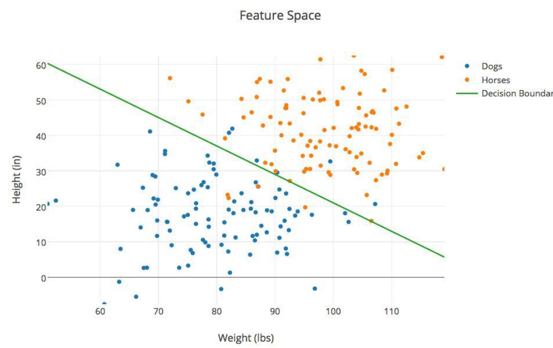
$$h_\theta(x) = .5$$

$$\rightarrow \frac{1}{1+e^{-\theta^T x}} = .5$$

$$\rightarrow 1 = e^{-\theta^T x}$$

$$\rightarrow \theta^T x = 0$$

- That region is called the "decision boundary"



Coefficients estimation

- Coefficients for logistic regression can be estimated using Maximum Likelihood Estimation (MLE)
- Recall that MLE picks model (coefficients) that maximizes likelihood of observations

$$\underset{\vec{\theta}}{\operatorname{argmax}}\ P(X|\vec{\theta})$$

Coefficients estimation

- Likelihood of an observation given the model:

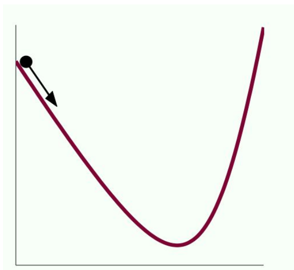
$$p(y_i|x_i;\theta) = h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{1-y_i}$$

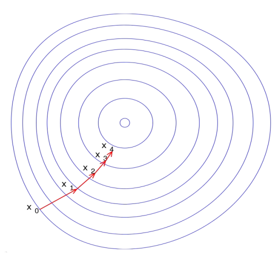
- Assuming each observation is independent:

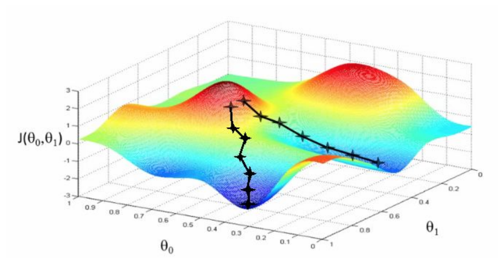
$$p(\vec{y}|X;\theta) = \prod_{i=1}^{n} h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{1-y_i}$$

$$\ln p(\vec{y}|X;\theta) = \sum_{i=1}^{n} \left(y_i \ln h_\theta(x_i) + (1 - y_i)\ln(1 - h_\theta(x_i))\right)$$

Gradient descent



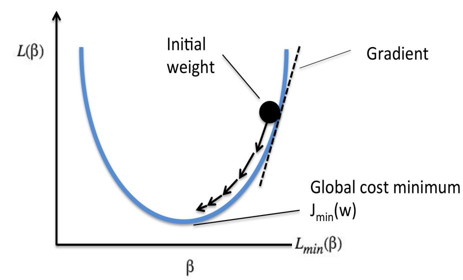




Gradient descent steps

- Step 0: find an initial $\beta_j^{(0)}$
- Step 1: $\beta_j^{(t+1)} \leftarrow \beta_j^{(t)} - \eta \dfrac{\partial l(\beta)}{\partial \beta_j}$

Learning rate   Gradient



- step 2: check if $\nabla_\beta l(\beta) = 0$

  if not, repeat step 1.

$$\beta_0^{(t+1)} \leftarrow \beta_0^{(t)} - \eta \sum_i \left(y_i - \hat{p}(y=1 \mid x_i, \beta^{(t)})\right)$$

$$\beta_j^{(t+1)} \leftarrow \beta_j^{(t)} - \eta \sum_i x_{ij} \left(y_i - \hat{p}(y=1 \mid x_i, \beta^{(t)})\right)$$

Coefficients estimation

- We estimate the model parameters to minimizing:

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)} + (1-y^{(i)}) \log\left(1-h_\theta(x^{(i)})\right)\right]$$

- Which has a gradient:

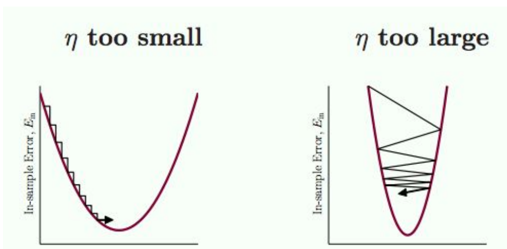
$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right)$$

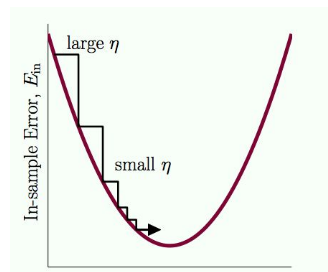
- So we can find the minimum by iteratively doing:

$$\theta_j := \theta_j - \alpha \quad \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$
$$(j = 0, 1, 2, 3, \ldots, n)$$

Gradient descent learning rate

- Impact of learning rate on convergence speed





Interpreting coefficients

- Logistic regression implies a linear relationship between the features and the logit odds:

$$\ln \frac{p}{1-p} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_k x_k$$

- Increasing feature value by 1 increases logit odds by θ and odds by e^θ

Underfitting and overfitting

- **Underfitting**: The model doesn't fully capture the relationship between predictors and the target. The model has not learned the data's signal.

  → What should we do if our model underfits the data?

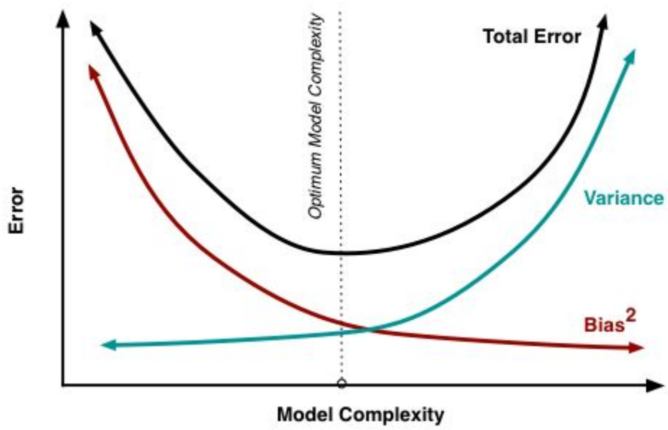
- **Overfitting**: The model has tried to capture the sampling error. The model has learned the data's signal and the noise.

  → What should we do if our model overfits the data?

“HELP, my model is overfitting!”

- You have a few options.
  - Get more data: not always possible/practical
  - Subset Selection: keep only a subset of your predictors (i.e, dimensions)
  - Regularization: restrict your model’s parameter space
  - Dimensionality Reduction: project the data into a lower dimensional space

The bias/variance tradeoff



How is the **bias/variance tradeoff** related to **underfitting** and **overfitting**?

How can we find the best tradeoff point? I.e. The optimum model complexity

Logistic regression with regularization

- We model the world as:

$$h(x) = \frac{1}{1+e^{-\theta x}}$$

- We estimate the model parameters to minimizing:

The "regularization" parameter

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

$$\min_\theta J(\theta)$$

The term being penalized

Logistic regression with regularization

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

Note that you should *not* be regularizing θ_0 which is used for the bias term.

Correspondingly, the partial derivative of regularized logistic regression cost for θ_j is defined as

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \qquad \text{for } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \qquad \text{for } j \geq 1$$

Logistic regression with regularization

- Finding coefficient by gradient descent
- The coefficient updating scheme for regularized logistic regression

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

- Compared to original logistic regression

$$\theta_j := \theta_j - \alpha \quad \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

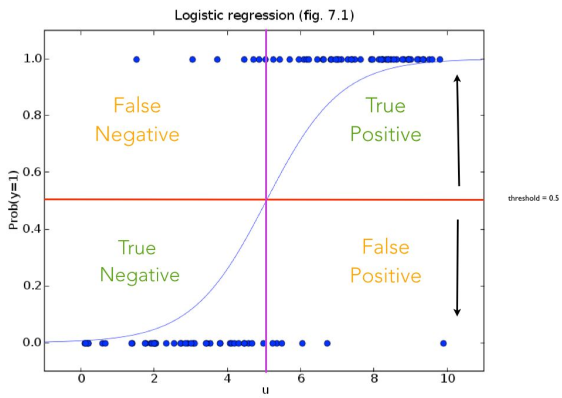
$$(j = 0, 1, 2, 3, \ldots, n)$$

Evaluate logistic regression

- Type I and Type II Error

	H_0 is true	H_0 is false
Accept H_0	Correct Decision (1-α)	Type II Error (β)
Reject H_0	Type I Error (α)	Correction Decision (1-β)

Binary classification results



- True Positives (TP): Correct positive predictions
- False Positives (FP): Incorrect positive predictions (false alarm)
- True Negatives (TN): Correct negative predictions
- False Negatives (FN): Incorrect negative predictions (a miss)

	Predicted Yes	**Predicted No**
Actual Yes	True positive	False negative
Actual No	False positive	True negative

Binary classification results

		True class	
		p	**n**
Hypothesized class	**Y**	True Positives	False Positives (Type I error)
	N	False Negatives (Type II error)	True Negatives
Column totals:		**P**	**N**

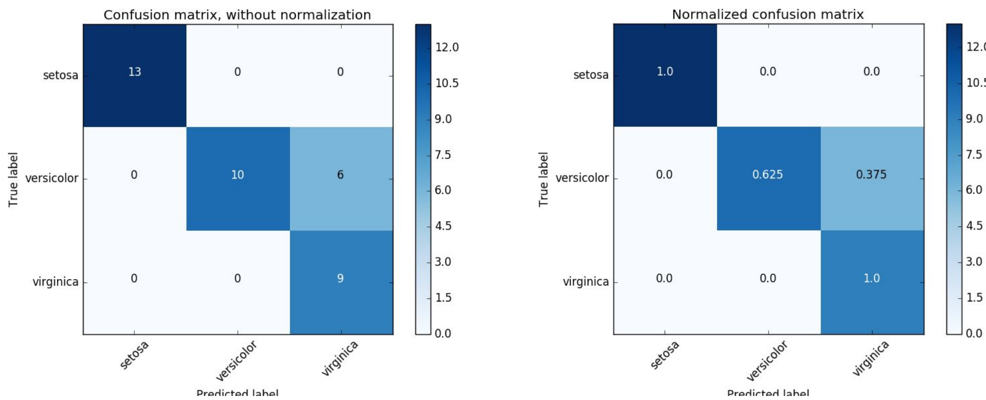
$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

Multinomial confusion matrix



How is threshold affecting the metrics?

$y = 1/(1 + e^{5-x})$

size	1	2	3	4	5	6	7	8	9	10
prob	0.018	0.047	0.119	0.269	0.5	0.731	0.881	0.923	0.982	0.993
actual	0					1				

t = 0.05

	positive	negative
positive	5	3
negative	0	2

Sensitivity: 100%
Specificity: 40%

t = 0.9

	positive	negative
positive	3	0
negative	2	5

Sensitivity: 60%
Specificity: 100%

What about when t = 0 and t = 1?

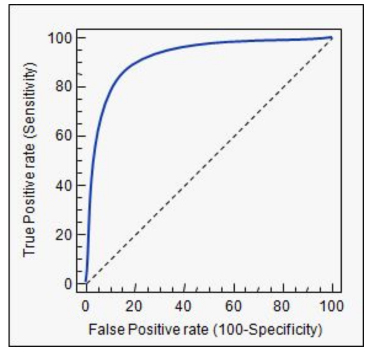
ROC curve

- ROC is a plot of the TPR against the FPR for a binary classification problem as you change the threshold
  - y-axis: True Positive Rate (aka Recall)

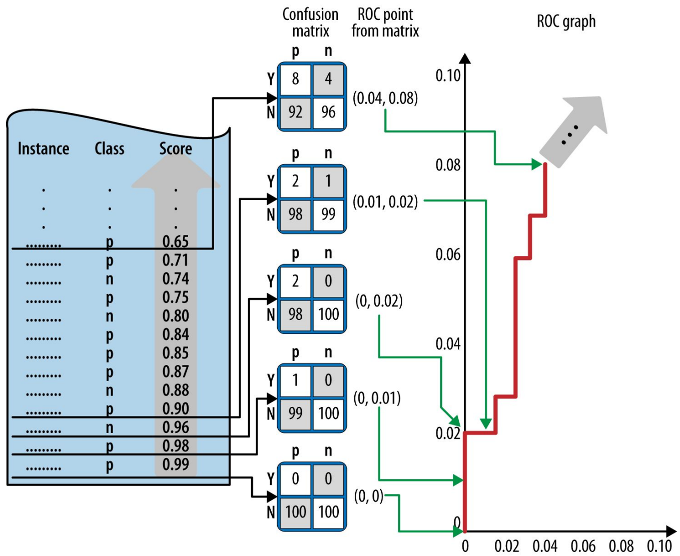
$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

  - x-axis: False Positive Rate (aka 1 – Specificity)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$



Building ROC curve



Building ROC curve

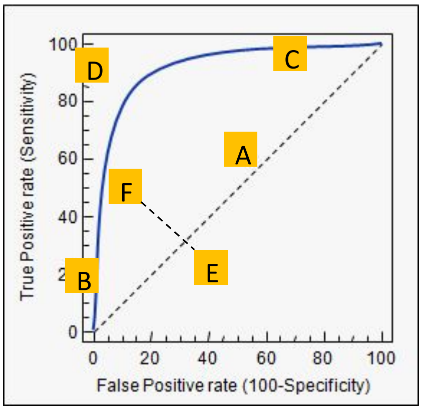
For a given model f, each threshold value T gives a point on the ROC Curve

Model score is the probability of class membership $(Y = 1)$

1. Allow T to be the maximum score
2. $TP = 0, FP = 0$
3. For each observation, i:
   - If $\hat{\pi}_i > T \longrightarrow$ increment TP
   - Else $\longrightarrow$ increment FP
4. Add point (FP/N, TP/P) to the ROC Graph

Increment T from max-score to min-score, repeating steps 1-4

Understanding ROC curve



A: line y=x, Random guessing the class, no model

B: Positive predicted only on strong evidence, low FP rate, low TP rate

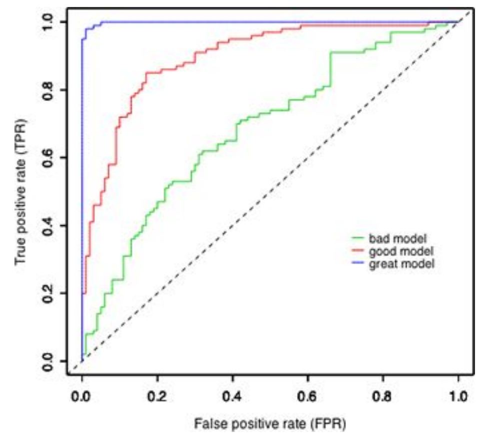
C: Positive predicted with weak evidence, high TP rate, high FP rate also

D: High TP rate with low FP rate, ideal model

E: Worse than random guessing, negation of **F**

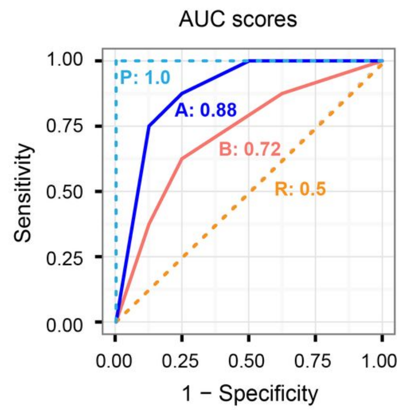
Model selection from ROC curve

- ROC Curve
  - If classifier A's ROC curve is strictly greater than classifier B's, then classifier A is always preferred



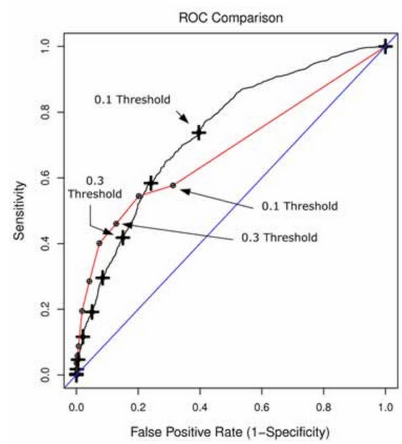
Model selection from ROC curve

- ROC - Area Under Curve (AUC)
  - Equals the probability that the model will rank a randomly chosen positive observation higher than a randomly chosen negative observation
  - Useful for comparing different classes of models in general setting



Model selection from ROC curve

- ROC Curve
  - If two classifier's ROC curves intersect, then the choice depends on relative importance of sensitivity and specificity



Summary

- Collinearity
  - Regularization
- Cross validation
- Logistic regression
  - Relationship with linear regression
  - Gradient descent
  - Evaluation model performance

Summary

- Logistic regression
  - Definition, sigmoid function
  - Relationship with linear regression
  - Decision boundary
  - Coefficients estimation
  - Gradient descent
  - Interpreting Coefficients
  - Regularization
  - Evaluation model performance
    - Confusion matrix
    - ROC curve