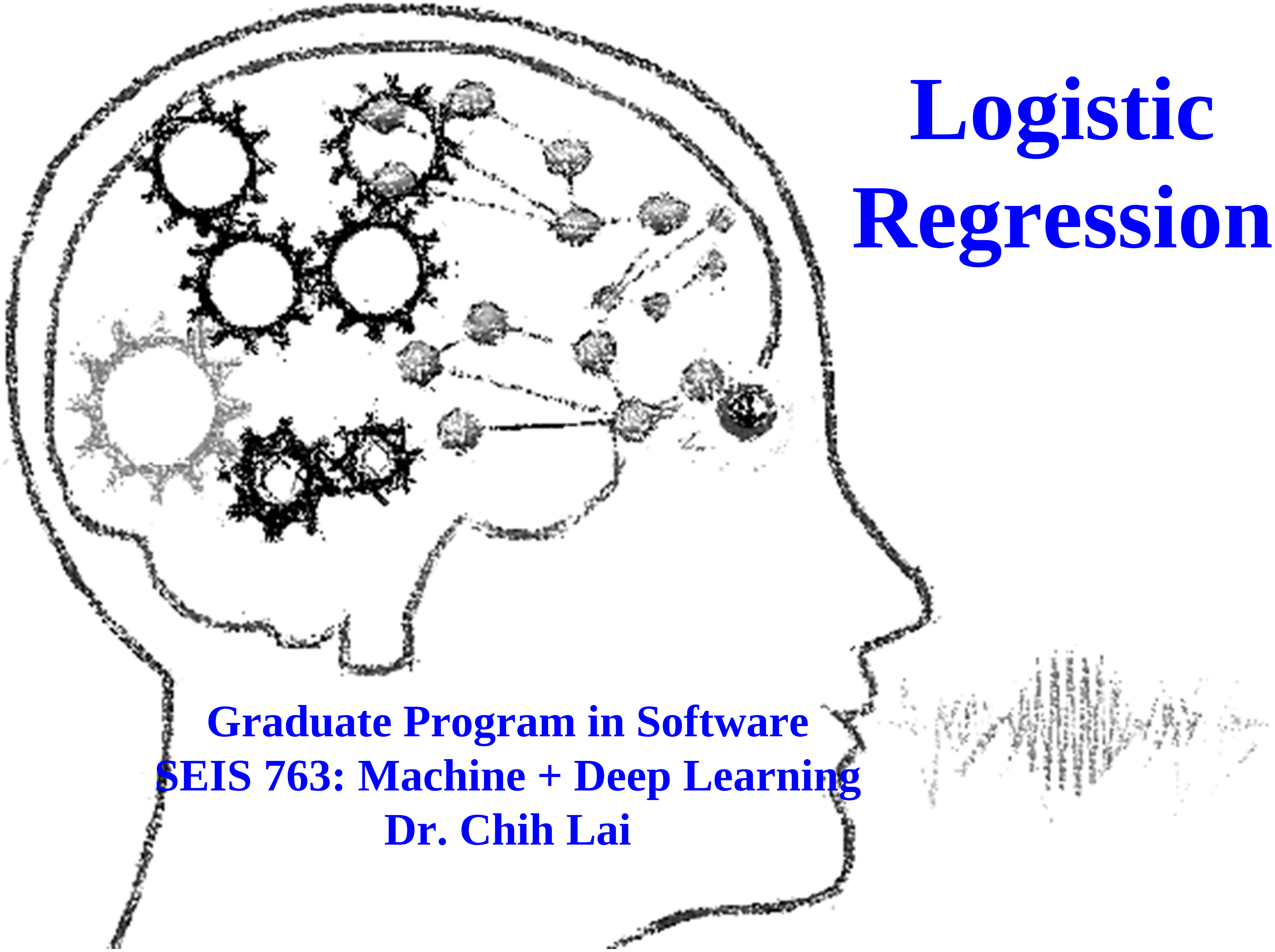
Logistic Regression



Graduate Program in Software
SEIS 763: Machine + Deep Learning
Dr. Chih Lai

Matlab / Python Logit-Related Functions and References

- Matlab Functions: <fitglm, predict> <mnrfit, mnrval>

- Overall
  - http://www.mathworks.com/help/stats/generalized-linear-regression.html
  - http://www.mathworks.com/help/stats/generalized-linear-regression-2.html

- Generalized LinearModel class
  - **http://www.mathworks.com/help/stats/generalizedlinearmodel-class.html**
  - **http://www.mathworks.com/help/stats/examples/fitting-data-with-generalized-linear-models.html**
  - **Multinomial logistic regression https://www.mathworks.com/help/stats/mnrfit.html#namevaluepairarguments**

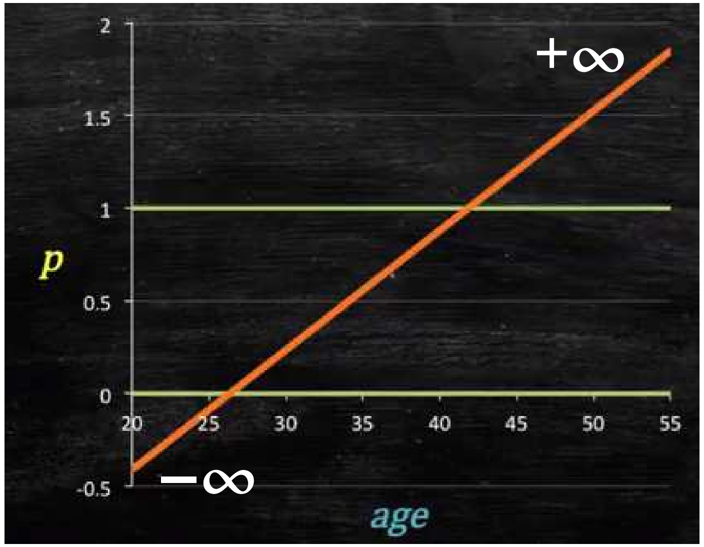
- **Python scikit-learn, logistic regression**
  - **http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html**
  - **http://scikit-learn.org/0.15/modules/generated/sklearn.linear_model.LogisticRegression.html**
  - **L1 Penalty & Sparsity** http://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_l1_l2_sparsity.html
  - **LogisticRegressionCV** http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html
  - **Multiclass and multilabel algorithms** http://scikit-learn.org/stable/modules/multiclass.html

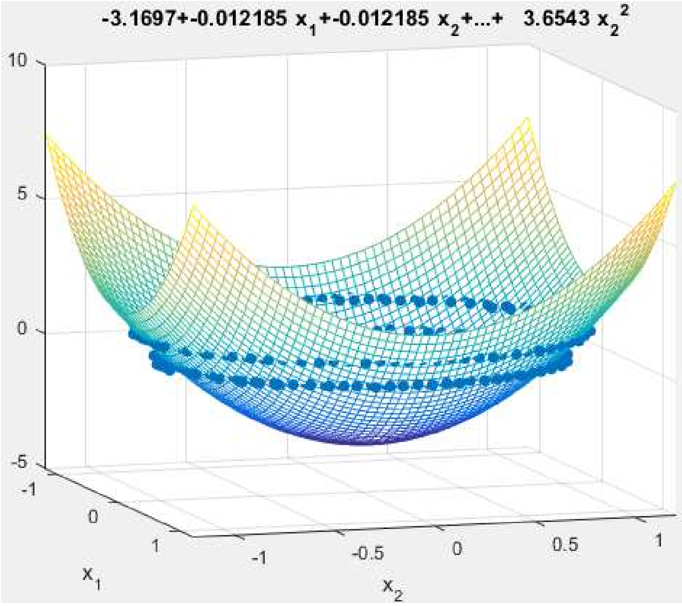
Outline

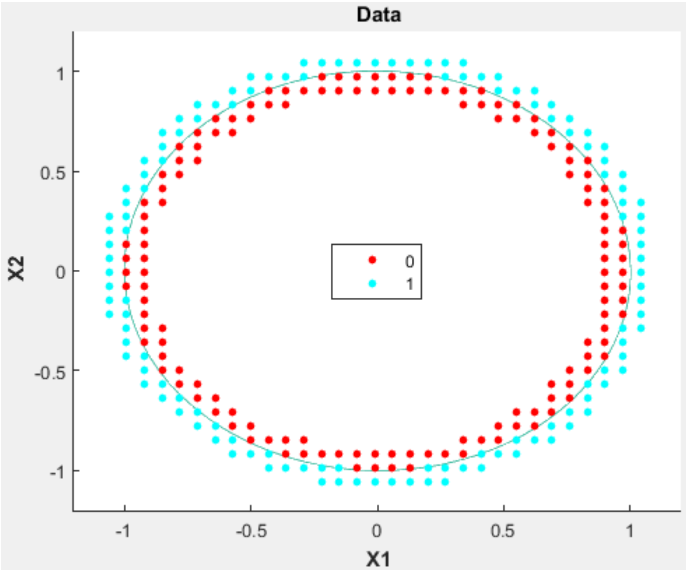
- **How it works? Matlab / Python functions.**
- **Diagnoses / Visualization.**
- Prediction and Quality.
- Outliers.
- Multiclasses Prediction.
- Nonlinearity / High Dimensionality / Visualization.
- Regularization.
- Cost Function.

Why Not Just Linear Regression?

- Linear Regression $\quad h_\theta(x) = \theta^T X \in [-\infty .. +\infty]$ (i.e. **ANY** range in ℜ).



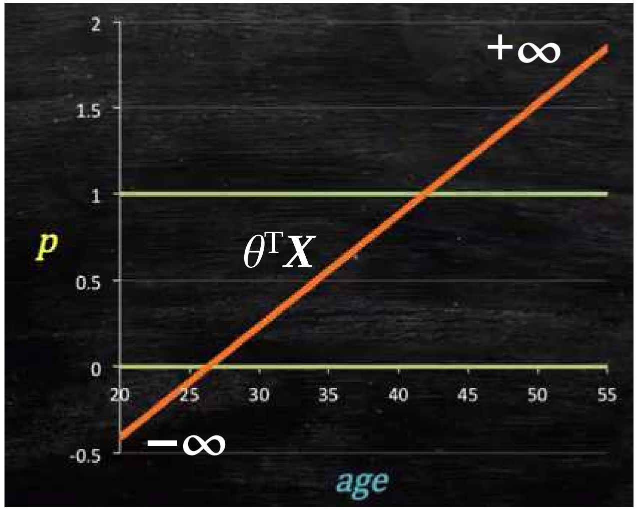




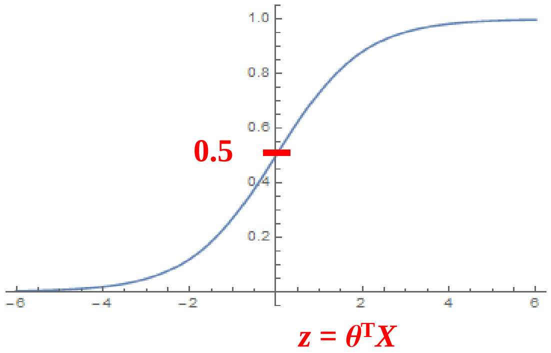
- But, outcome may be categorical…
  - Gender, Rank, City…
  - Buy or no buy…

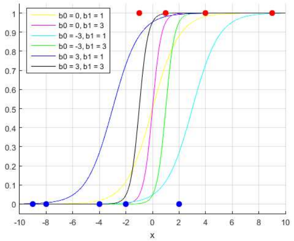
Logistic Regression…Why Not Just Linear Regression?

- Linear Regression $\boldsymbol{h}_\theta(\boldsymbol{x}) = \boldsymbol{z} = \theta^{\mathrm{T}}\boldsymbol{X} \in [-\infty \, .. \, +\infty]$ (i.e. **ANY** range in ℝ).
  - But, outcome may be categorical… Gender, Diseases, Rank, …



$^{1}/_{1+e^{-z}}$ **Sigmoid**



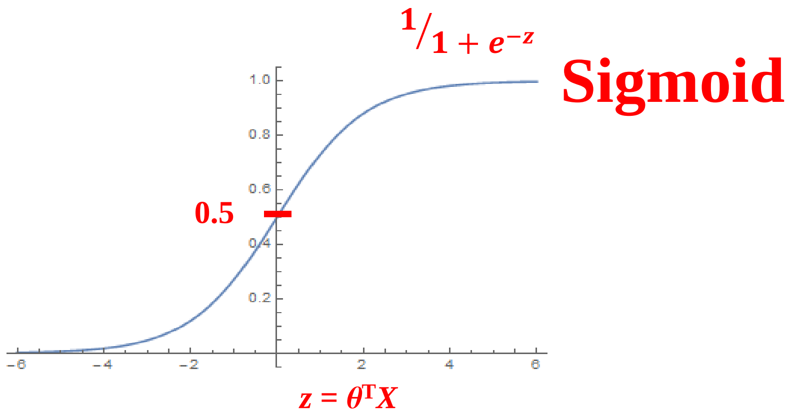


- Logistic (or logit) regression is a model where the dependent var is **categorical**.
  - Predict classes by ***linking*** $\theta^{\mathrm{T}}X$ to probabilities [0..1].

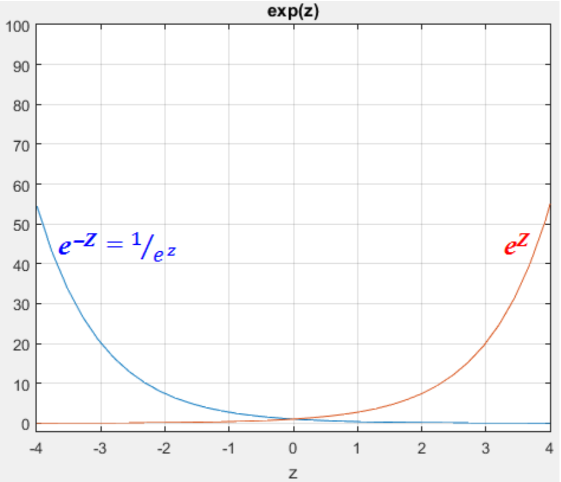
Link Categorical Response to Continuous Probability

- Logistic (or logit) regression is a model where the dependent var is categorical.
  - Predict classes by ***linking*** $\theta^{T}X$ to ***likelihood*** [0..1] via ***logistic function***.
    - https://en.wikipedia.org/wiki/Logistic_regression
  - Also refer to as ***generalized linear model*** (**GLM**).

- Why $\boldsymbol{P = {}^{1}/_{1+e^{-z}}}$? ➔ guarantee $\mathbf{0 \le P \le 1}$
  - $e^{z} \geq \mathbf{0}$ for $z \in [-\infty, \infty]$, no matter what is $\boldsymbol{z}$.
  - $\frac{P}{1-P} = e^{\theta^{T}X} = e^{z}$ $\qquad P = {}^{1}/_{1+e^{-z}}$
    - $\frac{P}{1-P}$ is ***odds*** = prob. of "true" over prob. of "not true".

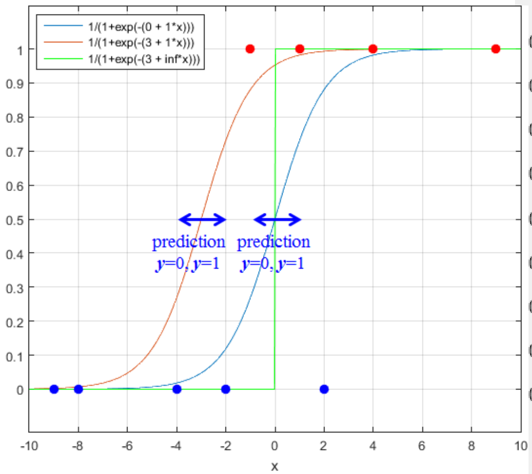


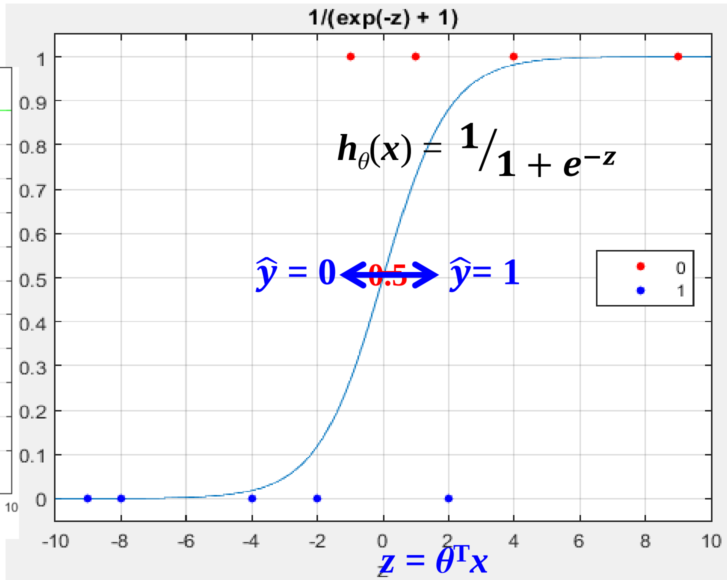

```
syms z y
ezplot('exp(-z)', [-4 4 -2 100])
hold on
ezplot('exp(z)', [-4 4 -2 100])
hold off
grid on
```



Logistic Regression and Interpretation

- Logistic regression → $0 \le \boldsymbol{h}(\theta^{\mathrm{T}}\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{z}) = {}^{1}/_{1+e^{-z}} \le 1$, **asymptotes at 1 or 0**.
  - ${}^{1}/_{1+e^{-z}}$ is referred to as ***Sigmoid function*** or logistic function.
    - If $\theta\uparrow \Rightarrow \theta^{\mathrm{T}}X\uparrow \Rightarrow \boldsymbol{z}\uparrow \Rightarrow \boldsymbol{h}_\theta(\boldsymbol{x}) = {}^{1}/_{1+e^{-z}} \approx \mathbf{1}$. predict $\hat{y} = 1$.
    - If $\theta\downarrow \Rightarrow \theta^{\mathrm{T}}X\downarrow \Rightarrow \boldsymbol{z}\downarrow \Rightarrow \boldsymbol{h}_\theta(\boldsymbol{x}) = {}^{1}/_{1+e^{-z}} \approx \mathbf{0}$. predict $\hat{y} = 0$.
    - If $\boldsymbol{z} = 0 \Rightarrow \boldsymbol{h}_\theta(\boldsymbol{x}) = {}^{1}/_{1+e^{-z}} \approx$ **??**
  - θs tell us which attributes have **positive** or **negative impact** on ***P***, or classification.



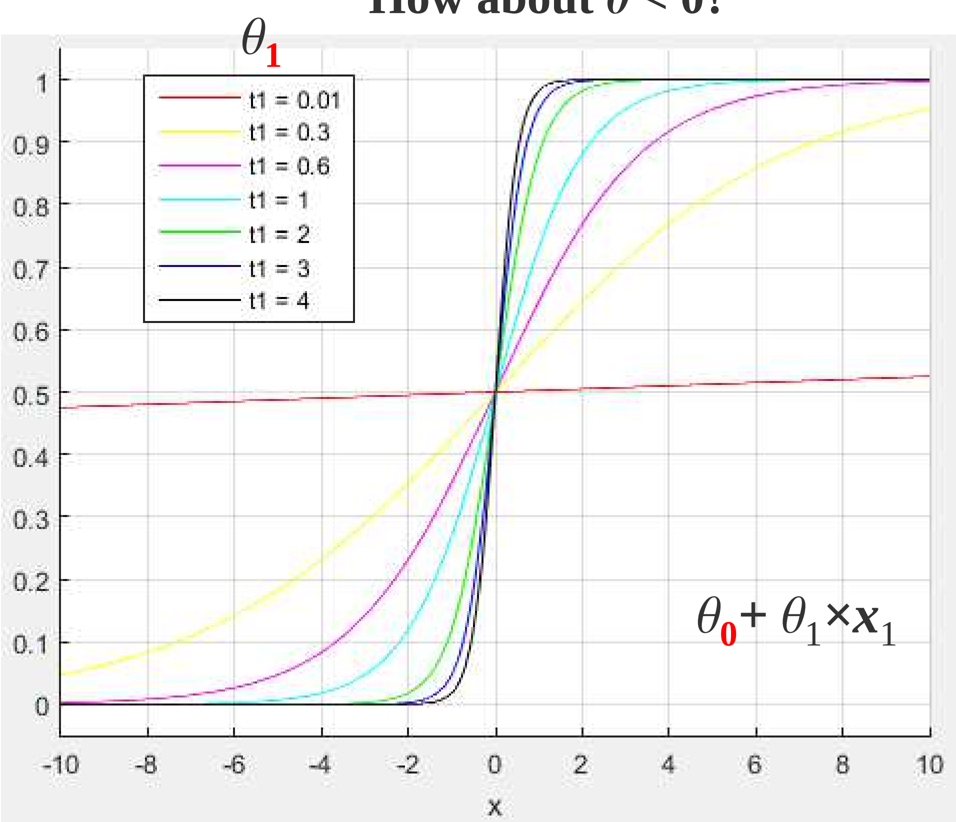



```
syms z
figure,
ezplot(1 / (1 + exp(-z)), [-10 10])

red = [-1 1; 1 1; 4 1; 9 1];
blue =[2 0; -2 0; -4 0; -8 0; -9 0];
data = [red; blue];
Y = [zeros(size(red, 1), 1); ...
            ones(size(blue, 1), 1)];
hold on
gscatter(data(:, 1), data(:, 2), Y, 'rb')
hold off
ylim([-0.05 1.05])
grid on
```

Logistic Function– Changing θ_1

- Logistic (sigmoid) function $\boldsymbol{\sigma}$ takes LR output ($\hat{y}$) as input & convert to ***P*** **[0..1]**.
  - For smaller θ_1, ***x*** must be very large to reach ***P*** = 1.

How about θ < 0?



See Appendix for
changing both θ_0 & θ_1

```
syms x
t0 = 0;                                    % θ0
t1_arr = [0.01, 0.3, 0.6, 1, 2, 3, 4];     % θ1 slope
fstr = '1/(1+exp(-(t0 + t1*x)))';
c = ['rymcgbk'];                 % line colors
figure, hold on
loops = 0;  legendStr = [];
for t1 = t1_arr
    fstrX = strrep(fstr, 't0', num2str(t0));
    fstrX = strrep(fstrX, 't1', num2str(t1));
    h = ezplot(fstrX, [-10 10]);
    loops = loops + 1;
    set(h, 'color', c(loops));
    legendStr{loops} = ['t1 = ' num2str(t1)];
end
legend(legendStr),   title(''),    grid on,
ylim([-0.05 1.05]),     hold off
```

Matlab fitglm() Functions

- Matlab fitglm() functions.
  - GLM– Generalized Linear Model.

- fitglm().
  - mdl = **fitglm**(X, Y, 'distr', '**binomial**', 'link', '**logit**')

See Appendix for
1. Different distributions,
2. Different link functions.

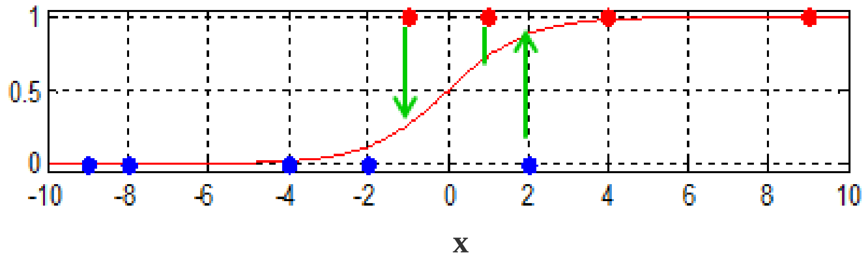
Link Function Name	Link Function	Mean (Inverse) Function
'log'	$f(\mu) = \log(\mu)$	$\mu = \exp(Xb)$
'logit'	$f(\mu) = \log(\mu/(1-\mu))$	$\mu = \exp(Xb) / (1 + \exp(Xb))$
'probit'	$f(\mu) = \Phi^{-1}(\mu)$	$\mu = \Phi(Xb)$
'comploglog'	$f(\mu) = \log(-\log(1-\mu))$	$\mu = 1 - \exp(-\exp(Xb))$
'reciprocal'	$f(\mu) = 1/\mu$	$\mu = 1/(Xb)$

$e^z/_{1+e^z} = 1/_{1+e^{-z}}$

Main Output from fitglm()

```
red = [ -1 1; 1 1; 4 1; 9 1];                % class 1
blue =[-9 0; -8 0; -4 0; -2 0; 2 0];         % class 0
X = [blue(:, 1); red(:, 1);];
Y = [blue(:, 2); red(:, 2)];
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
```

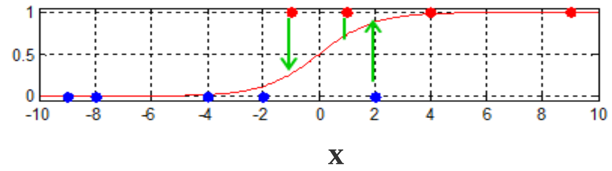
$$h_\theta(x) = P = {}^{1}/_{1+e^{-z}}$$




```
Estimated Coefficients:
                   Estimate      SE         tStat       pValue
                      θ       ________    _________    ________

    (Intercept) θ0  -0.0146     0.9583    -0.015235     0.98784
    x1          θ1  0.51551    0.35954       1.4338     0.15162
```

- How do we decide best θ? ←←←
- Class probability of each instance?

sklearn Logistic Regression



- sklearn logistic regression requires **<u>at least</u>** 2-D data.
  - So we copy 1^{st}-D data into 2^{nd}-D.

```
import numpy as np
from sklearn import linear_model

X = [[-1, -1],  [1, 1],  [4, 4],  [9, 9], [-9, -9], [-8, -8], [-4, -4], [-2, -2], [2, 2]]

Y = [1,     1,     1,     1,     0,     0,     0,     0,     0]

# Inverse of regularization strength; ↓C ==> ↑regularization
# LOSS = C * E + theta,   default C = 1
logreg = linear_model.LogisticRegression(C = 90)

logreg.fit(X, Y)                                    # fit the data
print(logreg.intercept_, logreg.coef_)
```

```
%%%%% Matlab logistic regression, 2-D
X = [-1, -1;  1, 1;  4, 4;  9, 9; -9, -9; -8, -8; ...
             -4, -4; -2, -2; 2, 2];
Y = [1,    1,    1,    1,    0,    0,    0,    0,    0];
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
```

⇨ # [-0.01457369]   [[0.25773197,   0.25773197]]

⇨

(Intercept)	-0.0146
x1	0.51551
x2	0

Detailed Information Returned from Matlab Logistic Regression

- Model info returned from fitglm()
  - Use workspace to exam info in the model.

http://www.mathworks.com/help/stats/generalizedlinearmodel-class.html

- **Coefficients** **.Estimate (θ)**
- **Fitted**
  - **.Response (P)** **=** **.Probability (P)**
  - **.LinearPredictor ($z = \theta^T X$)**
- **Residuals,** **.Raw**
- **SSE, SST, SSR**
- **Diagnostics,** **.Leverage** **.CooksDistance**
- **Rsquared**

- **LogLikelihood**

mdl | mdl.Fitted | mdl.Coefficients | mdl.Residuals

1x1 GeneralizedLinearModel

Property	Value
Dispersion	1
DispersionEstimated	0
Deviance	6.7354
Offset	[0;0;0;0;0;0;0;0;0]
Distribution	1x1 struct
Link	1x1 struct
Residuals	9x5 table
Fitted	9x3 table
Diagnostics	9x3 table
Steps	[ ]
Formula	1x1 LinearFormula
LogLikelihood	-3.3677
DFE	7
SSE	1.1714
SST	2.2222
SSR	1.1319
CoefficientCovariance	[0.9183,0.0124;0.0124,0.1293]
CoefficientNames	1x2 cell
NumCoefficients	2
NumEstimatedCoefficients	2
Coefficients	2x4 table
Rsquared	1x1 struct
ModelCriterion	1x1 struct
VariableInfo	2x4 table
ObservationInfo	9x5 table
Variables	9x2 table
NumVariables	2
VariableNames	2x1 cell
NumPredictors	1
PredictorNames	1x1 cell
ResponseName	'y'
NumObservations	9
ObservationNames	0x0 cell

Predicted Response (Probability)

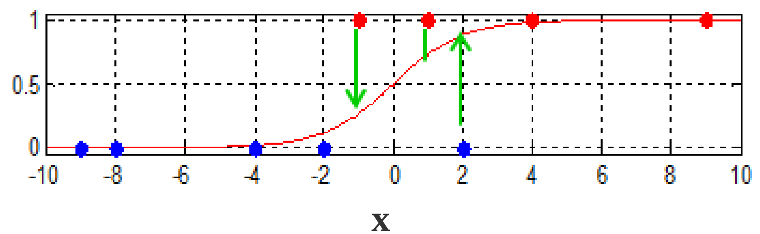
- mdl.**Fitted.Response** = mdl.**Fitted.Probability** = probability = $^1/_{1+e^{-z}}$ = 1 ./ (1+exp(-***Z***))
  - = **predict**(mdl, ***X***);
  - ***Z*** = mdl.**Fitted.LinearPredictor** = $\theta^{T}X$.

		θ
(Intercept)	θ_0	-0.0146
x1	θ_1	0.51551

mdl.Fitted.Response
= probability
= 1 ./ (1+exp(-Z))

X		1 Response	Y	$\hat{Y}$
-9	1	0.0094	0	
-8	2	0.0157	0	
-4	3	0.1114	0	
-2	4	0.2601	0	
2	5	0.7343	0	1
-1	6	0.3705	1	0
1	7	0.6227	1	
4	8	0.8857	1	
9	9	0.9903	1	

Residual = ($\boldsymbol{Y} - \hat{\boldsymbol{Y}}$)



mdl = **fitglm**(X, Y, 'distr', 'binomial', 'link', 'logit')

B = mdl.Coefficients.Estimate;   **% θ = -0.0146   0.5155**

XX = [ones(length(X), 1) X];

Z = XX * B;   **% $\theta^{T}X$**

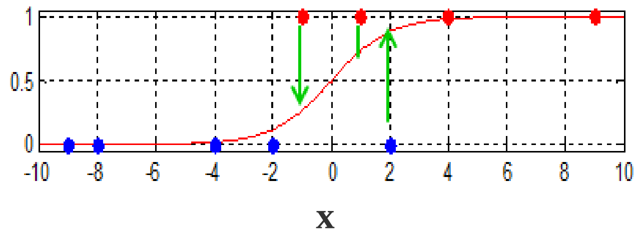
[Z   mdl.Fitted.LinearPredictor]   **% $\theta^{T}x = z$**

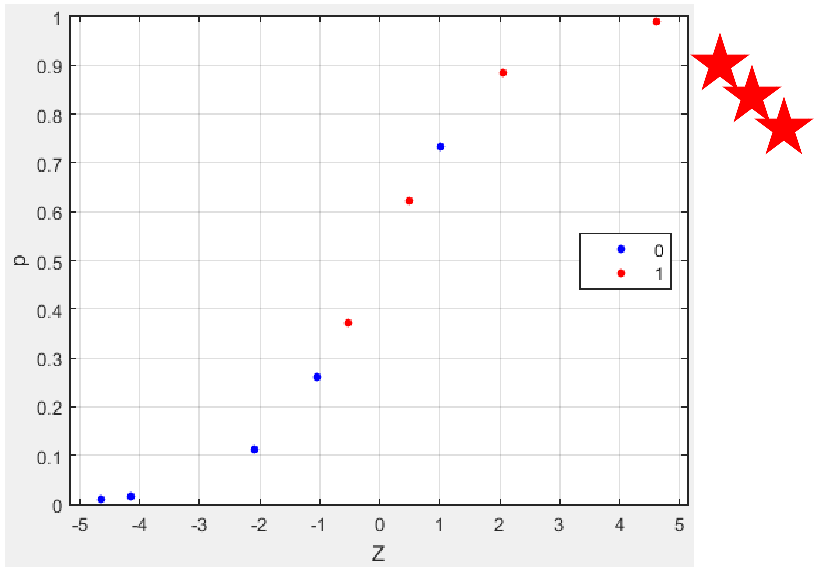
[1 ./ (1+exp(-Z))   **predict**(mdl, X)   ...   **% probability**

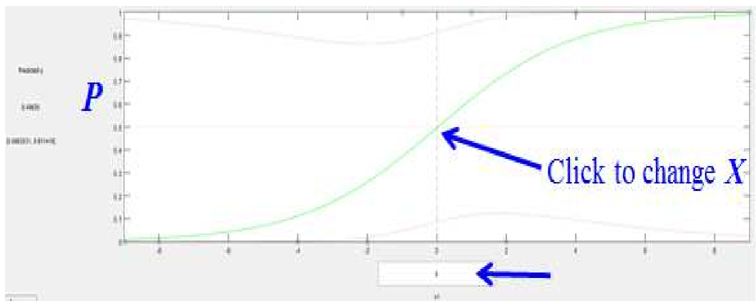
mdl.Fitted.Probability , mdl.Fitted.Response]

Plotting Probability = $^1/_{1+e^{-z}}$ Against *X* or *Z*

```
red = [ -1 1; 1 1; 4 1; 9 1];
blue =[-9 0; -8 0; -4 0; -2 0; 2 0];
X = [blue(:, 1); red(:, 1);];
Y = [blue(:, 2); red(:, 2)];

mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')


%% Plot Z wrt P
p = mdl.Fitted.Response;              % probability
Z = mdl.Fitted.LinearPredictor;       % Z = θ^T X
figure, gscatter(Z, p, Y, 'br');  grid on
plotSlice(mdl)
```

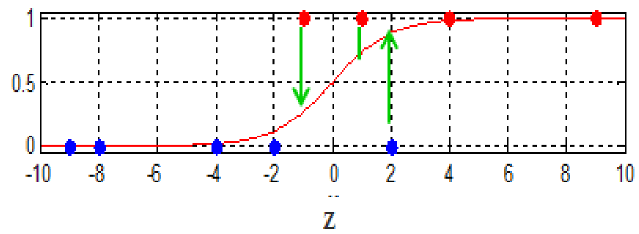






Accuracy & Confusion Matrix

- Class prediction = $\boldsymbol{P} \geq 0.5$

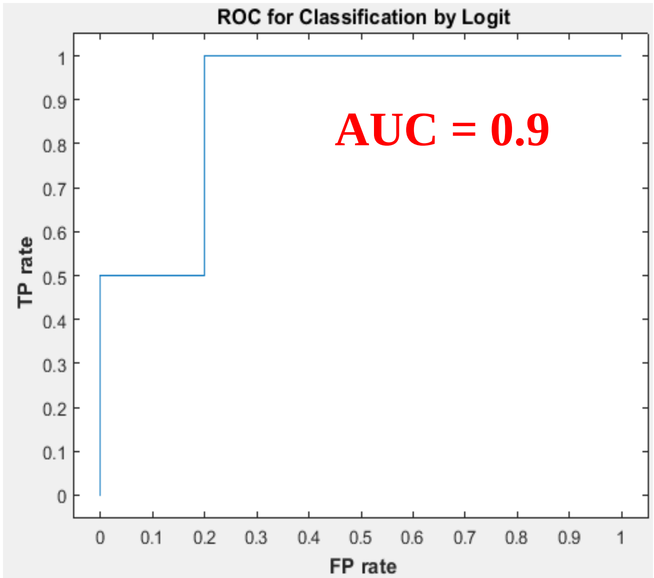



```
red = [ -1 1; 1 1; 4 1; 9 1];
blue =[-9 0; -8 0; -4 0; -2 0; 2 0];
X = [blue(:, 1); red(:, 1);];
Y = [blue(:, 2); red(:, 2)];

mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')

scores = predict(mdl, X)
P_Labels = double(scores >= 0.5);

CFM = confusionmat(Y,  P_Labels)

[xpos, ypos, T, AUC] = perfcurve(Y, scores, 1);
figure,     plot(xpos, ypos)     % plot ROC

xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('\bf FP rate'),  ylabel('\bf TP rate')
title('\bf ROC for Classification by Logit')
```

CFM for class-0

	0	1
0	4 (TP)	1 (FN)
1	1 (FP)	3 (TN)

Accuracy = 0.78



Fuzzy Probability Boundary

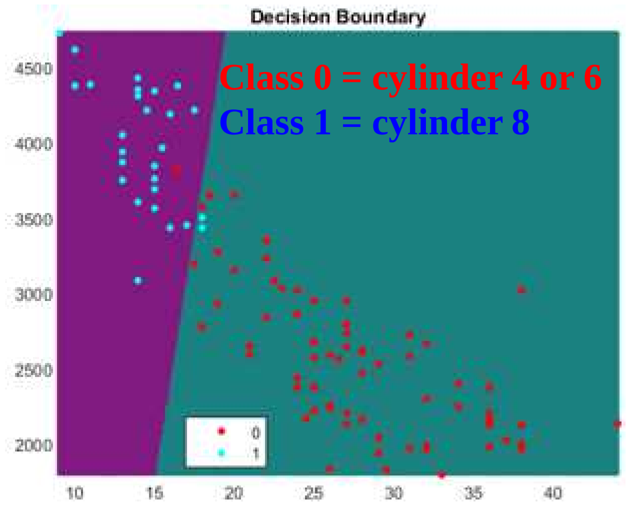
- Predicting Car's Cylinder from **MPG + Weight**

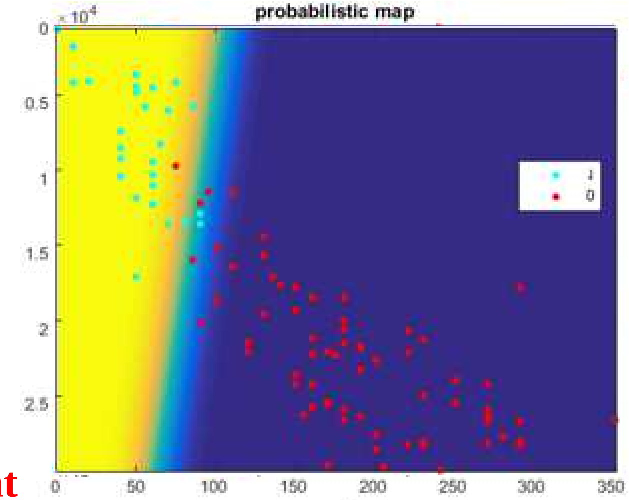
%% use fitglm() to build a model first.

```
load carsmall; X = ([MPG Weight]);     Y = (Cylinders >= 8);
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
```

%% créate test data for prediction

```
[x1Grid, x2Grid] = meshgrid(min(X(:, 1)) : 0.01 : max(X(:, 1)), ...
                   min(X(:, 2)) : 0.01 : max(X(:, 2)));

xGrid = [x1Grid(:), x2Grid(:)];

probability = predict(Your_Mdl, xGrid);

imagesc(reshape(probability, size(x1Grid)));
```

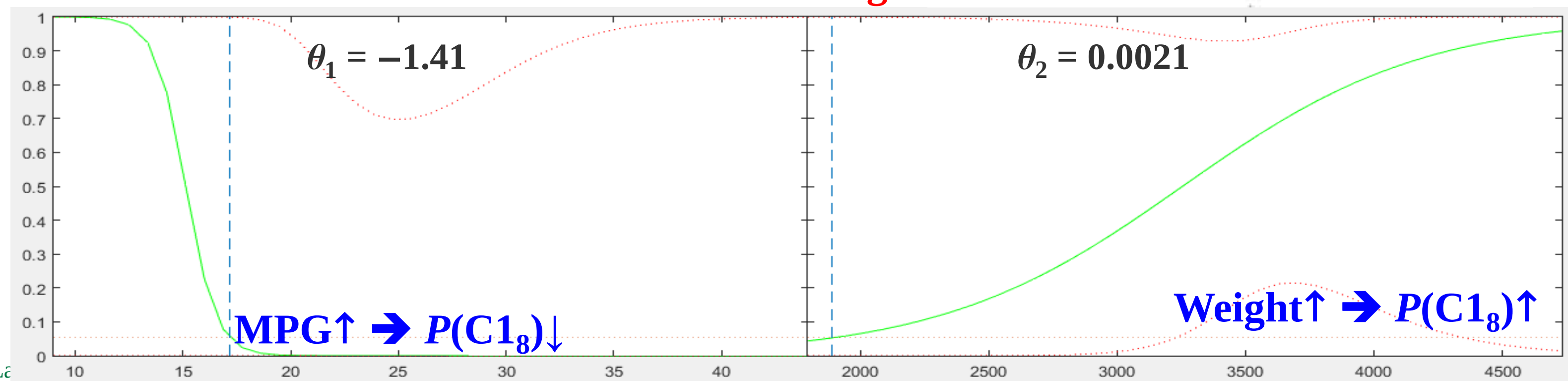




MPG + Weight

CFM = 0.92

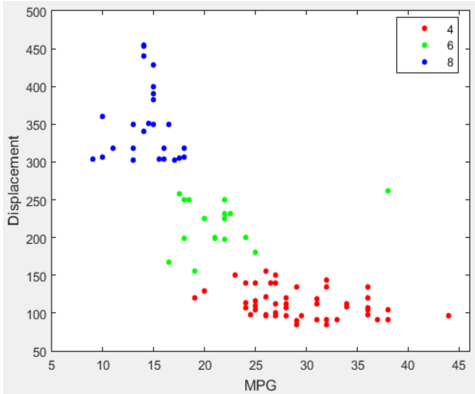
67	1
7	25

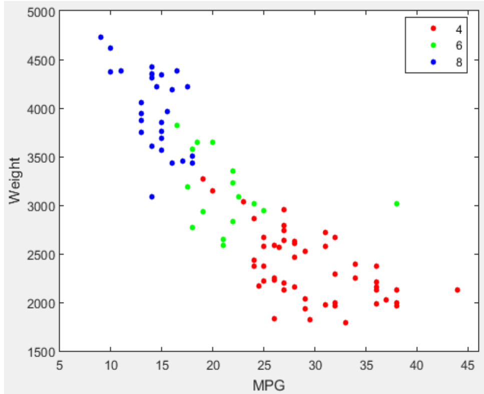


Predicting Car's Cylinder from **MPG + Displacement**

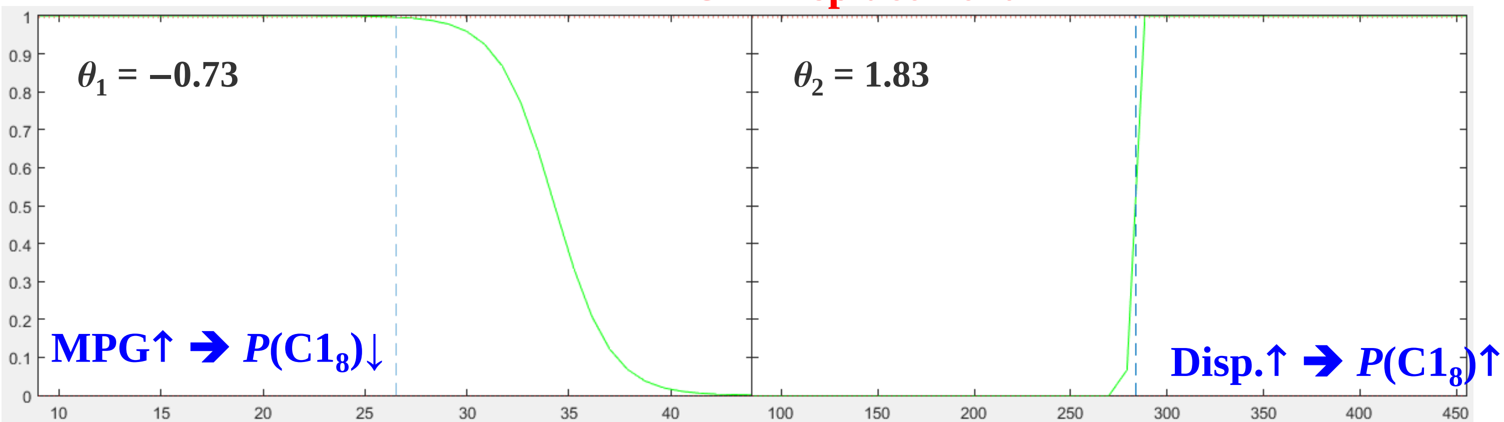
- Displacement is a sensitive predictor in predicting cylinder.

- Class 0
  - Cylinder 4 or 6.
- Class 1
  - Cylinder 8.

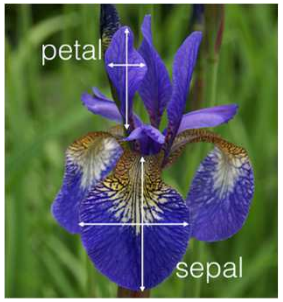




MPG + Displacement

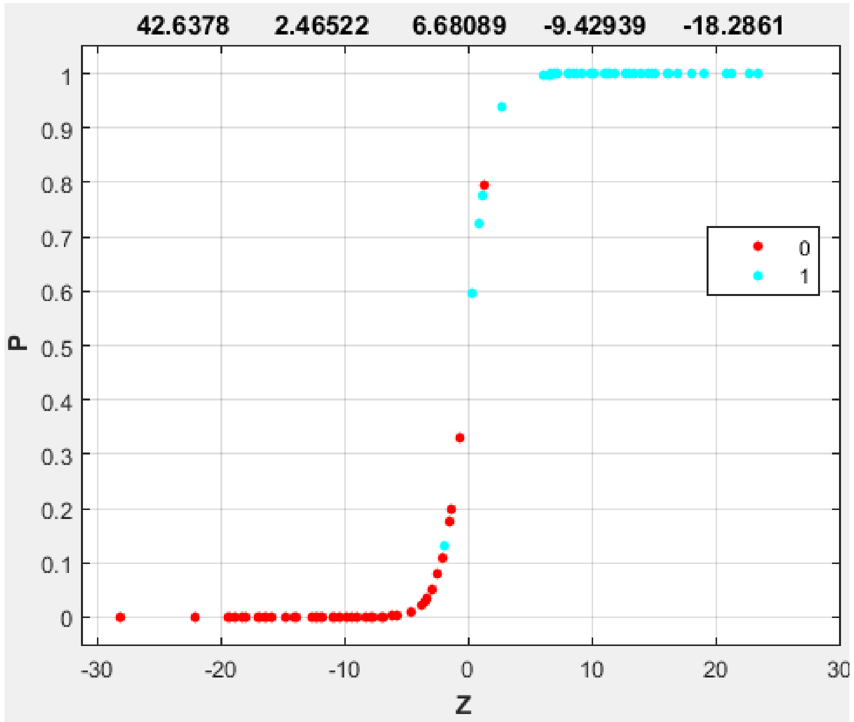


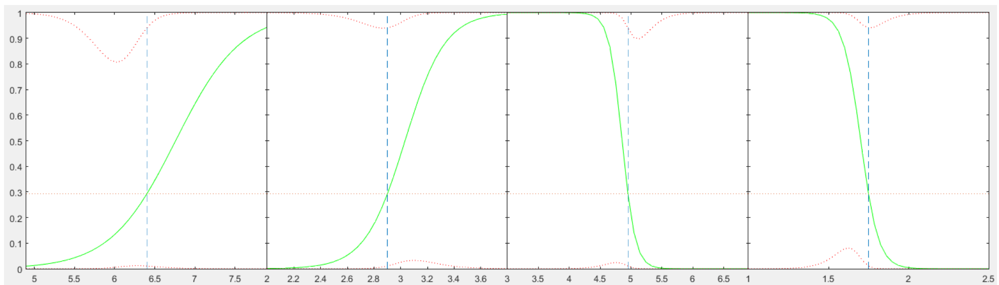
Visualizing High Dimension Data



http://sebastianraschka.com/Articles/2014_python_lda.html

```
load fisheriris
X = meas(51:end,:);    % 100×4, 4-dimension
% Next, create 100×1   BINARY class
Y = strcmp('versicolor',species(51:end));
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
plotSlice(mdl)

p = mdl.Fitted.Response;
Z = mdl.Fitted.LinearPredictor;
figure, gscatter(Z, p, Y);          grid on

zmin = min(Z);     zmax = max(Z);
ylim([-0.05 1.05]), xlabel('\bf Z'),
ylabel('\bf P'),
title(num2str(mdl.Coefficients.Estimate'));
```





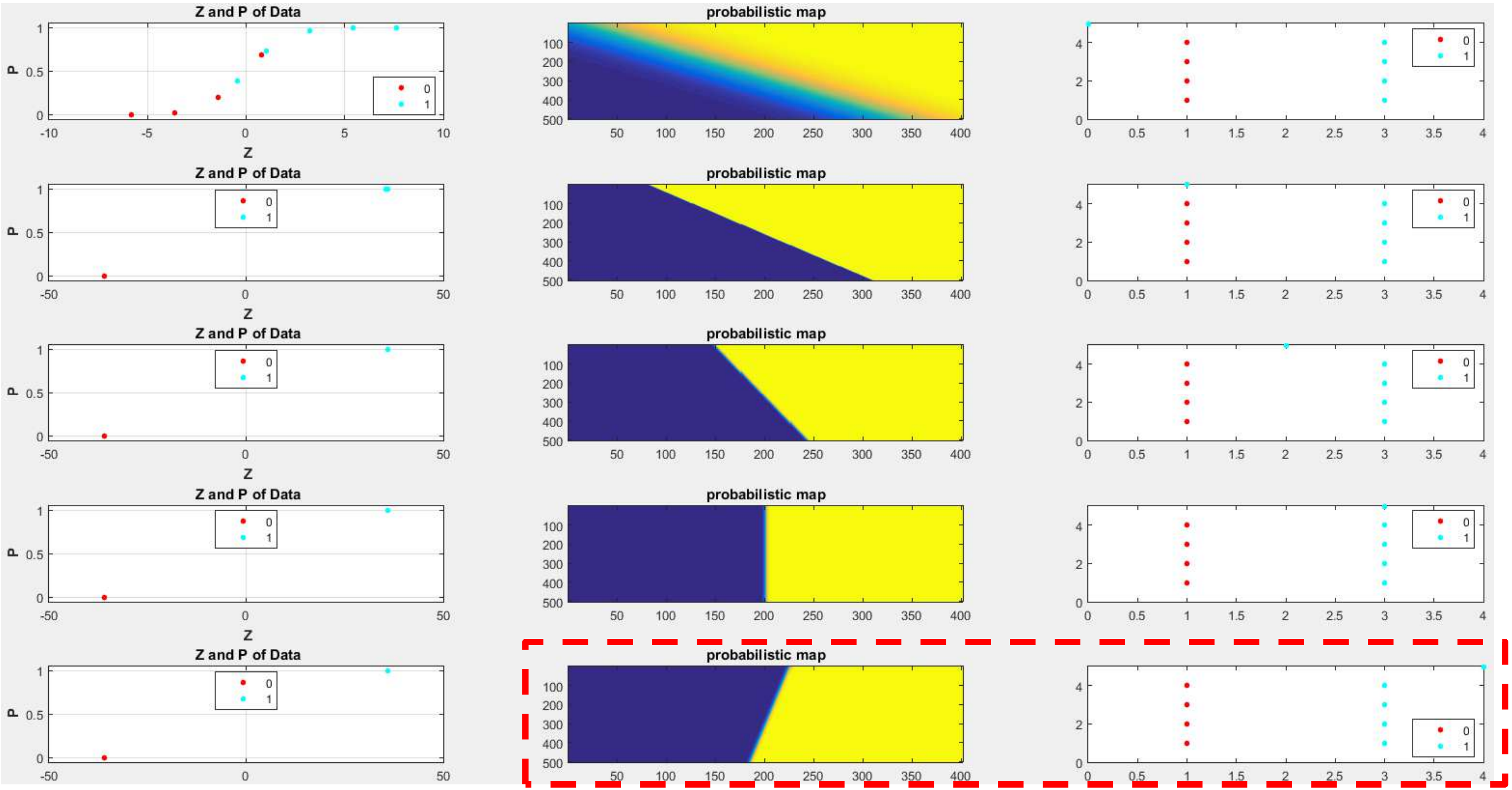
Python Logistic Regression

```
import numpy as np
from sklearn import linear_model, datasets

# import data, and use some
iris = datasets.load_iris()
#X = iris.data[:, :2]         # use all records from first two features.
X = iris.data[50:, :2]        # use records 51 to end from first two features.
#X = iris.data  # use all 4 features.
Y = iris.target;
Y = Y[50:]      # use only records 51 to end, so only 2 classes

# Inverse of regularization strength; smaller C ==> stronger regularization.
# LOSS = C * E + theta,   default C = 1
logreg = linear_model.LogisticRegression()

# fit the data.
logreg.fit(X, Y)
logreg.predict(X)    # predicted classes
```

```
print(logreg.intercept_, logreg.coef_, '\n')
print(logreg.predict(X), '\n')
print(logreg.predict_proba(X) , '\n')
print(logreg.predict_log_proba(X))
```

- In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the ‘multi_class’ option is set to ‘ovr’, and …
  - It uses the cross- entropy loss if the ‘multi_class’ option is set to ‘multinomial’. (Currently the ‘multinomial’ option is supported only by the ‘lbfgs’, ‘sag’ and ‘newton-cg’ solvers.)

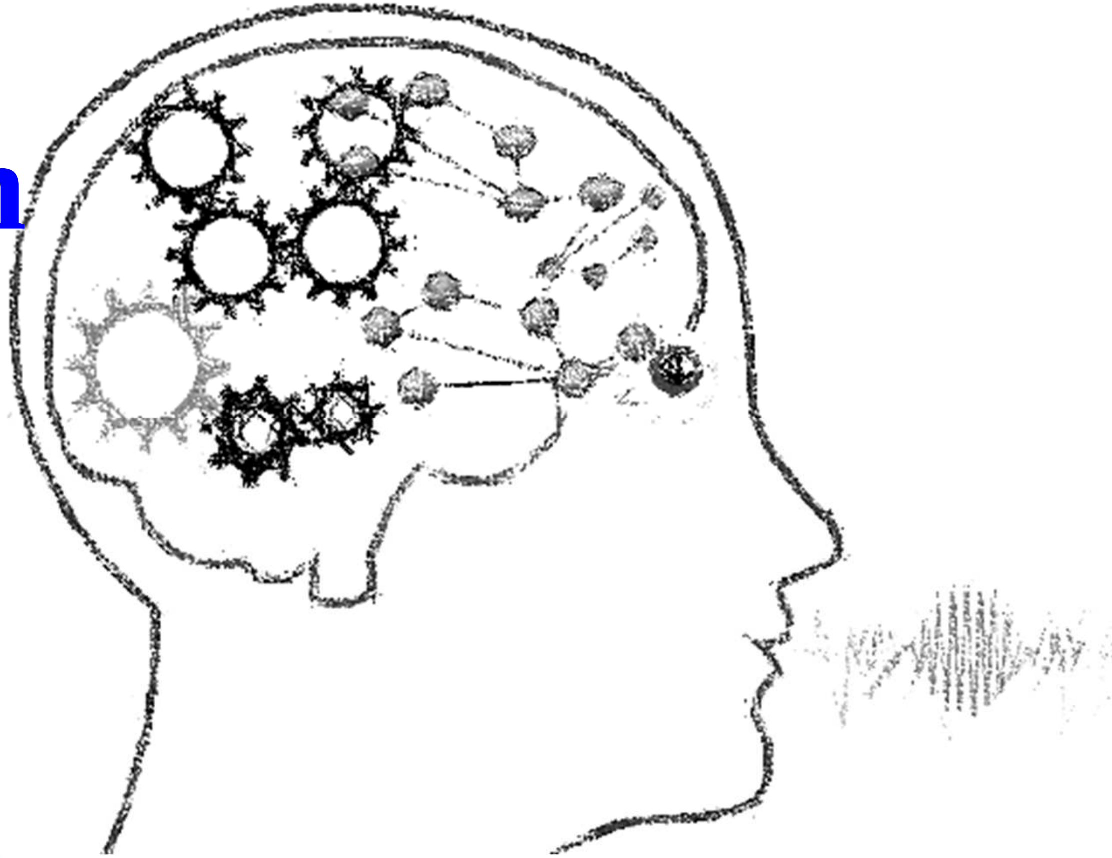
Visualizing Outlier Impacts

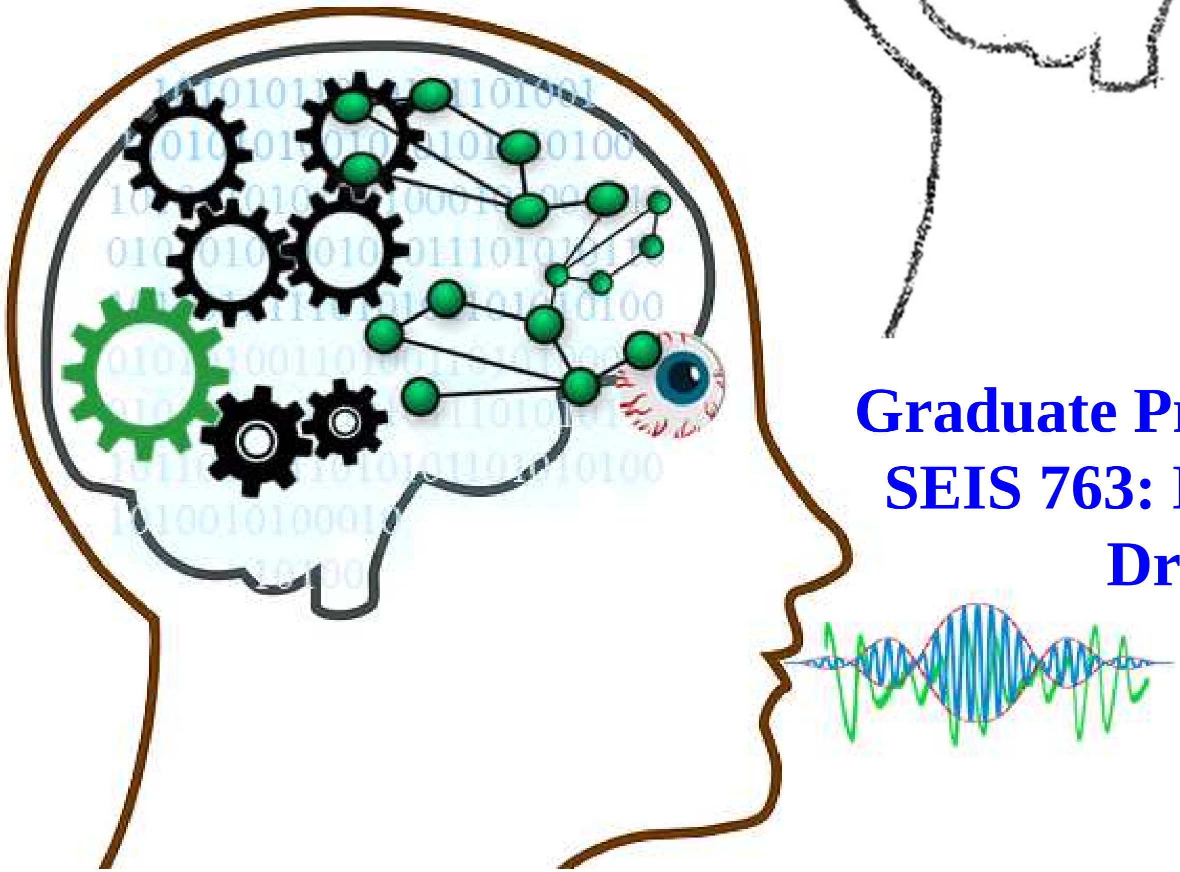


Outline

- How it works? Matlab / Python functions.
- Diagnoses / Visualization.
- **Prediction and Quality.**
- **Outliers.**
- Multiclasses Prediction.
- Nonlinearity / High Dimensionality / Visualization.
- Regularization.
- Cost Function.

Classification Quality





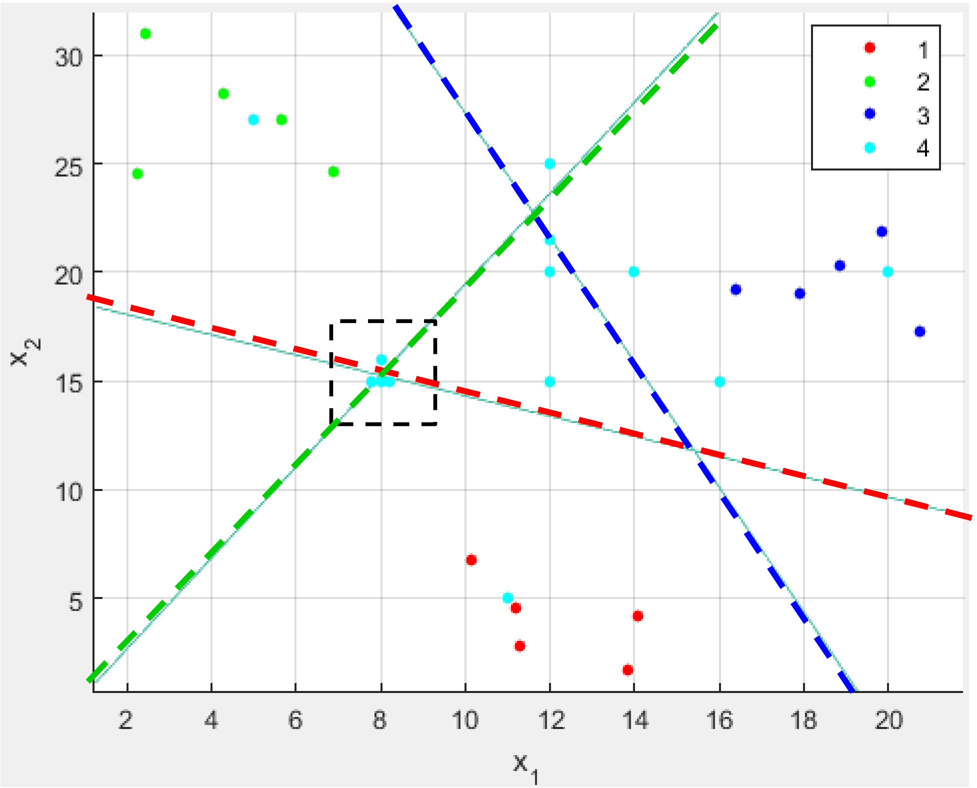
Graduate Program in Software
SEIS 763: Machine Learning
Dr. Chih Lai

Outline

- How it works? Matlab / Python functions.
- Diagnoses / Visualization.
- Prediction and Quality.
- Outliers.
- **Multiclasses Prediction.**
- Nonlinearity / High Dimensionality / Visualization.
- Regularization.
- Cost Function.

One-vs-All (Rest) Prediction

- Train multiple binary models.
  - One for each class ***i*** **against** rest of records of other classes.
  - Feed a test data ***X*** to **EVERY** classifier, predict ***X*** as class ***i*** that has maximum ***P***.
- There can be a region that cannot be classified.

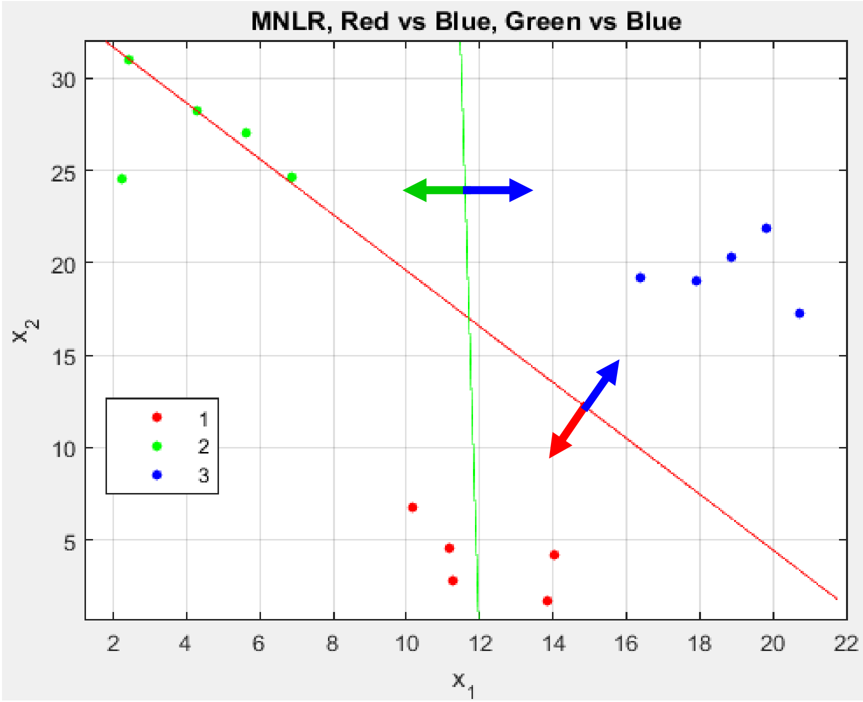


Test Data		Class Probability		
x1	x2	C1	C2	C3
11	5	1	0	0
5	27	0	1	0
20	20	0	0	1
14	20	0	0	1
12	15	0	0	0
12	20	0	0	0
12	21.5000	0	0	0.3800
16	15	0	0	1
12	25	0	1	1
7.8000	15	0.9800	0.7200	0
8	15	0.9500	0.1800	0
8.2000	15	0.8600	0.0200	0
8	16	0	0.9900	0

One vs. Reference

- Multinomial logistic regression, 3 classes.
  - Compute θ for **red** & **green** classes
  - vs the 3rd class (**blue**)

	θ for red vs blue	**θ for green vs blue**
θ_0	56.5740	37.2130
θ_1	-2.4661	-3.1063
θ_2	-1.6311	-0.0493



[B, dev, stats] = **mnrfit**(X, Y);

y_hat = **mnrval**(B, TestData, stats);

linear_model.**LogisticRegression**(C=999)

linear_model.**LogisticRegression**(multi_class='multinomial', solver='newton-cg')

- Matlab functions for ***multinomial*** logistic regression→ **mnrfit()**, **mnrval()**.
  - https://www.mathworks.com/help/stats/mnrfit.html#namevaluepairarguments
  - http://www.mathworks.com/help/stats/mnrval.html
  - http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.predict_proba
  - http://scikit-learn.org/stable/modules/multiclass.html

Multinomial Logistic Regression, Compare to One-vs-All

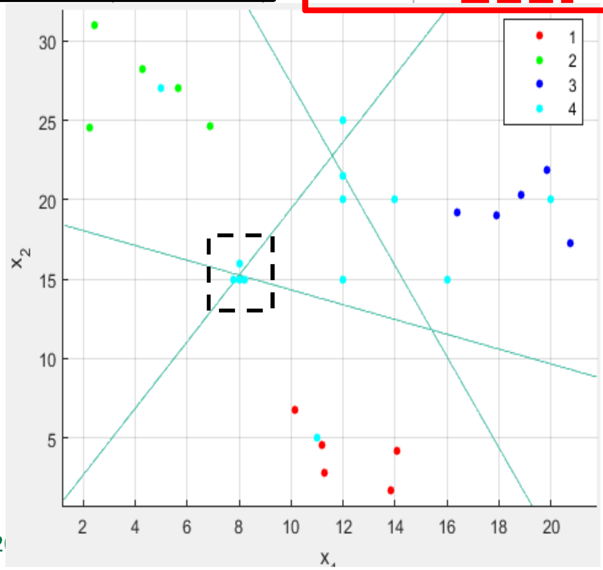
- Not sure how the probability being normalized in the One-vs-Reference method.

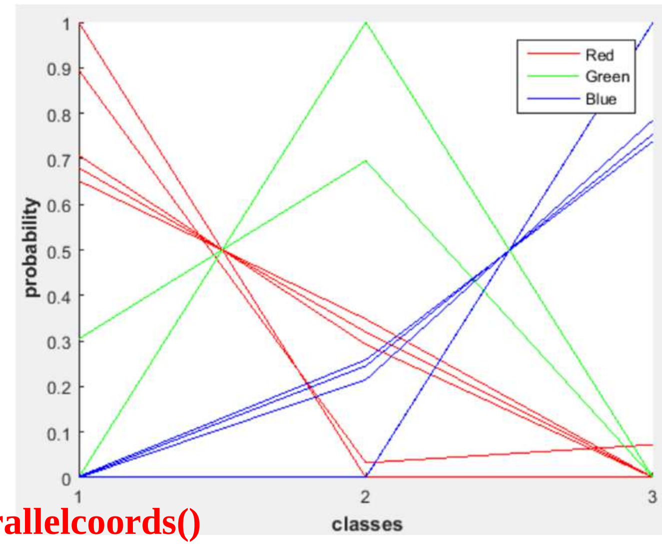
One-vs-All

Test Data x1	Test Data x2	Class Probability C1	Class Probability C2	Class Probability C3
11	5	1	0	0
5	27	0	1	0
20	20	0	0	1
14	20	0	0	1
12	15	0	0	0
12	20	0	0	0
12	21.5000	0	0	0.3800
16	15	0	0	1
12	25	0	1	1
7.8000	15	0.9800	0.7200	0
8	15	0.9500	0.1800	0
8.2000	15	0.8600	0.0200	0
8	16	0	0.9900	0

One-vs-Reference

Test Data x1	Test Data x2	Class Probability C1	Class Probability C2	Class Probability C3
11	5	1	0	0
5	27	0	1	0
20	20	0	0	1
14	20	0	0	1
12	15	0.9000	0.0300	0.0700
12	20	0	0.2600	0.7400
12	21.5000	0	0.2500	0.7500
16	15	0	0	1
12	25	0	0.2100	0.7900
7.8000	15	0.6500	0.3500	0
8	15	0.6800	0.3200	0
8.2000	15	0.7100	0.2900	0
8	16	0.3000	0.7000	0





parallelcoords()

Case Study– Predicting Car Prices

- From many predictors in training cases, want to build a model to predict car $$.
- Is this problem always a regression problem?
- OK, how many classes do we have?

data

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Make	Model	Year	Engine Fu	Engine HP	Engine Cy	Transmiss	Driven_Wh	Number of	Market Ca	Vehicle Si	Vehicle St	highway M	city mpg	Popularity	MSRP
2	BMW	1 Series M	2011	premium u	335	6	MANUAL	rear wheel	2	Factory Tu	Compact	Coupe	26	19	3916	46135
3	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	40650
4	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	36350
5	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	29450
6	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	34500
7	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	31200
8	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	26	17	3916	44100
9	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39300
10	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	36900
11	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	27	18	3916	37200
12	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39600
13	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
14	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	44400
15	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	19	3916	37200
16	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
17	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Convertible	25	18	3916	48250
18	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	43550
19	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000
20	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000

Regular

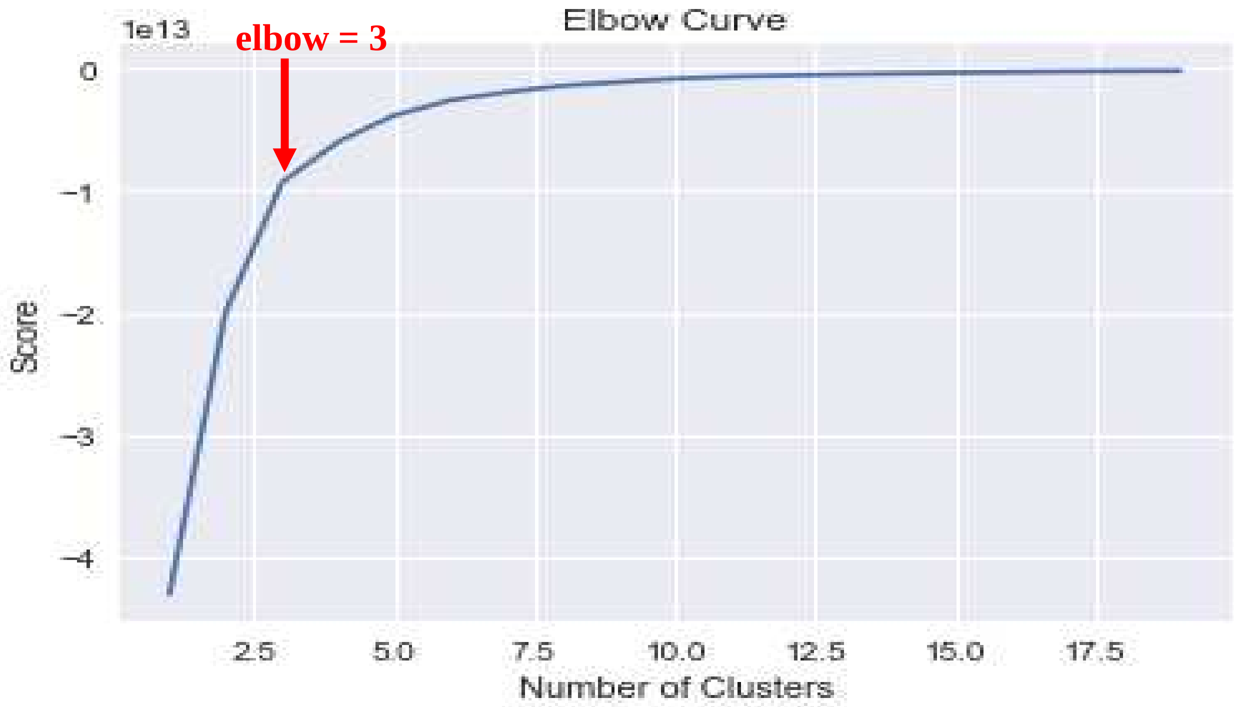
Cheap

Predicting Car Prices, MDL-01 2018 spring

Saleh Alkadayar , Rathana Sorn
Jose Rodriguez, Julie Flater, Gassan Zaid

Deciding # of Classes

- How many classes?
  - Do some kind of clustering (i.e. k-means).
  - How do we do classification on more than 2 classes?



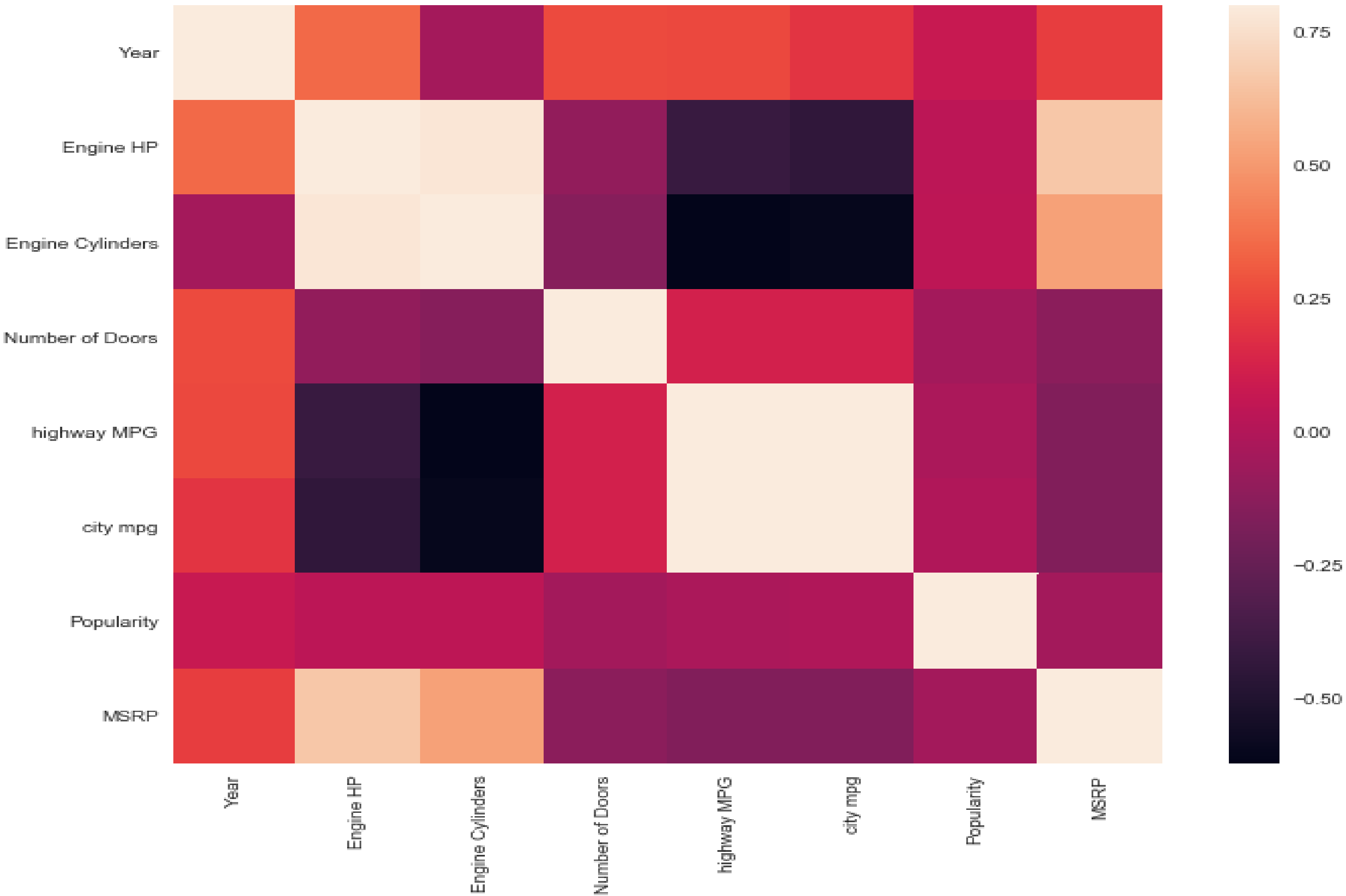
<u>Predicting Car Prices, MDL-01 2018 spring</u>

Saleh Alkadayar , Rathana Sorn
Jose Rodriguez, Julie Flater, Gassan Zaid

Data Leaking Problem??

Predicting Car Prices, MDL-01 2018 spring

Saleh Alkadayar , Rathana Sorn
Jose Rodriguez, Julie Flater, Gassan Zaid



Outline

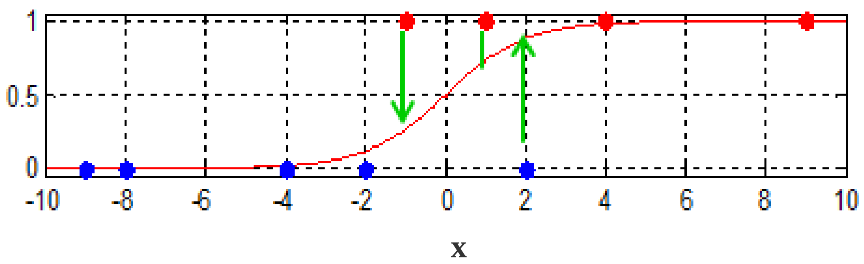
- How it works? Matlab / Python functions.
- Diagnoses / Visualization.
- Prediction and Quality.
- Outliers.
- Multiclasses Prediction.
- Nonlinearity / High Dimensionality / Visualization.
- **Regularization.**
- Cost Function, Other Issues / Examples / Demos.

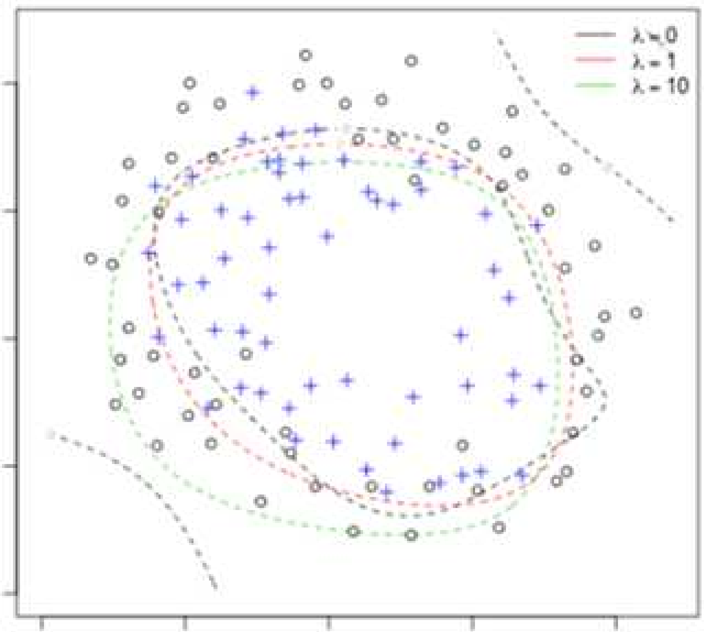
Regularize Logistic Regression

- Regularization.
  - Reduce the number of predictors and identify important ones.
  - Shrink θs, potentially avoid ***overfitting*** problem.

- **<u>Assume</u>** the cost function for logistic regression:
  - $J(\boldsymbol{\theta}) \approx \boldsymbol{E} + \boldsymbol{\lambda}\textbf{Complexity} \approx \boldsymbol{E} + \lambda \sum_{i=1}^{n} |\boldsymbol{\theta}_i| \approx \boldsymbol{P}_{\textbf{joined}} + \boldsymbol{\lambda}\textbf{Complexity}$
  - $\lambda\uparrow$ ➔ simpler model ➔ smoother decision boundary.

$P = {}^{1}/_{1+e^{-z}}$





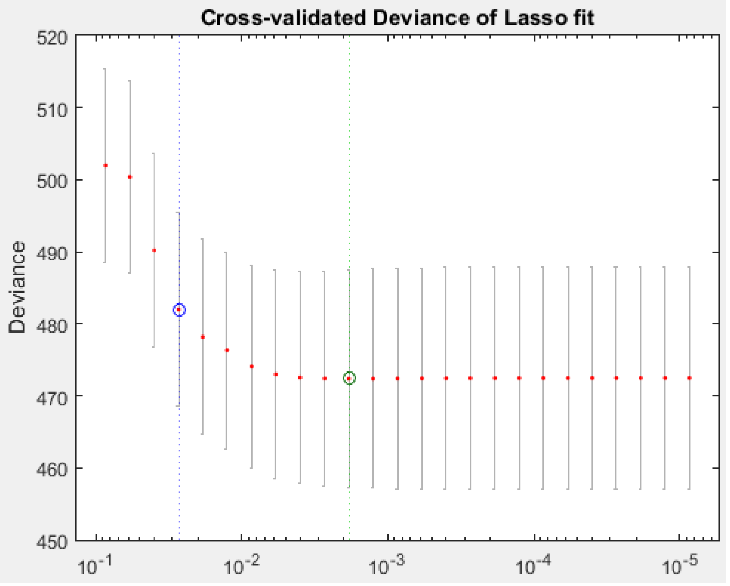
Regularizing Logistic Regression– **lassoglm()**

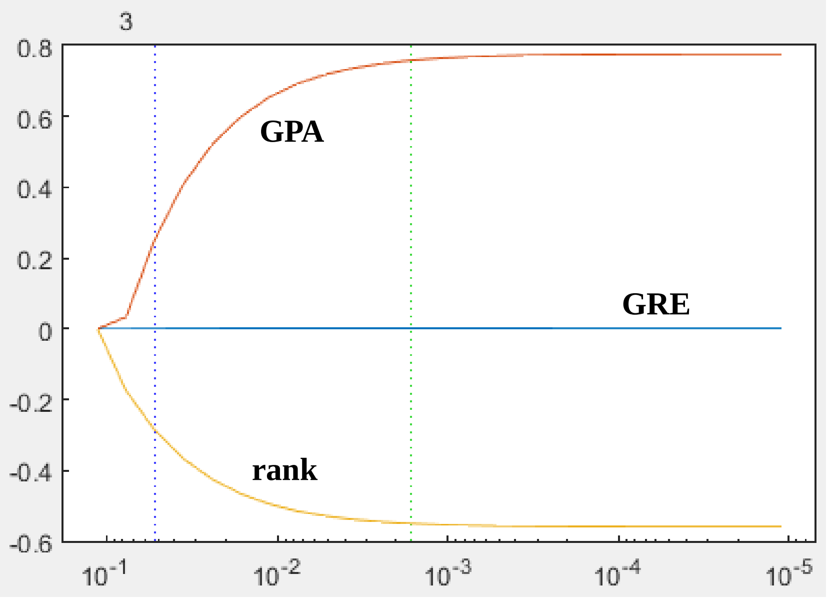
- Graduate school admission prediction from 400 applications.
  - 3 predictors: GRE, GPA, Institute rank
  - Response: admit or not.
  - **Do we need any preprocessing?**

θ
-3.4495
0.0023
0.7770
-0.5600

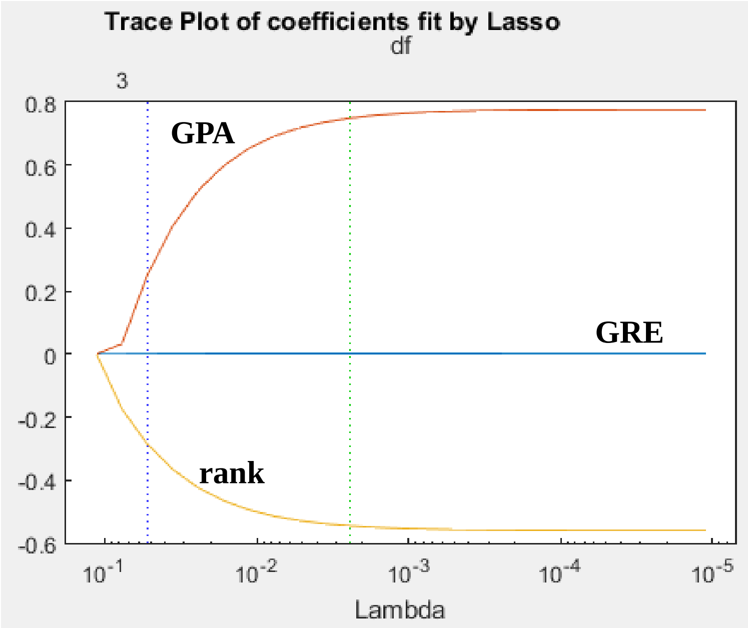
- CFM = (71%)
  - 254 19
  - 97 30

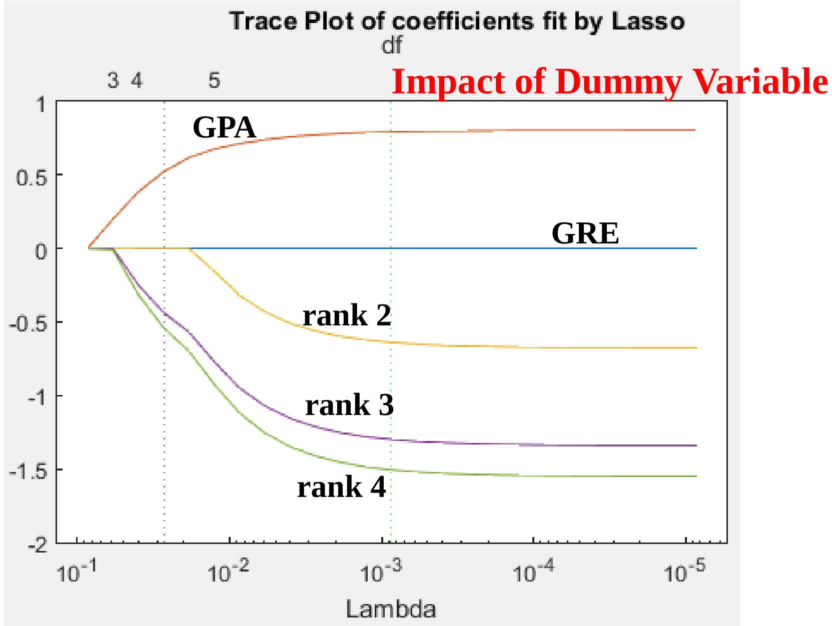
```
[LB, FitInfo] = lassoglm(X, Y, 'binomial', 'NumLambda', 25, 'CV', 10);
lassoPlot(LB, FitInfo, 'PlotType', 'CV');
lassoPlot(LB, FitInfo, 'PlotType', 'Lambda', 'XScale', 'log');
```

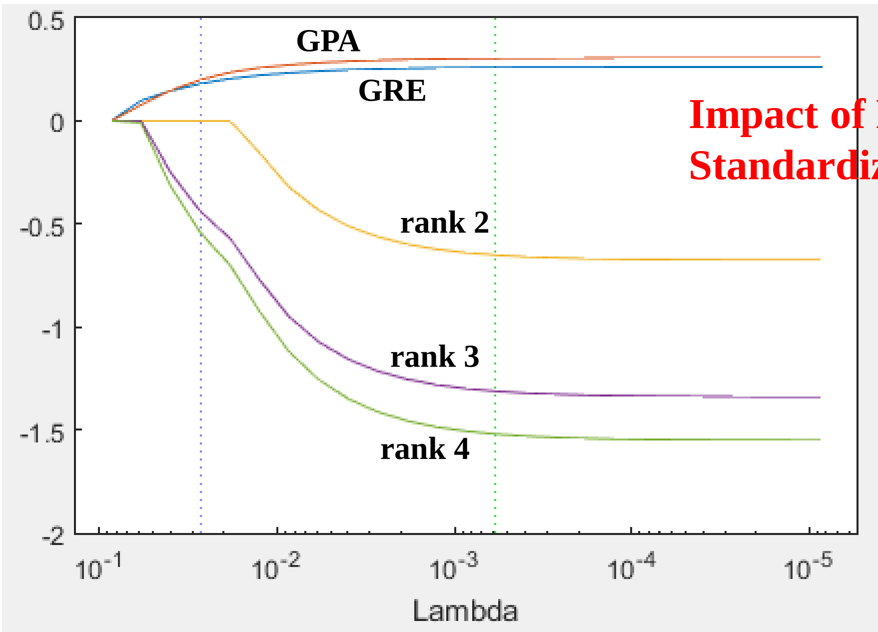




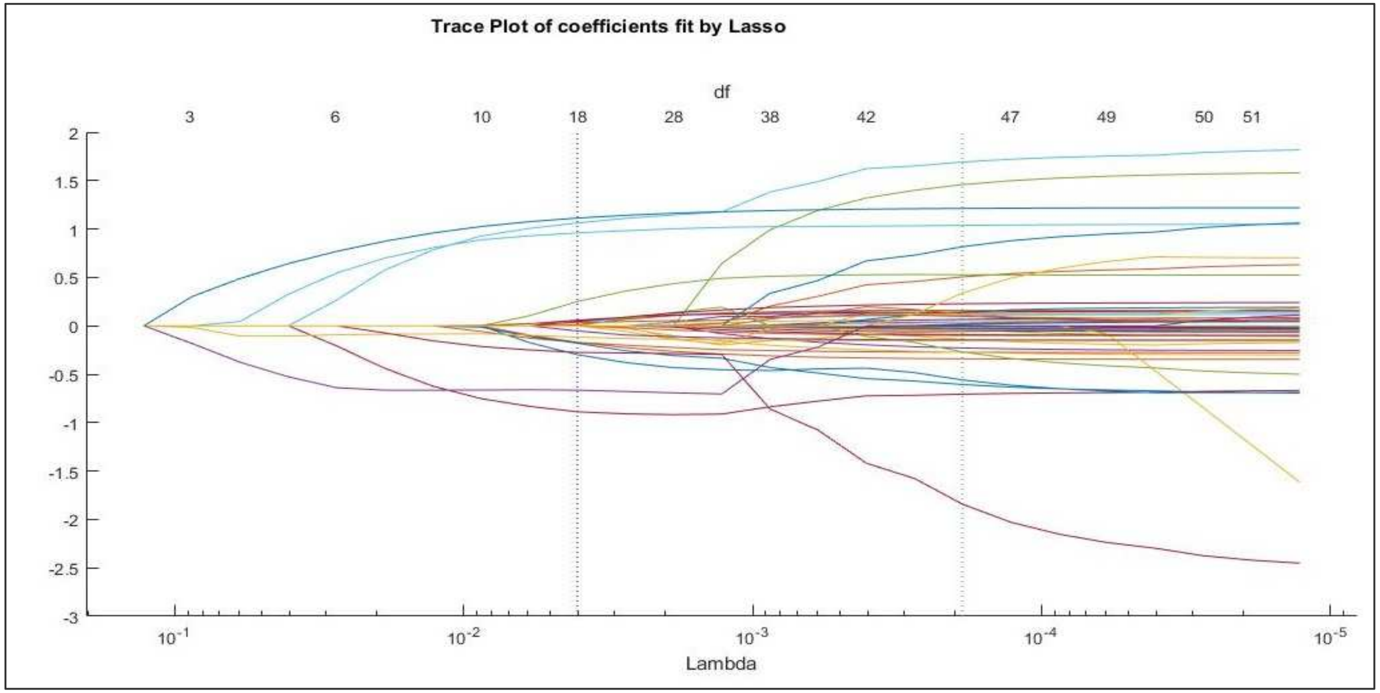
Impact of Dummy Variable







Bank's Marketing Strategy



MDL-01 2018S, Bank Marketing

Terrence White
Ronald E Twite
Leela Sowjanya Chippada
Ahmad K Lubnani
Mowlid Abdillahi
Nathan Adams

Predictor	Description (likely to open an account)
Month - August	Month of campaign
Duration	Contact durations (seconds) (longer better)

Predictor	Description (unlikely to open an account)
Employment Rate	Quarterly employment index
Month - September	Month of campaign

Regularization = Identifying Important Predictors

- Application or implication of “important” predictors?

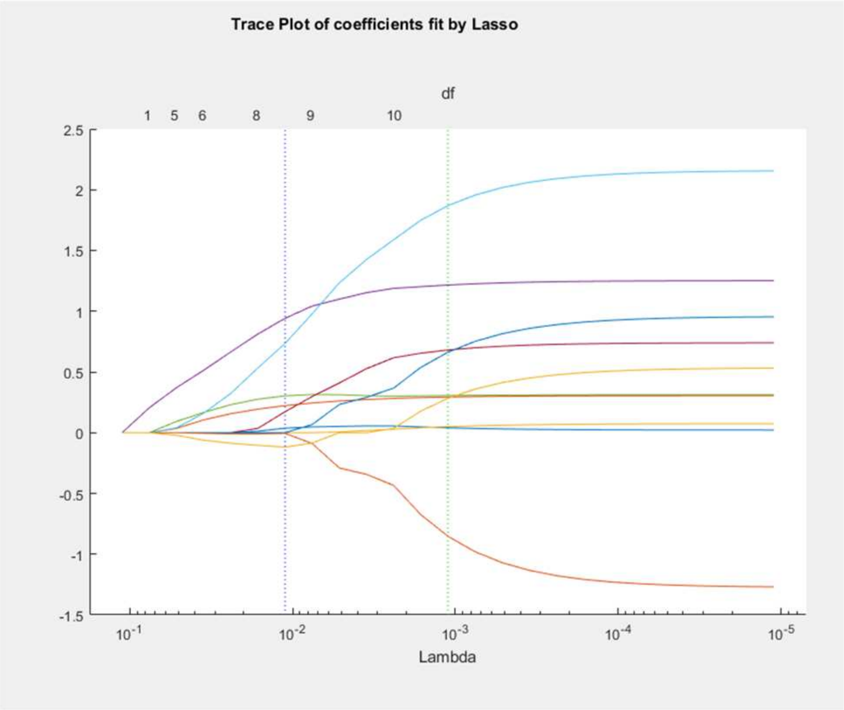
- Couple previous student projects on Chicago or SF crime data…
  - If race is an important predictors in predicting arrest after a traffic stop, does it imply “bias”?

Regularization in the Logistic Regression Model

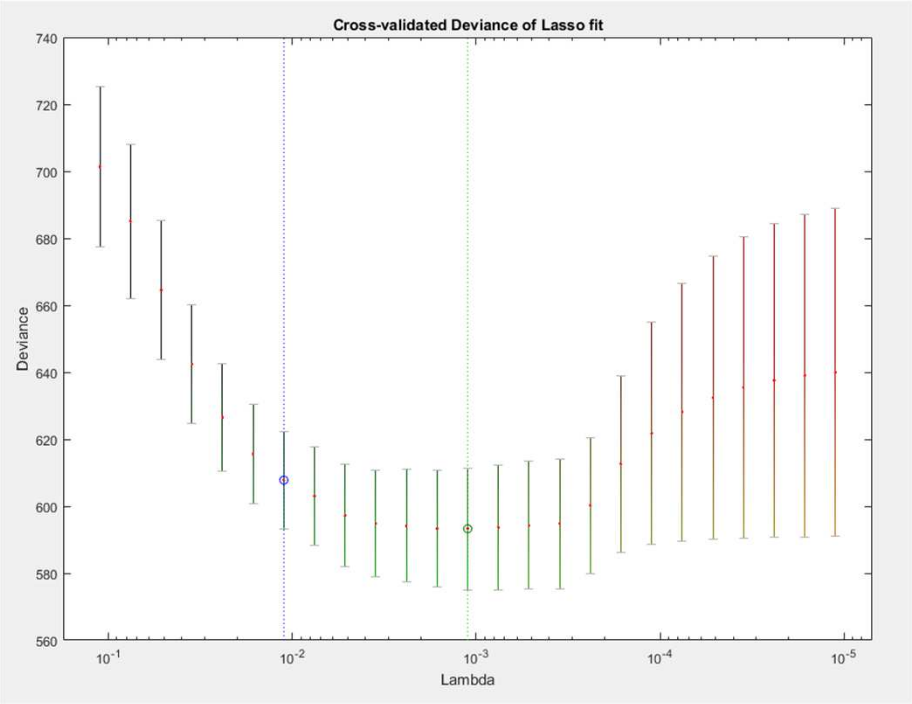
- Any comment?

$$J(\boldsymbol{\theta}) \approx \boldsymbol{E} + \lambda\boldsymbol{\theta}$$

<u>Predicting Liver Disease, MDL-01 2018 S</u>
Abdulaziz Alreshedi,
Beau Birkholz, Matt Conroy,
Adam Grams,
Dorothy Lesher, Don Stryker



Significant reduction between the minimum MSE and 1 standard deviation from the minimum MSE



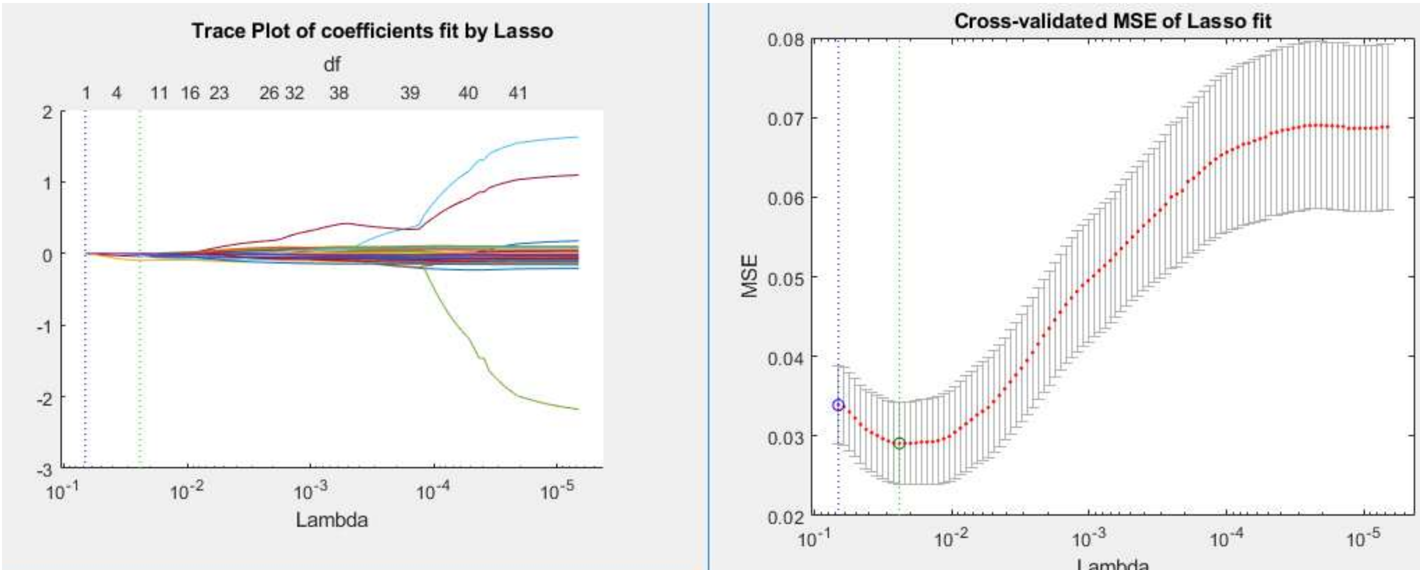
Models built with theta values at the minimum MSE and at 1 standard deviation.

$J(\boldsymbol{\theta}) \approx MSE + \lambda\boldsymbol{\theta}$

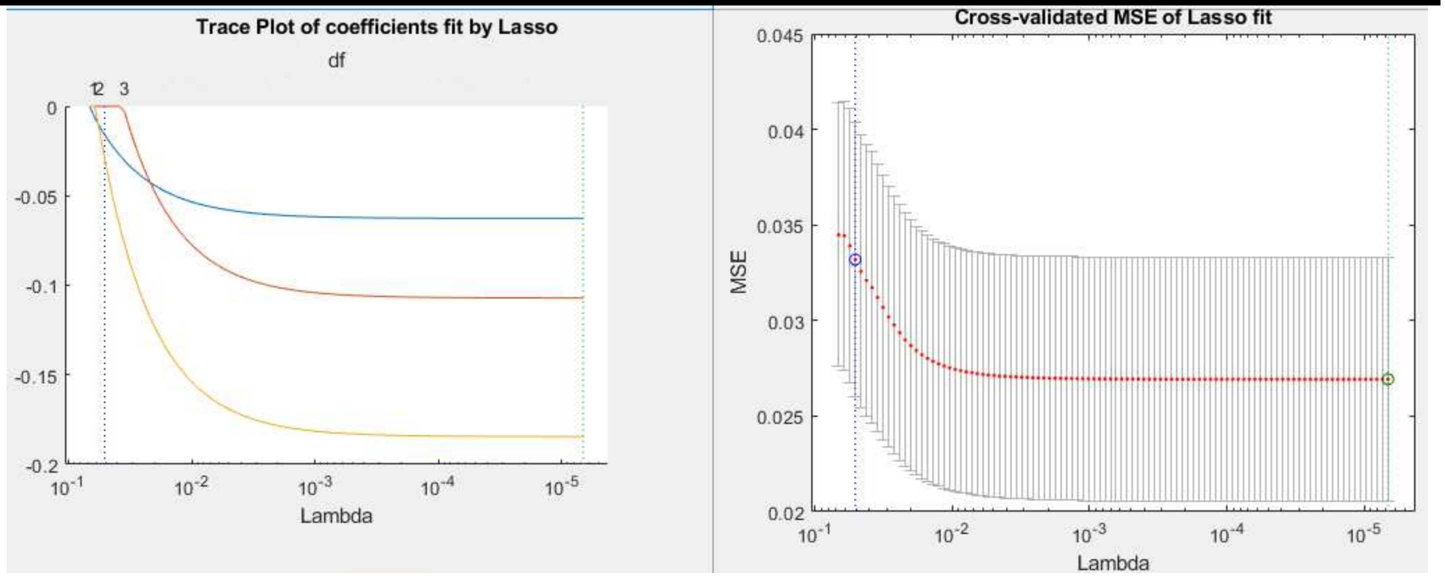
Interpreting This Result??

- 91 records of 53 predictors to predict software readability.

Before Removing Predictors



After Removing Predictors



Regularization “**MAY**” Improve Minority Class Prediction

Lasso Regularization Applied to

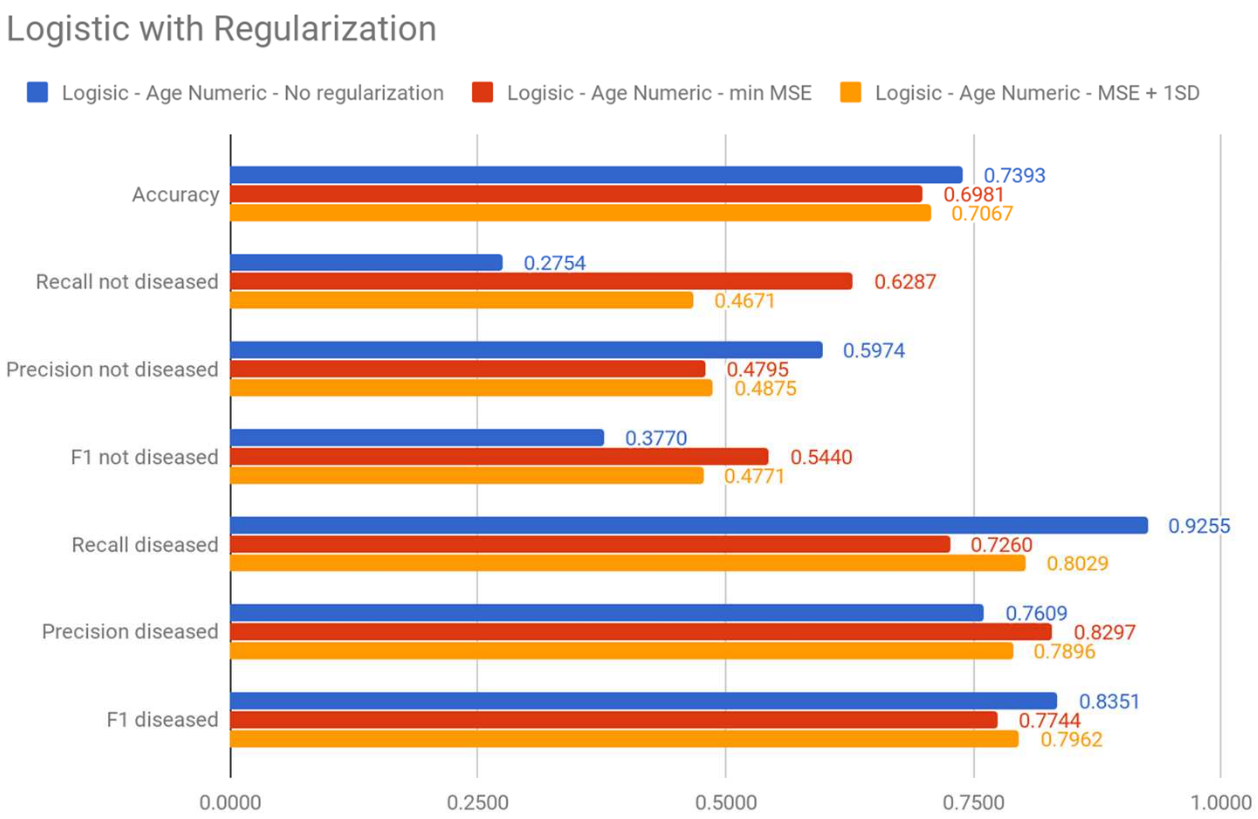
- Minimum MSE
- MSE + 1 SD

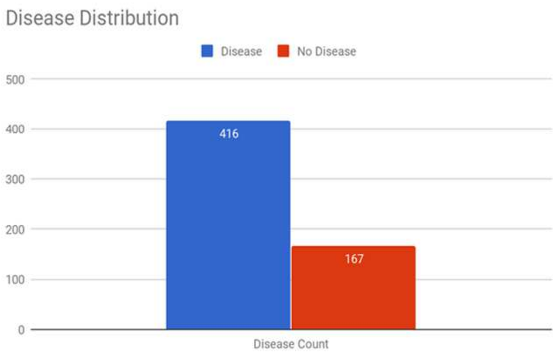
Lasso Regularization Effects

- Not Diseased Class
  - Improved: Recall & F1
  - Reduced: Precision
- Diseased Class
  - Improved: Precision
  - Reduced: Recall & F1

Predicting Liver Disease, MDL-01 2018 S

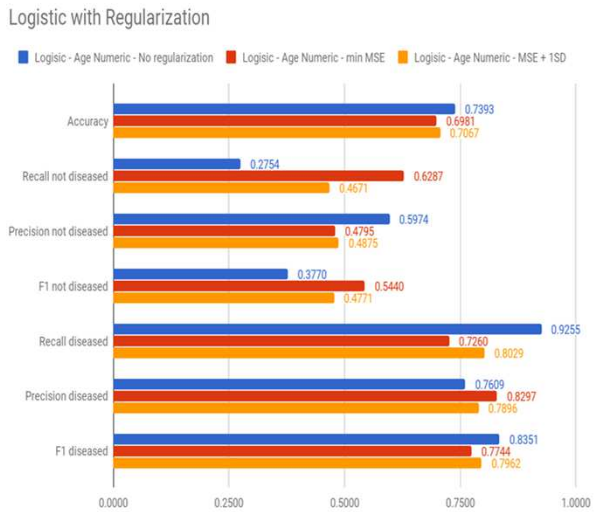
Abdulaziz Alreshedi,
Beau Birkholz,
Matt Conroy,
Adam Grams,
Dorothy Lesher, Don Stryker





Visualizing Quality

- CFM tells us if our dataset has class-balancing issue.
- So, please always include CFM.



Training Dataset						Test Dataset				
CFM				Precision / Recall / F	Target Class	Precision / Recall / F	CFM			
104015	0	366	84	0.9967	**1**	0.9955	26051	0	116	22
0	81030	83	5	0.9957	**(1234)**	0.9947	0	20206	20	5
280	52	71675	10	**0.9962**		**0.9951**	104	22	17818	2
61	58	9	142272				14	13	3	35604
81030	0	83	5	0.9986	**2**	0.9983	20206	0	20	5
0	104015	366	84	0.9989	**(2134)**	0.9988	0	26051	116	22
52	280	71675	10	**0.9988**		**0.9985**	22	104	17818	2
58	61	9	142272				13	14	3	35604
71675	280	52	10	0.9937	**3**	0.9923	17818	104	22	2
366	104015	0	84	0.9953	**(3124)**	0.9929	116	26051	0	22
83	0	81030	5	**0.9945**		**0.9926**	20	0	20206	5
9	61	58	142272				3	14	13	35604
142272	61	58	9	0.9993	**4**	0.9992	35604	14	13	3
84	104015	0	366	0.9991	**(4123)**	0.9992	22	26051	0	116
5	0	81030	83	**0.9992**		**0.9992**	5	0	20206	20
10	280	52	71675				2	104	22	17818

Predicting Liver Disease, MDL-01 2018 S

Abdulaziz Alreshedi,
Beau Birkholz, Matt Conroy,
Adam Grams,
Dorothy Lesher, Don Stryker

Satillite Image Classification

MDL-01, 2018 S

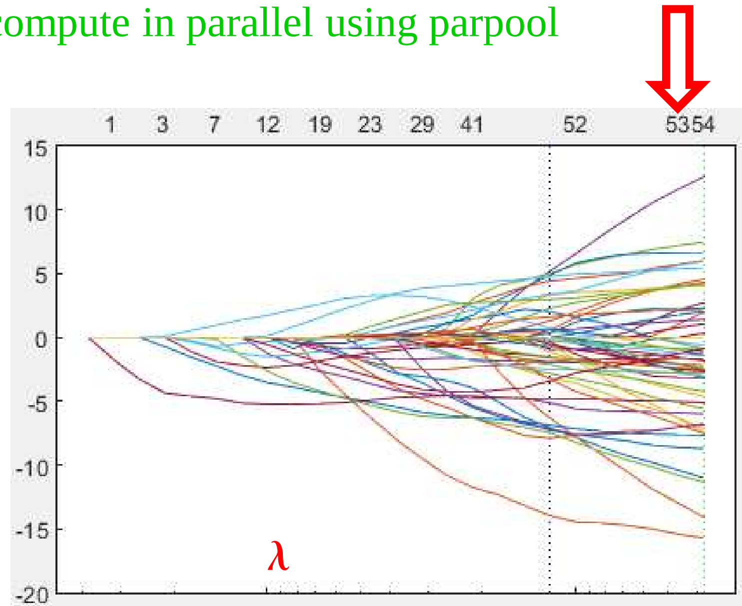
Erin Jacot, Jared Gilbert
Khaled Mahmud
Laura Nicla, Lydia Xu

Regularize Wide Data in Parallel

- Ovarian cancer data. **load ovariancancer**
  - 216 observations and 4,000 predictors.
  - Wide data # predictors >> records.
  - Use **lasso** to screen & select a smaller set of important predictors from wide data.
  - Use **parallel computing** to speed up **cross validation** & **λ testing**.

- Parallel processing using Matlab
  - **opt = statset('UseParallel',true);** % Set options to use parallel computing.
  - parpool() % Prepare to compute in parallel using parpool

- Further time saving:
  - Fewer cross-validation folds.
  - Fewer λ testing.
  - $\alpha\uparrow$ ➔ more lasso than ridge.
    - Remove more predictors. $\frac{(1-\alpha)}{2}\times \boldsymbol{R} + \alpha\times \boldsymbol{L}$



Python Logistic Regression + Regularization

- Logistic regression + regularization
  - Classify 8x8 images of digits into two classes: 0-4 against 5-9.
  - Figures show coefficients of the models for varying *C*.
  - *C* = **Inverse** of regularization strength.
  - http://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_l1_l2_sparsity.html

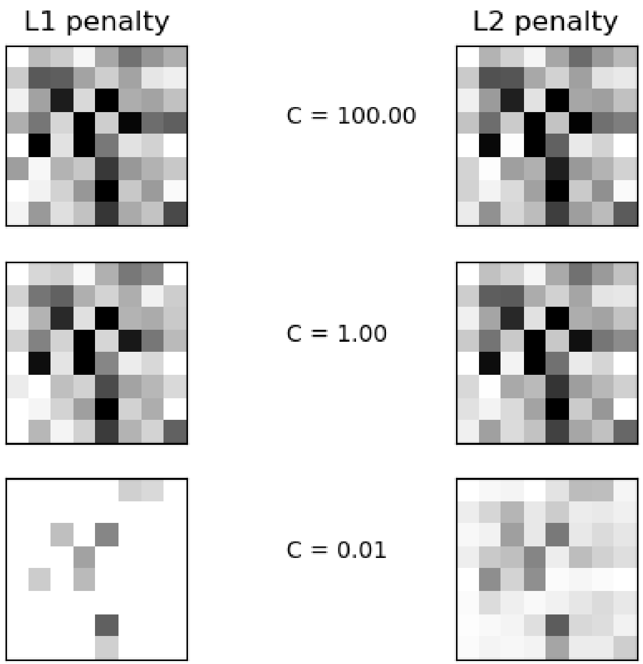
$$J(\theta) \approx E + \lambda \times \theta$$

$$J(\theta) \approx C \times E + \theta$$

$C\uparrow \rightarrow MSE\downarrow \rightarrow \theta\uparrow$

$C\downarrow \rightarrow MSE\uparrow \rightarrow \theta\downarrow$

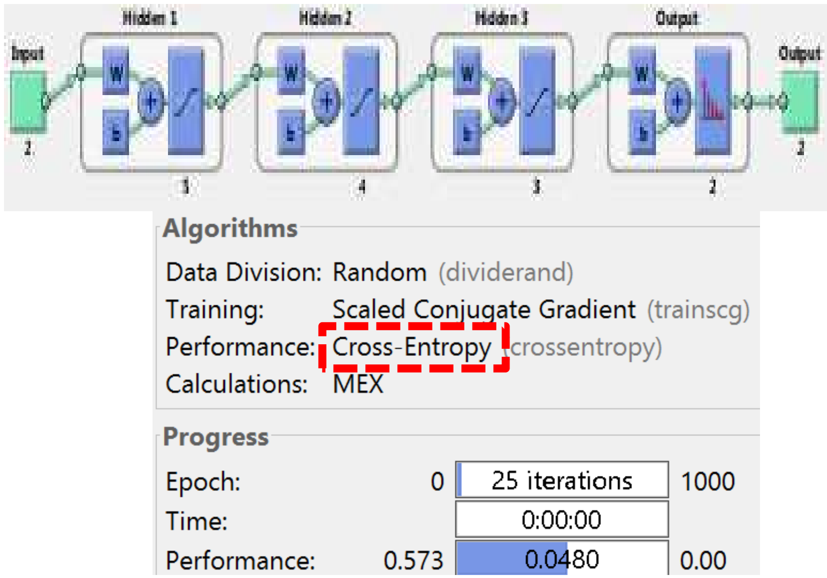
```
# ...
for i, Cx in enumerate((100, 1, 0.01)):
    # turn down tolerance for short training time
    clf_l1_LR = LogisticRegression(C= Cx, penalty = 'l1' , tolerance=0.01)
    clf_l2_LR = LogisticRegression(C= Cx, penalty = 'l2' , tolerance=0.01)
    clf_l1_LR.fit(X, y)
    clf_l2_LR.fit(X, y)

    coef_l1_LR = clf_l1_LR.coef_       # .ravel()
    coef_l2_LR = clf_l2_LR.coef_       # .ravel()
# ...
```



Outline

- How it works? Matlab / Python functions.
- Diagnoses / Visualization.
- Prediction and Quality.
- Outliers.
- Multiclasses Prediction.
- Nonlinearity / High Dimensionality / Visualization.
- Regularization.
- **Cost Function.**

http://bido.com/Auction/difficult.info



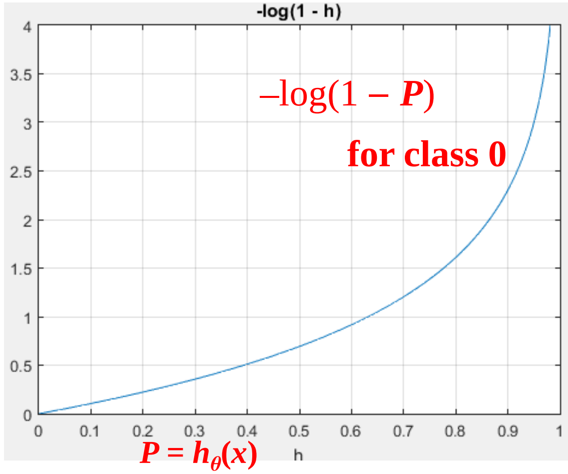
Cost Function? Why Not Just Residual?

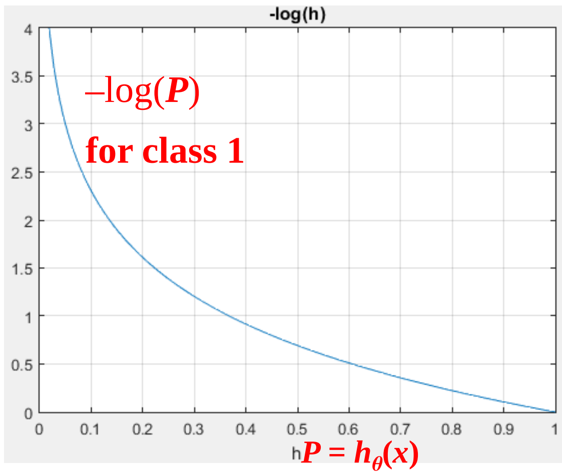
- Residual = (***Y*** – Response) = (***Y*** – Probability)
- For **<u>VERY</u>** wrong prediction ➔ Penalty↑↑

= probability | Y − mdl.Fitted.Response

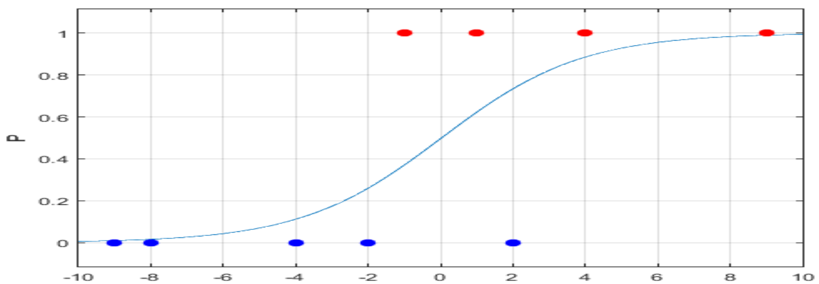
	1 Response	X	Y	1 Raw	Cost
1	0.0094	-9	0	-0.0094	0.0095
2	0.0157	-8	0	-0.0157	0.0158
3	0.1114	-4	0	-0.1114	0.1181
4	0.2601	-2	0	-0.2601	0.3012
5	$\boxed{0.7343}$	$\boxed{2}$	$\boxed{0}$	-0.7343	1.3253
6	$\boxed{0.3705}$	$\boxed{-1}$	$\boxed{1}$	0.6295	0.9929
7	0.6227	1	1	0.3773	0.4737
8	0.8857	4	1	0.1143	0.1214
9	0.9903	9	1	0.0097	0.0098

mdl.Fitted.Response | mdl.Residuals.Raw





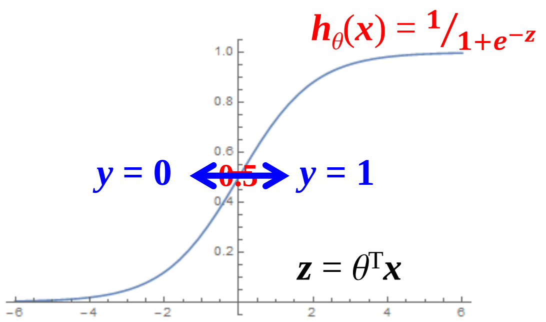

```
figure,
subplot(2, 1, 1),
syms h
ezplot(-log(h),   [0 1   0 4]),
title('y = 1 class')
grid on
subplot(2, 1, 2),
ezplot(-log(1 - h), [0 1   0 4])
title('y = 0 class')
grid on
```

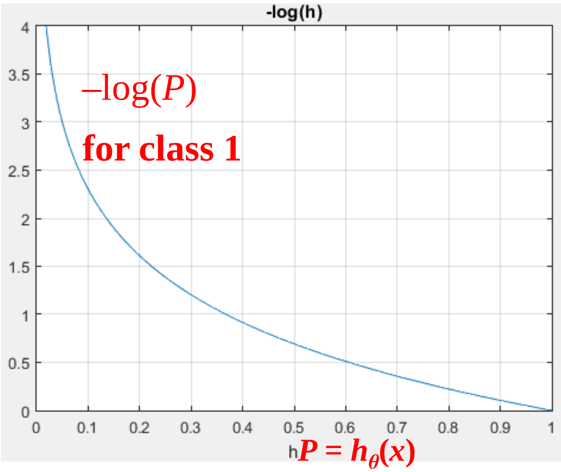


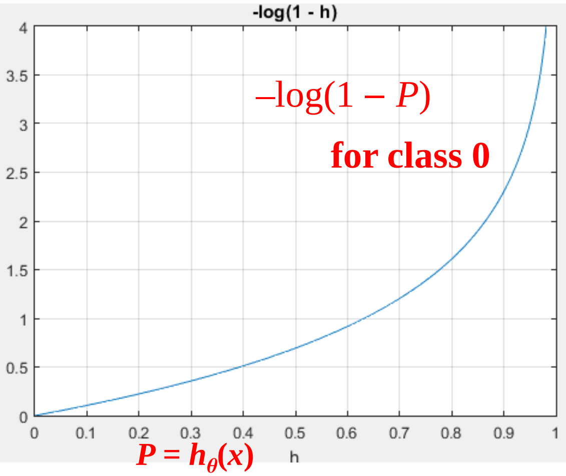
Logistic Regression Cost Function

$\boldsymbol{P} = \boldsymbol{h}_\theta(\boldsymbol{x}) = {}^{1}/_{1+e^{-z}}$

- If $\boldsymbol{y} = 1$ and $\boldsymbol{P} \approx 1$, then $-\log(\boldsymbol{P}) \approx 0$. Correct prediction wo/ penalty.
  - If $\boldsymbol{y} = 1$ and $\boldsymbol{P} \approx 0$, then $-\log(\boldsymbol{P}) \approx \infty$. Error prediction w/ huge penalty.
- If $\boldsymbol{y} = 0$ and $\boldsymbol{P} \approx 0$, then $-\log(1 - \boldsymbol{P}) \approx 0$. Correct prediction wo/ penalty.
  - If $\boldsymbol{y} = 0$ and $\boldsymbol{P} \approx 1$, then $-\log(1 - \boldsymbol{P}) \approx \infty$. Error prediction w/ huge penalty.
- Logistic regression cost function ➔ $\boldsymbol{J}(\theta) = \begin{cases} -\log(P) & if\ \ Y_i = 1 \\ -\log(1-P) & if\ Y_i = 0 \end{cases}$
  - It is **convex**!!!







Logistic Regression Cost Function, **Cross-Entropy**

- Logistic regression cost function ➔ $J(\theta) = \begin{cases} -\log(P) & if\ \ Y_i = 1 \\ -\log(1-P) & if\ Y_i = 0 \end{cases}$

- Put together for $y \in \{0, 1\}$,
  - $J(\theta) = \frac{-1}{m}\sum_{i=1}^{m}[Y_i log(P_i) + (1 - Y_i) log(1 - P_i)\ ]$

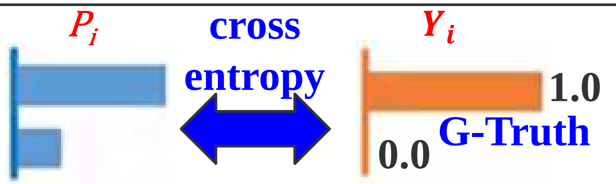
$$\boxed{y \log \frac{1}{1+e^{-\theta^T x}} + (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})}$$

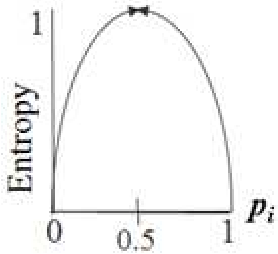
NOTE: $0 \le P \le 1$ ➔ $-\infty \le \log(P) \le 0$.

"cross-entropy" cost function

$Y = 1, P = h(\theta^T x)$

$Y = 0, P = 1 - h(\theta^T x)$





Entropy$(p_1, p_2) = -\ p_1\log_2 p_1 - p_2\log_2 p_2$

- Why this cost function? ***maximum likelihood***.

Why That Cost Function? ➔ Maximum Likelihood

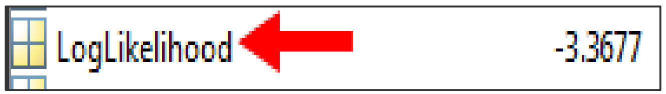
- How do we decide best $\boldsymbol{\theta}$? The answer is…

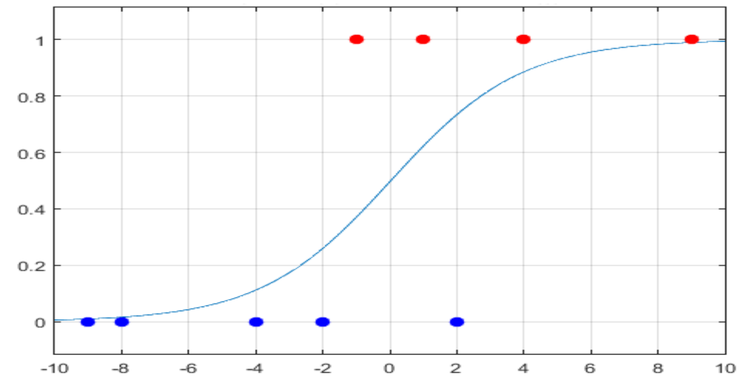
- Select θ to maximize the ***joined probability*** of each prediction.
  - Maximize multiplications of all predicted probabilities.
  - That is, for $Y = 1$, maximize P_i^{Yi}. for $Y = 0$, maximize $(1 - P_i)^{(1 - Yi)}$.
  - Put together, maximize $\boldsymbol{L} = \prod_{i=1}^{m} P_i^{Yi} \times (1 - P_i)^{(1 - Yi)}$. Solve it by set $\frac{\partial L}{\partial P} = 0$.
  - Difficult to solve (chain rule), but we know log(a×b) = log(a) + log(b).
  - = **maximize** ***log-liklihood*** $\log(\boldsymbol{L}) = \log(\prod_{i=1}^{m} P_i^{Yi} \times (1 - P_i)^{(1 - Yi)}) = \sum_{i=1}^{m} log(P_i^{Yi} \times (1 - P_i)^{(1 - Yi)})$

    $= \sum_{i=1}^{m}[Y_i log P_i + (1 - Y_i) log(1 - P_i)]$

$$\boxed{y \log \frac{1}{1+e^{-\theta^T x}} + (1-y)\log(1 - \frac{1}{1+e^{-\theta^T x}})}$$

NOTE: $0 \le P \le 1$ ➔ $-\infty \le \log(P) \le 0$.





- **Same as to minimize** ***negative log likelihood***

$$\boldsymbol{\frac{-1}{m}\sum_{i=1}^{m}[Y_i log(P_i) + (1 - Y_i) log(1 - P_i)]}$$

Maximize Likelihood = Minimize Negative Log-Likelihood

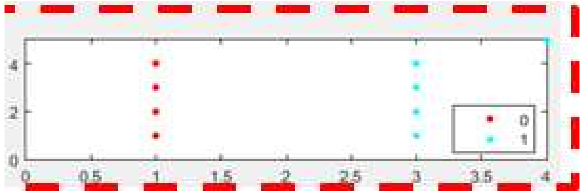
- **Maximize joined probability➔ Maximize Log Likelihood➔ Minimize Negative Log Likelihood**.
  - Access it from the model returned in Matlab:
    - mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
    - mdl.**LogLikelihood** **% this is a log likelihood, NOT negative log likelihood**
    - To **compare performance** between different logistic models.

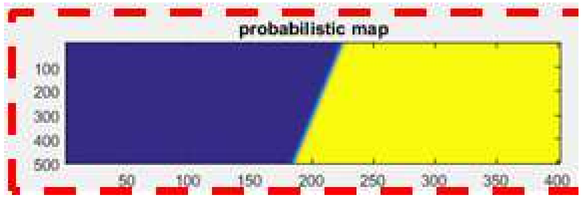
See Appendix for detail matlab code

$$\boxed{y\log\frac{1}{1+e^{-\theta^T x}}+(1-y)\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)}$$

NOTE: $0 \le P \le 1$ ➔ $-\infty \le \log(P) \le 0$.

```
red = [ -1 1; 1 1; 4 1; 9 1];              % class 1
blue =[-9 0; -8 0; -4 0; -2 0; 2 0];       % class 0
X = [blue(:, 1); red(:, 1);];    Y = [blue(:, 2); red(:, 2)];
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
P = mdl.Fitted. LogLikelihood;
-sum(Y.*log(P)+(1-Y).*log(1-P))      % NLL = 3.3677
```





$$\frac{d}{dt}\left(\log\left(\frac{1}{1+e^{-t}}\right)\right)=\frac{1}{e^t+1}$$

$$\frac{\partial}{\partial t}\left(\log\left(\frac{1}{1+e^{-t\,x}}\right)\right)=\frac{x}{e^{t\,x}+1}$$

Alternate forms:

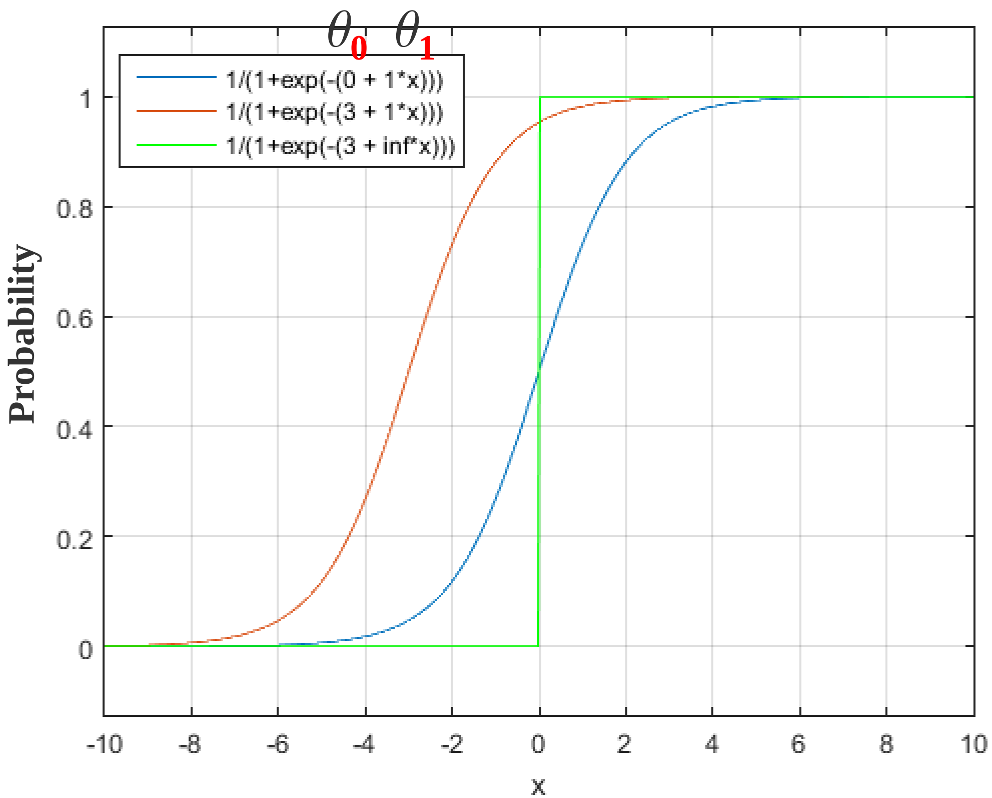
$$\frac{d}{dt}\left(\log\left(1-\frac{1}{1+e^{-t}}\right)\right)=-\frac{e^t}{e^t+1} \qquad x\left(\frac{1}{e^{t\,x}+1}-1\right)$$

$$\frac{\partial}{\partial t}\left(\log\left(1-\frac{1}{1+e^{-t\,x}}\right)\right)=-\frac{x\,e^{t\,x}}{e^{t\,x}+1} \qquad \frac{x}{e^{t\,x}+1}-x$$

Appendix

Shifting in Logistic Function – Changing θ_0

- Logistic Regression $\quad \boldsymbol{h}_\theta(\boldsymbol{x}) = \boldsymbol{h}(\theta^T\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{z}) = {}^1/_{1+e^{-z}}$
  - $\boldsymbol{z} = \theta^T\boldsymbol{x} = \theta_0\times\boldsymbol{x}_0 + \theta_1\times\boldsymbol{x}_1 + \ldots + \theta_n\times\boldsymbol{x}_n.$ $\quad \boldsymbol{z} = \theta^T\boldsymbol{x} = \theta_0\times\boldsymbol{x}_0 + \theta_1\times\boldsymbol{x}_1.$
  - What will be the curve if we set $\theta_1 \approx \infty$?

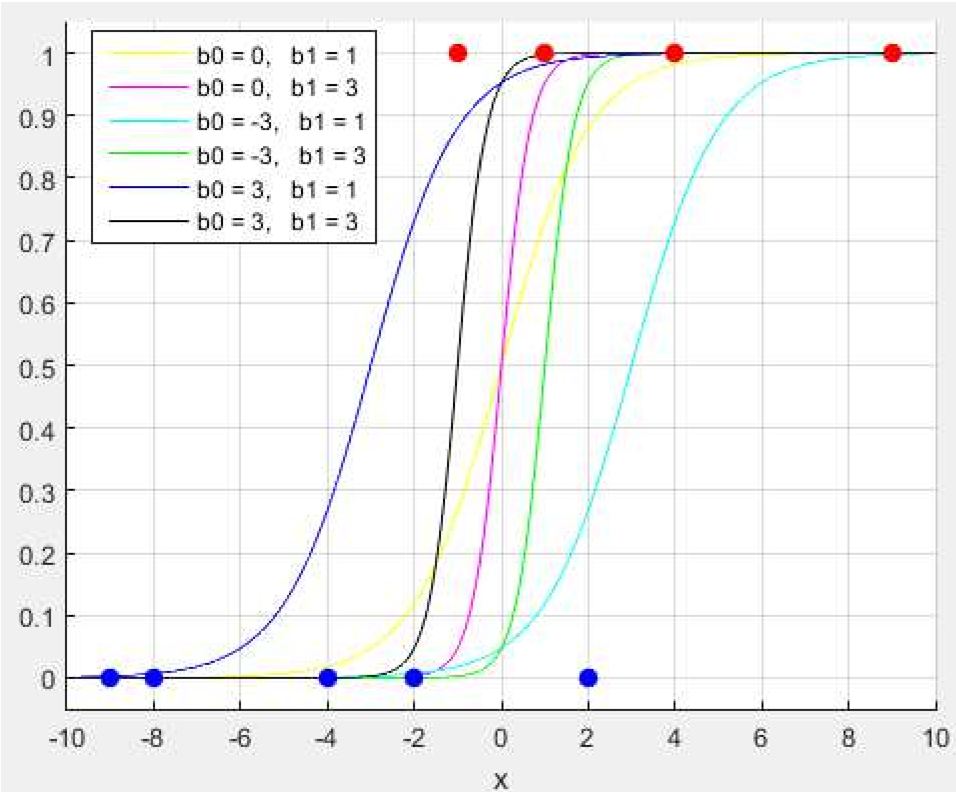



```
syms x
fstr1 = '1/(1+exp(-(0 + 1 * x)))';
fstr2 = '1/(1+exp(-(3 + 1 * x)))';
fstr3 = '1/(1+exp(-(3 + inf * x)))';

ezplot(fstr1, [-10 10]);
hold on
ezplot(fstr2, [-10 10]);
h3 = ezplot(fstr3, [-10 10]);
set(h3, 'color', 'g')
hold off
grid on
legend({fstr1, fstr2, fstr3})
title('')
```

Logistic Function– Changing Both θ_0 and θ_1

- Logistic (sigmoid) function σ takes LR output ($\hat{y}$) as input & convert to ***P* [0..1]**.

$\theta_0 \times x_0 + \theta_1 \times x_1$ $\quad$ $b_0 \times x_0 + b_1 \times x_1$




```
syms x
b0_arr = [0 -3 3]; b1_arr = [1  3];
fstr = '1/(1+exp(-(b0 + b1*x)))';
c = ['ymcgbk'];
figure, hold on
loops = 0;  legendStr = [];
for b0 = b0_arr
    for b1 = b1_arr
        fstrX = strrep(fstr, 'b0', num2str(b0));
        fstrX = strrep(fstrX, 'b1', num2str(b1));
        h = ezplot(fstrX, [-10 10]);
        loops = loops + 1;
        set(h, 'color', c(loops));
        legendStr{loops} = ['b0 = ' num2str(b0) ',   b1 = ' num2str(b1)];
    end
end
legend(legendStr),   title(''),    grid on,     ylim([-0.05 1.05])

red = [-1 1; 1 1; 4 1; 9 1];
blue =[2 0; -2 0; -4 0; -8 0; -9 0];
data = [red; blue];
Y = [zeros(size(red, 1), 1); ones(size(blue, 1), 1)];
gscatter(data(:, 1), data(:, 2), Y, 'rb', '..', 25, 'off')
hold off
```

Binomial Distribution

- A **binomial experiment** has the following properties:
  - The experiment consists of *n* repeated trials of two outcomes: a success & a failure.
  - The probability of success, denoted by *P*, is the same on every trial.
  - The trials are independent: one trial does not affect the outcome on other trials.
  - http://stattrek.com/probability-distributions/binomial.aspx
- Binomial experiment Example: flip a coin *N* times and count the # of heads.
  - Repeated trials of flipping a coin *N* times with outcomes - heads or tails: 0.5 .
  - The trials are independent; getting heads on one trial does not affect results of others.

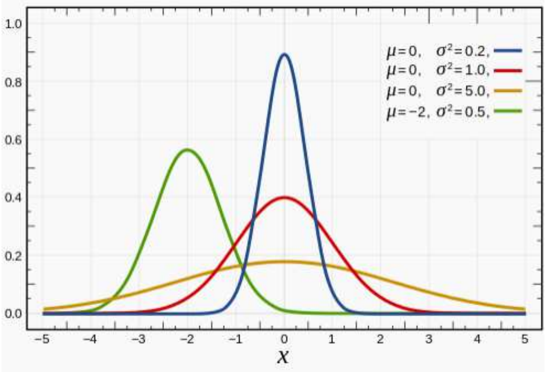
Different Distributions

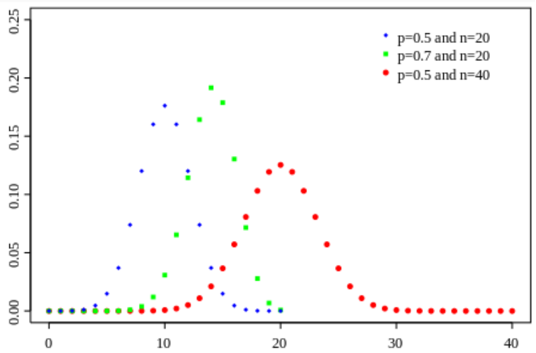
- http://www.mathworks.com/help/stats/generalizedlinearmodel-class.html

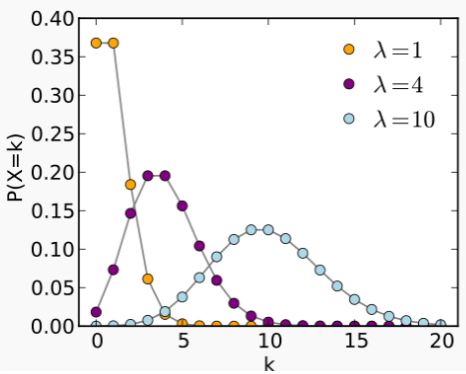
$$h_\theta(x) = P = {}^{e^z}/_{1+e^z} = {}^{1}/_{1+e^{-z}}$$

$$\frac{P}{1-P} = e^z.$$

Distribution	Link Function Name	Link Function	Mean (Inverse) Function
'normal'	'identity'	$f(\mu) = \mu$	$\mu = Xb$
'binomial'	'logit'	$f(\mu) = \log(\mu/(1-\mu))$	$\mu = \exp(Xb) / (1 + \exp(Xb))$
'poisson'	'log'	$f(\mu) = \log(\mu)$	$\mu = \exp(Xb)$
'gamma'	-1	$f(\mu) = 1/\mu$	$\mu = 1/(Xb)$
'inverse gaussian'	-2	$f(\mu) = 1/\mu^2$	$\mu = (Xb)^{-1/2}$

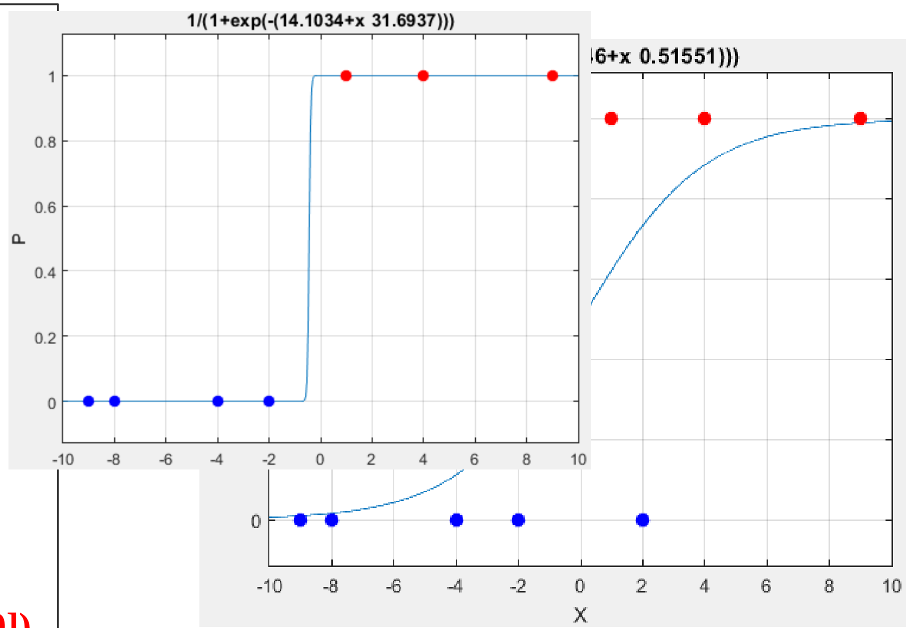






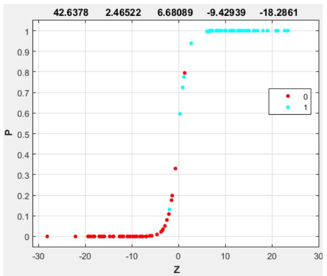
Plotting *X* against Response (Probability = $^{1}/_{1+e^{-z}}$)

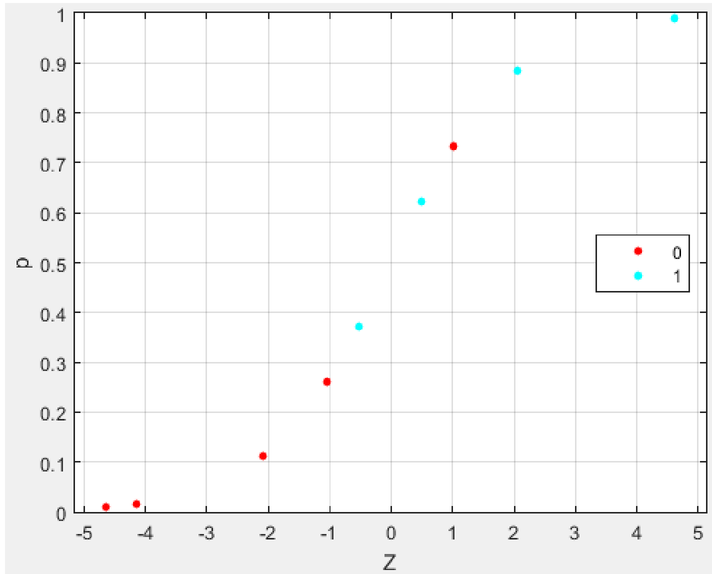
```
red = [ -1 1; 1 1; 4 1; 9 1];
blue =[-9 0; -8 0; -4 0; -2 0; 2 0];
X = [blue(:, 1); red(:, 1);];
Y = [blue(:, 2); red(:, 2)];

mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')
B = mdl.Coefficients.Estimate;          %% θ
Bstr = num2str(B);
Zstr = [Bstr(1,:) '+x*' Bstr(2,:)];     %% = θ^T x
syms x,
figure, ezplot(['1/(1+exp(-('   Zstr   ')))'], [-10 10])
hold on,     gscatter(X(:, 1), Y, Y, 'br', '..', 25, 'off')
hold off,    grid on
xlabel('X'), ylabel('P')
mdl.LogLikelihood
```



for 1-D

```
%% Plot Z wrt P (much easier)
p = mdl.Fitted.Response;
Z = mdl.Fitted.LinearPredictor;
figure, gscatter(Z, p, Y);  grid on
```





for H-D

Some Matlab **Link Functions**

- mdl = **fitglm**(X, Y, 'distr', 'binomial', **'link'**, **'logit'**)

Link Function Name	Link Function	Mean (Inverse) Function
'log'	$f(\mu) = \log(\mu)$	$\mu = \exp(Xb)$
'logit'	$f(\mu) = \log(\mu/(1-\mu))$	$\mu = \exp(Xb) / (1 + \exp(Xb))$
'probit'	$f(\mu) = \Phi^{-1}(\mu)$	$\mu = \Phi(Xb)$
'comploglog'	$f(\mu) = \log(-\log(1-\mu))$	$\mu = 1 - \exp(-\exp(Xb))$
'reciprocal'	$f(\mu) = 1/\mu$	$\mu = 1/(Xb)$

$\boldsymbol{e^z/_{1+e^z} = {}^1/_{1+e^{-z}}}$

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

normal distribution function

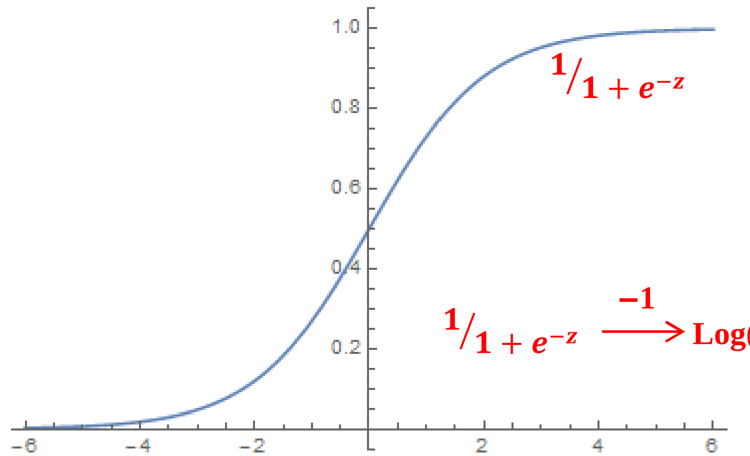
- **Logit function** $\log(\frac{P}{1-P}) = z = \theta^{\mathrm{T}}\boldsymbol{x}$.
  - **Inverse of the sigmoidal "logistic" function**.
  - It gives the log-odds (log of odds).
  - $\frac{P}{1-P}$ is refer to as ***odds***.
  - $\frac{P}{1-P} = e^z \Rightarrow P = e^z - e^z P \Rightarrow P = {}^{e^z}/_{1+e^z} = {}^1/_{1+e^{-z}} = h_\theta(x)$.

Logit Function (Visualize in Mathematica)

```
f[x_] := 1 / (1 + Exp[-x])
Plot[f[x], {x, -6, 6}]
```

$1/1+e^{-z}$

Out[110]=




```
f[x_]:=1/(1+Exp[-x])
Plot[f[x], {x, -6, 6}]

InverseFunction[f][x]

fi[x_]:=Log[x/(1-x)]
Plot[fi[x], {x, 0, 1}]

InverseFunction[fi][x]
```

Inverse

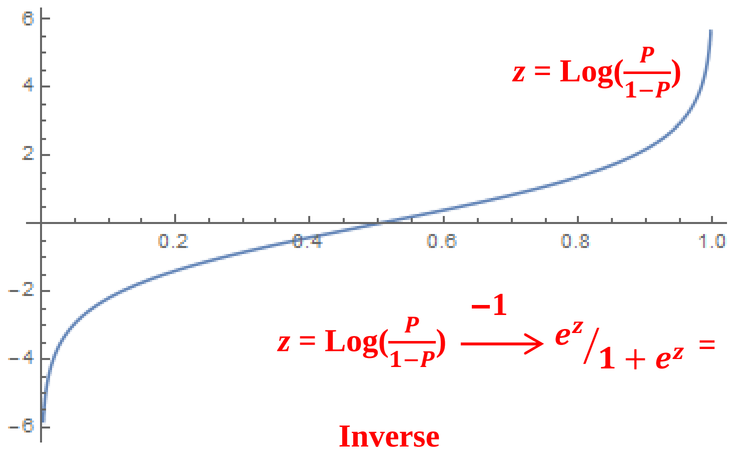
Out[111]= ConditionalExpression$\left[2\, i\, \pi\, C[1] + \boxed{\mathrm{Log}\left[-\frac{x}{-1+x}\right]},\ C[1] \in \text{Integers}\right]$

```
InverseFunction[f][x]
```

$\mathrm{Log}(\frac{P}{1-P})$

```
fi[x_] := Log[x / (1 - x)]
Plot[fi[x], {x, 0, 1}]

InverseFunction[fi][x]
```

Out[113]=



Inverse

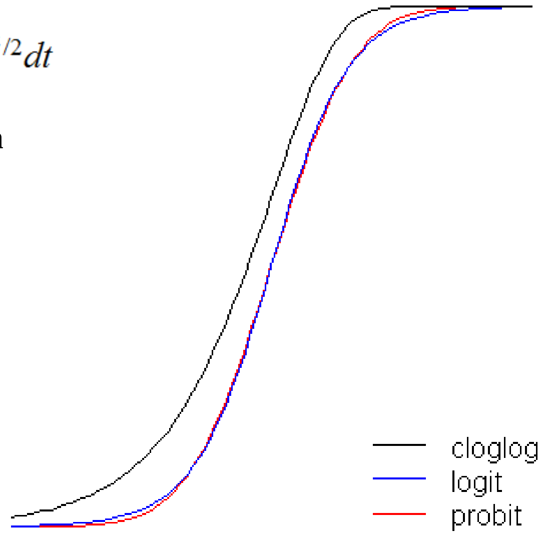
Out[114]= ConditionalExpression$\left[\boxed{\frac{e^x}{1+e^x}},\ -\pi < \mathrm{Im}[x] \le \pi\right]$

Logit and Probit

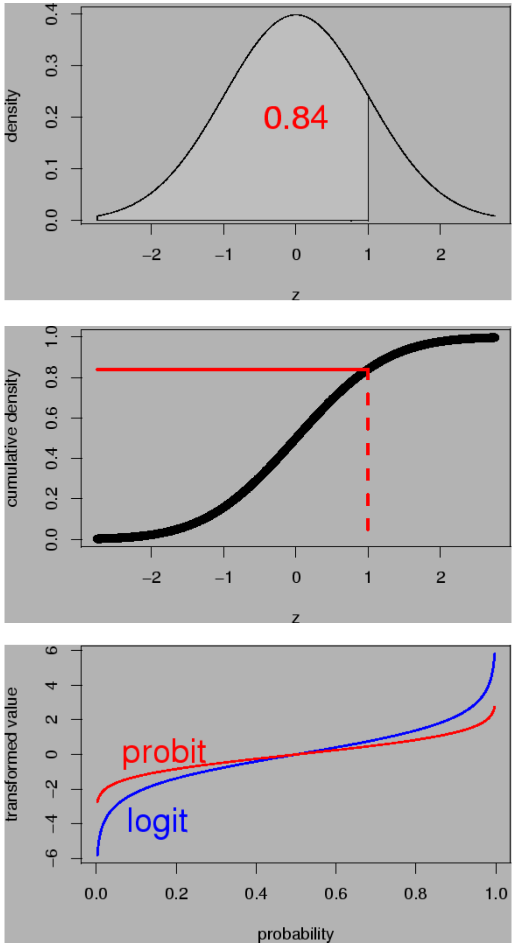
- The probit function is the inverse of the standard cumulative normal distribution.
- Logit and Probit generate very similar curves.
  - Logit has slightly fatter tails than probit, making it slightly more 'robust'.
  - For 2 classes, undifferentiated between logit & probit.
  - For 3+ classes, harder to apply probit.
  - Use **likelihood** value to decide logit or probit.

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

normal distribution function



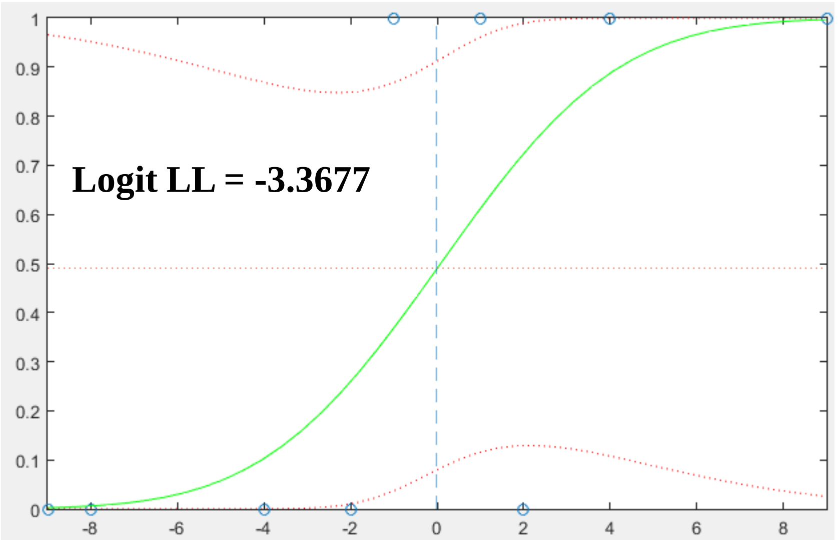
http://stats.stackexchange.com/questions/20523/difference-between-logit-and-probit-models

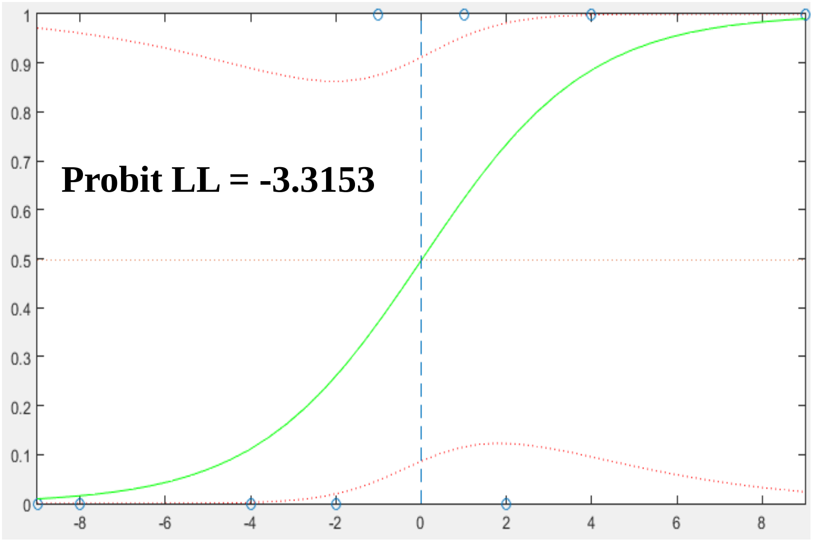


http://pages.uoregon.edu/aarong/teaching/G4075_Outline/node18.html

Logit and Probit Comparison, Likelihood

- Logit has slightly longer tails than probit ➔ logit is slightly more robust.
- In binary response, probit & logit are largely the same.
- Use **log likelihood** (larger ➔ better) value to decide logit or probit.





red = [**-1 1**; 1 1; 4 1; 9 1];
blue =[-9 0; -8 0; -4 0; -2 0; **2 0**];
X = [blue(:, 1); red(:, 1);]; Y = [blue(:, 2); red(:, 2)];

mdl = fitglm(X, Y, 'distr', 'binomial', 'link', '**logit**')
plotSlice(mdl)
mdl.LogLikelihood % **-3.3677** equal to following
sum(Y.*log(mdl.Fitted.Response) + (1-Y).*log(1 - mdl.Fitted.Response))

red = [**-1 1**; 1 1; 4 1; 9 1];
blue =[-9 0; -8 0; -4 0; -2 0; **2 0**];
X = [blue(:, 1); red(:, 1);]; Y = [blue(:, 2); red(:, 2)];

mdl = fitglm(X, Y, 'distr', 'binomial', 'link', '**probit**')
plotSlice(mdl)
mdl.LogLikelihood % **-3.3153**

Total Cost = –1 × Log Likelihood

$$y \log \frac{1}{1+e^{-\theta^T x}} + (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$$

- $\text{Cost}(\boldsymbol{h}(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{x}), \boldsymbol{y}) = \frac{-1}{m}\sum_{i=1}^{m}[Y_i log(P_i) + (1 - Y_i)log(1 - P_i)]$
  - Total cost = **–1** * **mdl.LogLikelihood** (***negative log likelihood***)

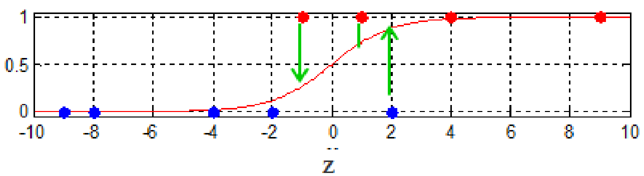
LogLikelihood	-3.3677

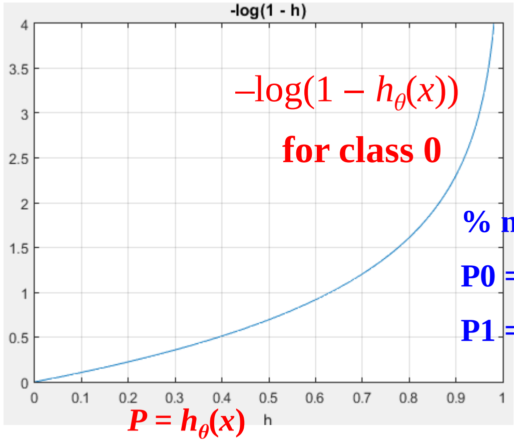
```
C0 = log(1 - mdl.Fitted.Response(1:5))
C1 = log(mdl.Fitted.Response(6:9))
-1 * sum([C0;C1])          % Total Cost = 3.367
```

= probability   **Y − mdl.Fitted.Response**

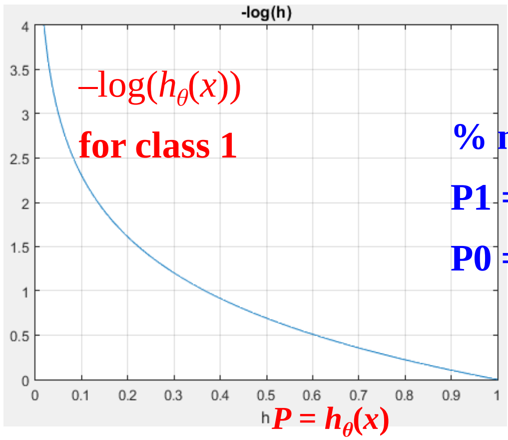
	Response	X	Y	Raw	Cost
1	0.0094	-9	0	-0.0094	0.0095
2	0.0157	-8	0	-0.0157	0.0158
3	0.1114	-4	0	-0.1114	0.1181
4	0.2601	-2	0	-0.2601	0.3012
5	0.7343	2	0	-0.7343	1.3253
6	0.3705	-1	1	0.6295	0.9929
7	0.6227	1	1	0.3773	0.4737
8	0.8857	4	1	0.1143	0.1214
9	0.9903	9	1	0.0097	0.0098

mdl.Fitted.Response   **mdl.Residuals.Raw**





% need P0↓
P0 = exp(L);
P1 = 1 – P0;



% need P1↑
P1 = exp(L);
P0 = 1 – P1;