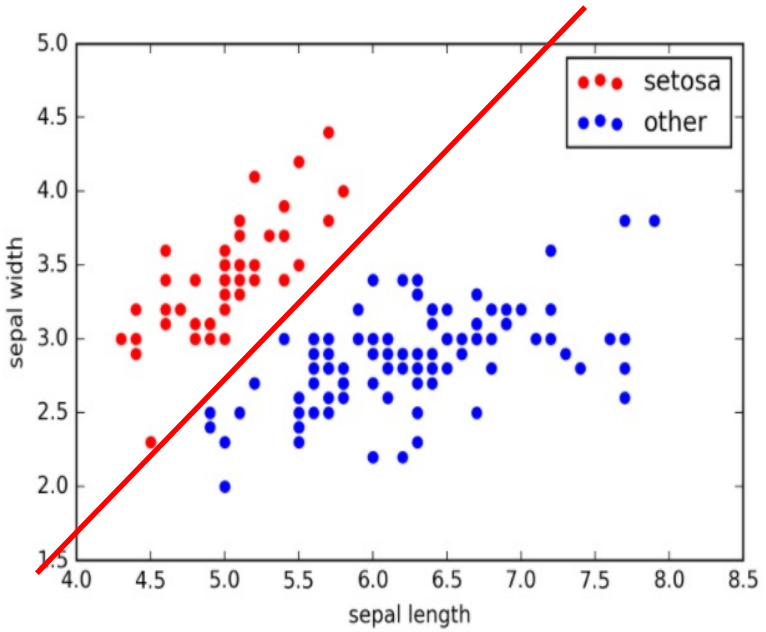
LOGISTIC REGRESSION

CS 334: Machine Learning

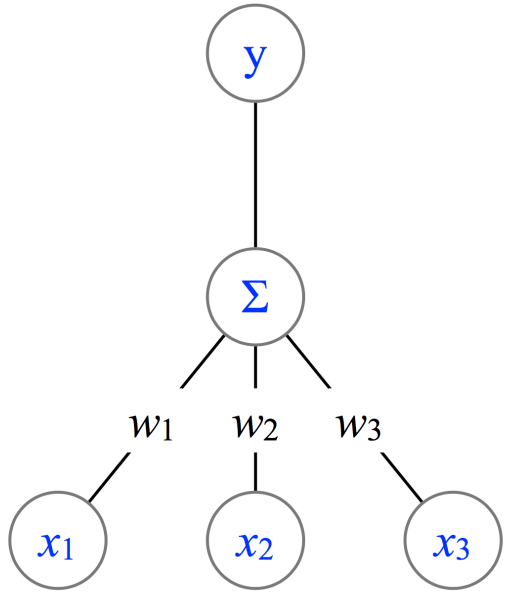
LINEAR CLASSIFIERS

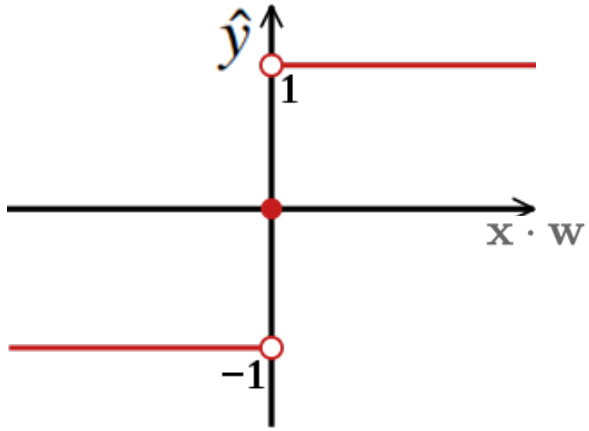
- Perceptron (minimize 0-1 loss)
- Logistic regression (minimize cross-entropy)
- Support Vector Machines (minimize margin)



REVIEW: PERCEPTRON

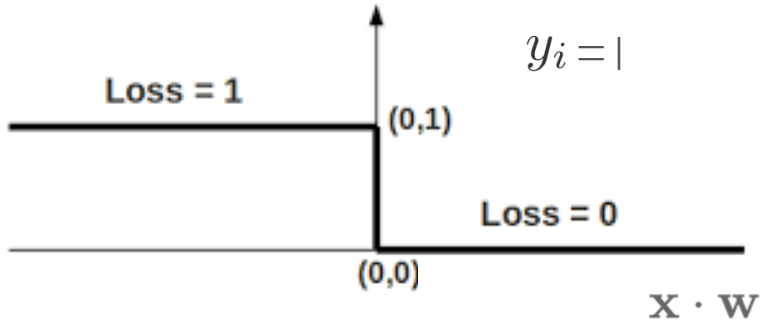
- Binary output: sign(**xw**)
  - xw is unbounded, using a threshold or sign function
- Not smooth, makes learning difficult

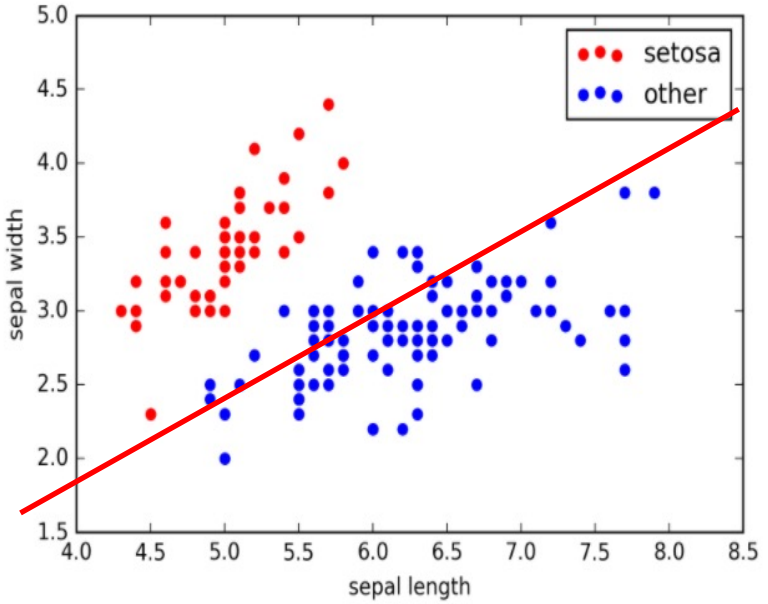




REVIEW: PERCEPTRON: 0-1 LOSS

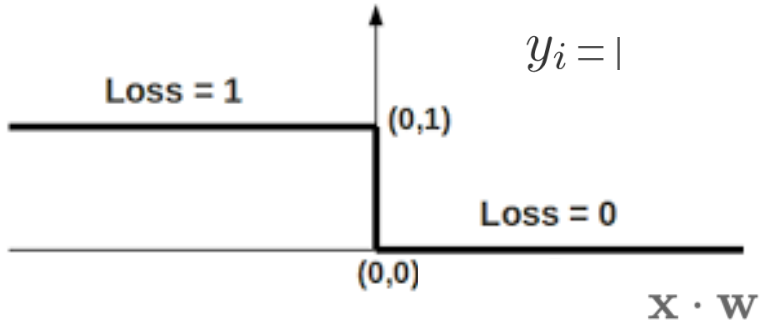
- 0-1 loss
  - Penalty 1 if the prediction is incorrect
  - Penalty 0 if the prediction is correct

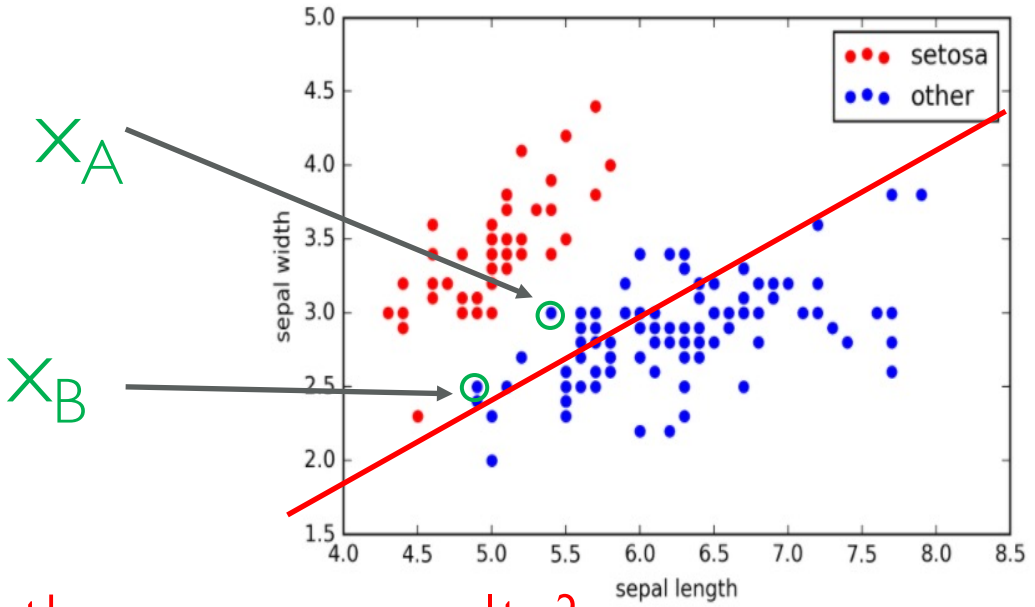




REVIEW: PERCEPTRON: 0-1 LOSS

- 0-1 loss
  - Penalty 1 if the prediction is incorrect
  - Penalty 0 if the prediction is correct





Should the two points have the same penalty?

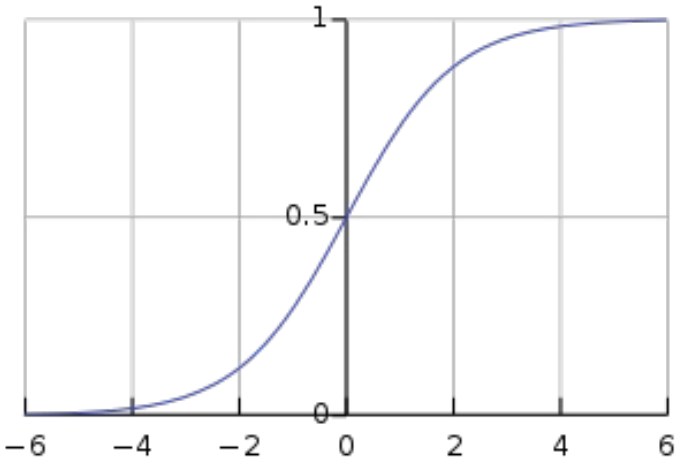
LOGISTIC REGRESSION

- Model the probability that the record belongs to a particular category
  - Gives continuous outputs between 0 and 1
- Want a loss function that penalizes the wrong predictions the correct way

LOGISTIC FUNCTION

- Logistic or sigmoid function

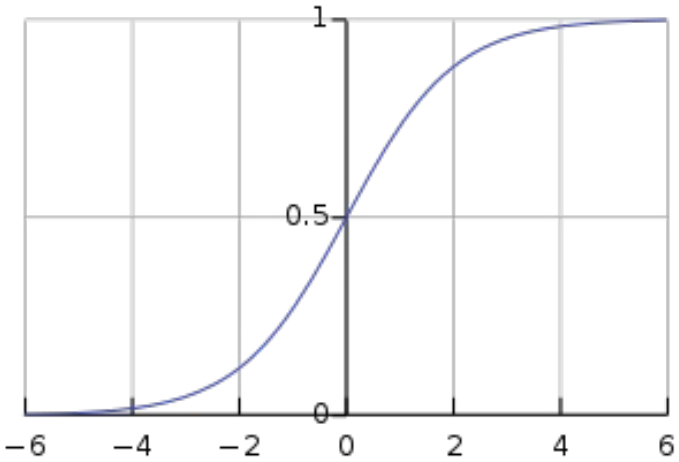
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



LOGISTIC REGRESSION

- Logistic or sigmoid function

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



- Apply sigmoid to linear function of the input features

$$z = \mathbf{x}\boldsymbol{\beta}$$

$$y = \sigma(z)$$

LEARNING LOGISTIC REGRESSION

- Want the output probability to be as close to 1 as possible for positive point (y=1), and as close to 0 as possible for negative point (y=0)

- Maximum likelihood estimation: maximize likelihood of predicting the true label in the training data given x

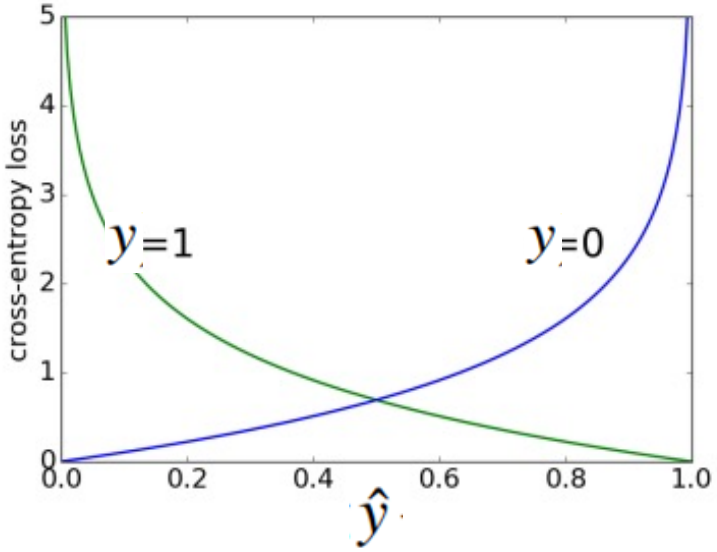
$$p(y|x) = \hat{y}^y (1-\hat{y})^{1-y}$$

$$\begin{aligned} \log p(y|x) &= \log \left[\hat{y}^y (1-\hat{y})^{1-y}\right] \\ &= y \log \hat{y} + (1-y) \log(1-\hat{y}) \end{aligned}$$

CROSS ENTROPY LOSS

- Minimize cross-entropy between predicted probability and ground truth

$$L_{CE}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$$



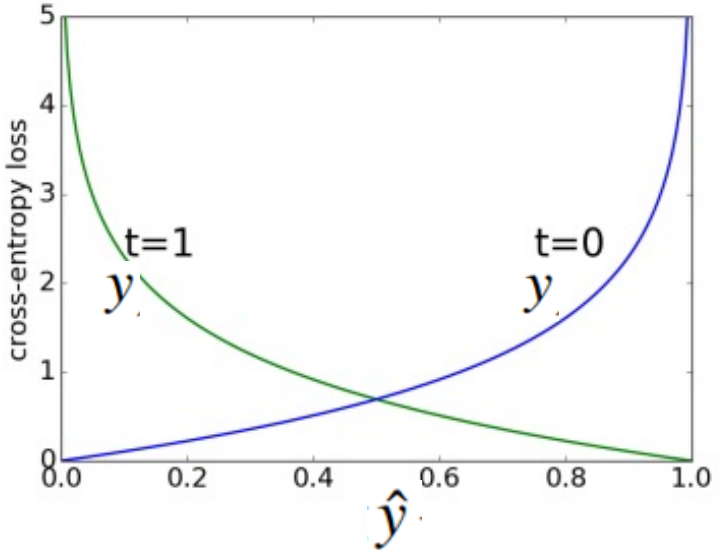
- Also work for multi-class classification):

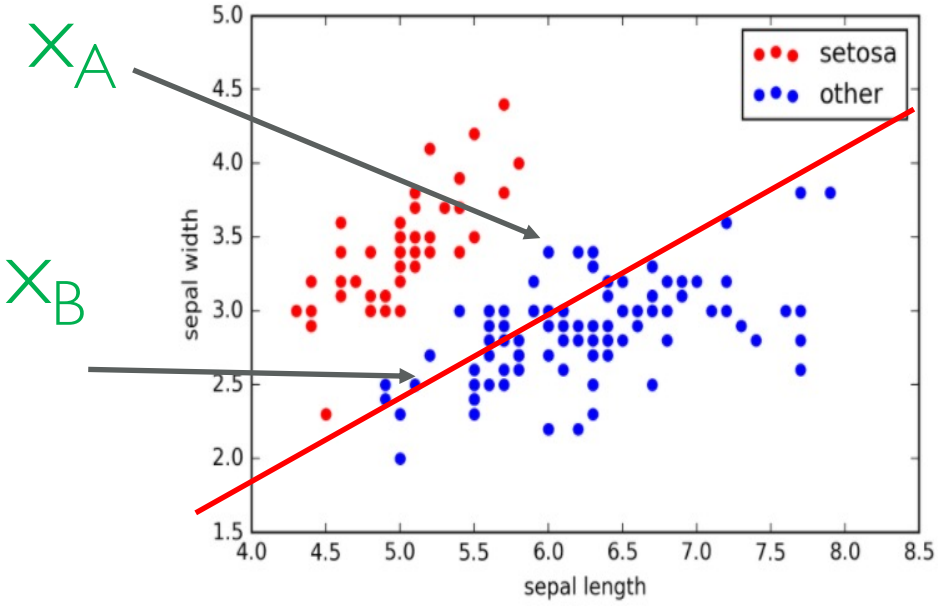
$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

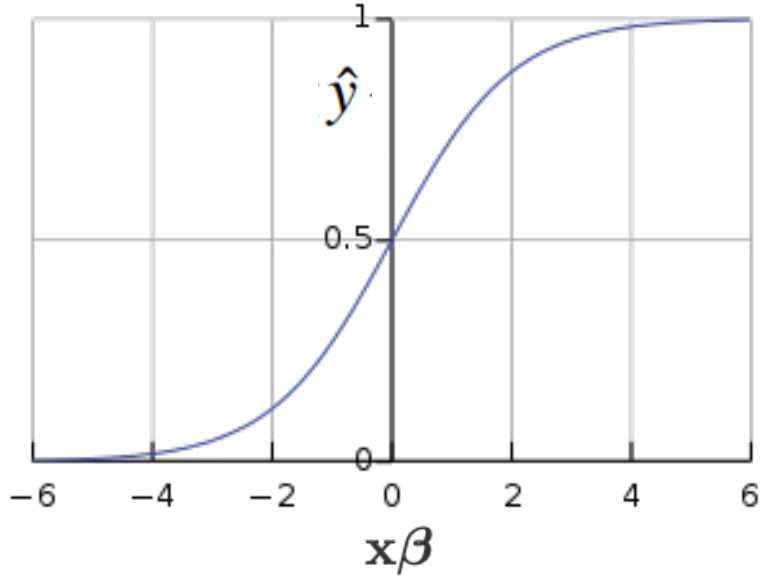
- Idea: An incorrect prediction that is more confident is penalized more

CROSS ENTROPY LOSS

- A point far away on the wrong side is penalized more ($\mathbf{x}\boldsymbol{\beta}$ measures the distance to the decision hyperplane)







REVIEW: HYPERPLANE

- General equation for a hyperplane

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0$$

- Normal vector $(\beta_1, \beta_2, \cdots, \beta_p)$ points in the direction orthogonal to the surface of a hyperplane

- $\mathbf{x}\boldsymbol{\beta}$ with the dummy attribute measures the perpendicular distance to the hyperplane

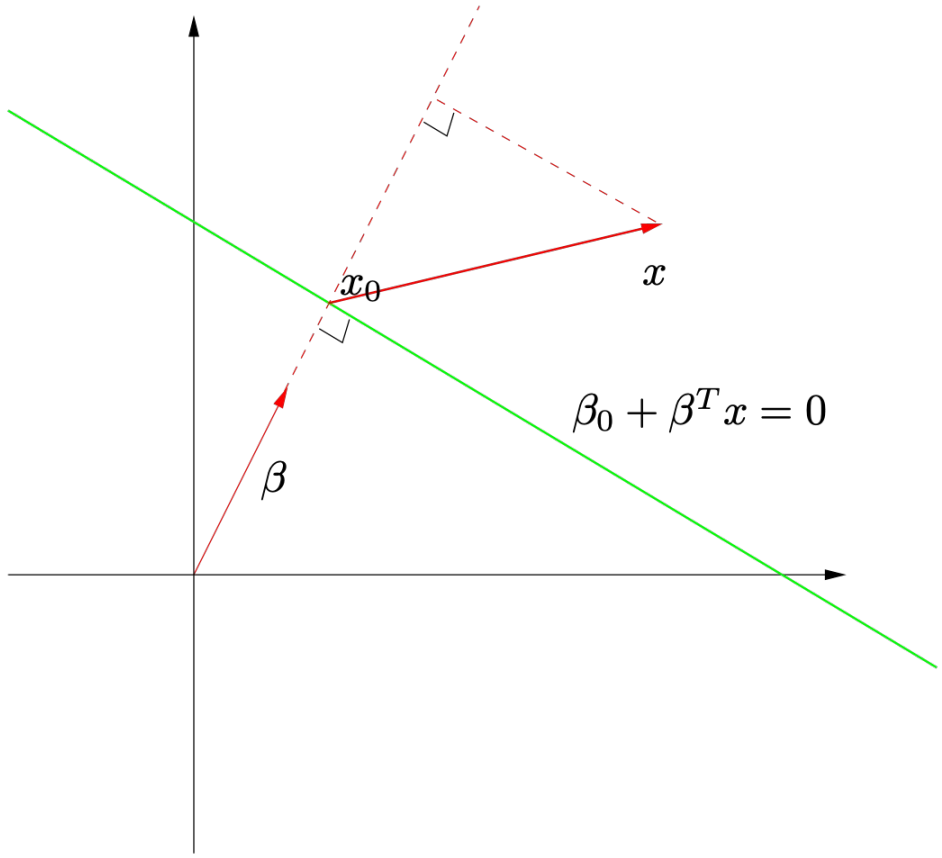
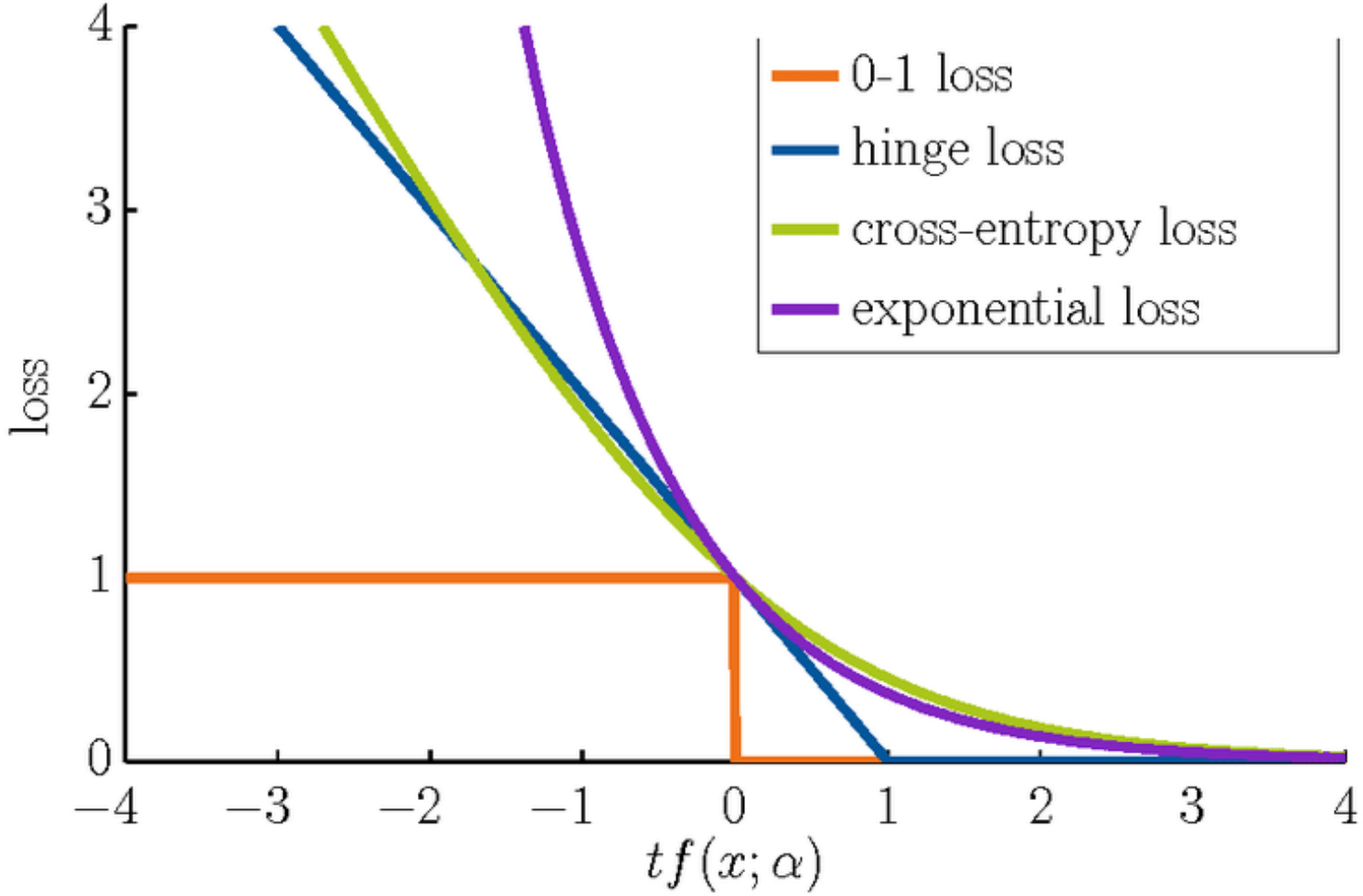


Figure 4.15 (Hastie et al.)

COMPARISON OF THE LOSS FUNCTIONS



FITTING LOGISTIC REGRESSION MODELS

- No longer straightforward (not simple closed form least squares)
- Use optimization methods (Newton's method) to find approximation
- In practice use a software library (e.g. newton-cg, lbfgs)
- Gradient descent

INTERPRETING LOGISTIC REGRESSION COEFFICIENTS

- Predicted probability of y=1

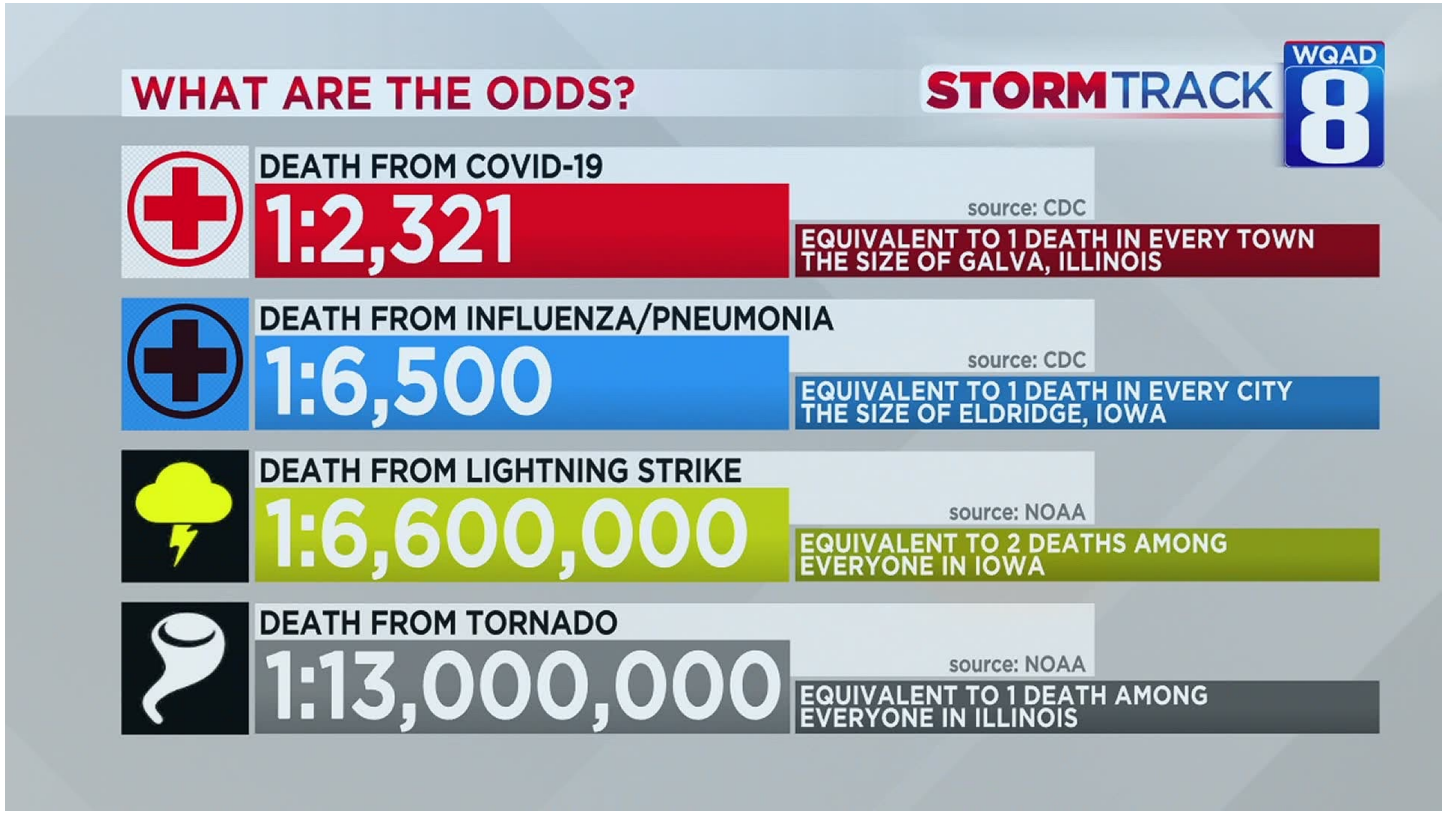
$$P(y = 1|\mathbf{x}, \beta) = \frac{\exp(\mathbf{x}\boldsymbol{\beta})}{1 + \exp(\mathbf{x}\boldsymbol{\beta})}$$

- Odds

$$\frac{P(y = 1|\mathbf{x}, \beta)}{1 - P(y = 1|\mathbf{x}, \beta)} = \exp(\mathbf{x}\boldsymbol{\beta})$$

ODDS RATIO

- Likelihood that an outcome will take place versus the outcome not taking place
- Example: Betting on rolling a number from a 6-sided fair die = 5:1



https://www.wqad.com/article/news/what-are-the-odds-of-dying-from-covid19-and-lightning/526-8596f974-c552-4a7a-ac21-1fdd87cb84da

LOG-ODDS (OR LOGIT)

- Odds

$$\frac{P(y=1|\mathbf{x},\beta)}{1-P(y=1|\mathbf{x},\beta)} = \exp(\mathbf{x}\boldsymbol{\beta})$$

- Log odds

$$\log\left(\frac{P(y=1|\mathbf{x},\beta)}{1-P(y=1|\mathbf{x},\beta)}\right) = \mathbf{x}\boldsymbol{\beta}$$

LOGISTIC REGRESSION COEFFICIENTS

- Increasing the ith predictor x_i by 1 unit and keeping all other predictors fixed increases:
  - Estimated log odds (class 1) by an additive factor β_i
  - Estimated odds (class 1) by a multiplicative factor $\exp \beta_i$

EXAMPLE: SOUTH AFRICAN HEART DISEASE

- Predict myocardial infarction – "heart attack"
- Variables:
  - sbp – Systolic blood pressure
  - tobacco – Tobacco use
  - ldl – Cholesterol measure
  - famhist – Family history of myocardial infarction
  - obesity, alcohol, age

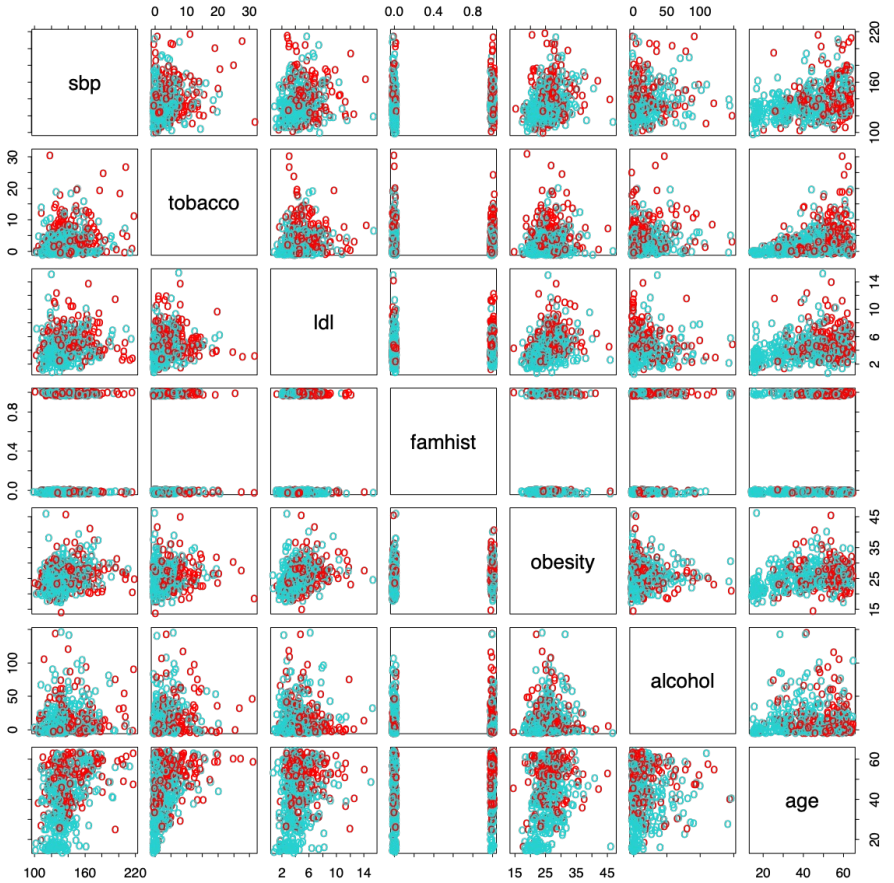


Figure 4.12 (Hastie et al.)

EXAMPLE: SOUTH AFRICAN HEART DISEASE

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Table 4.2 (Hastie et al.)

Why is the coefficient for age so small?

Can we do feature selection using logistic regression?

EXAMPLE: LOGISTIC REGRESSION WITH L1 REGULARIZATION (WITH FEATURE SCALING)

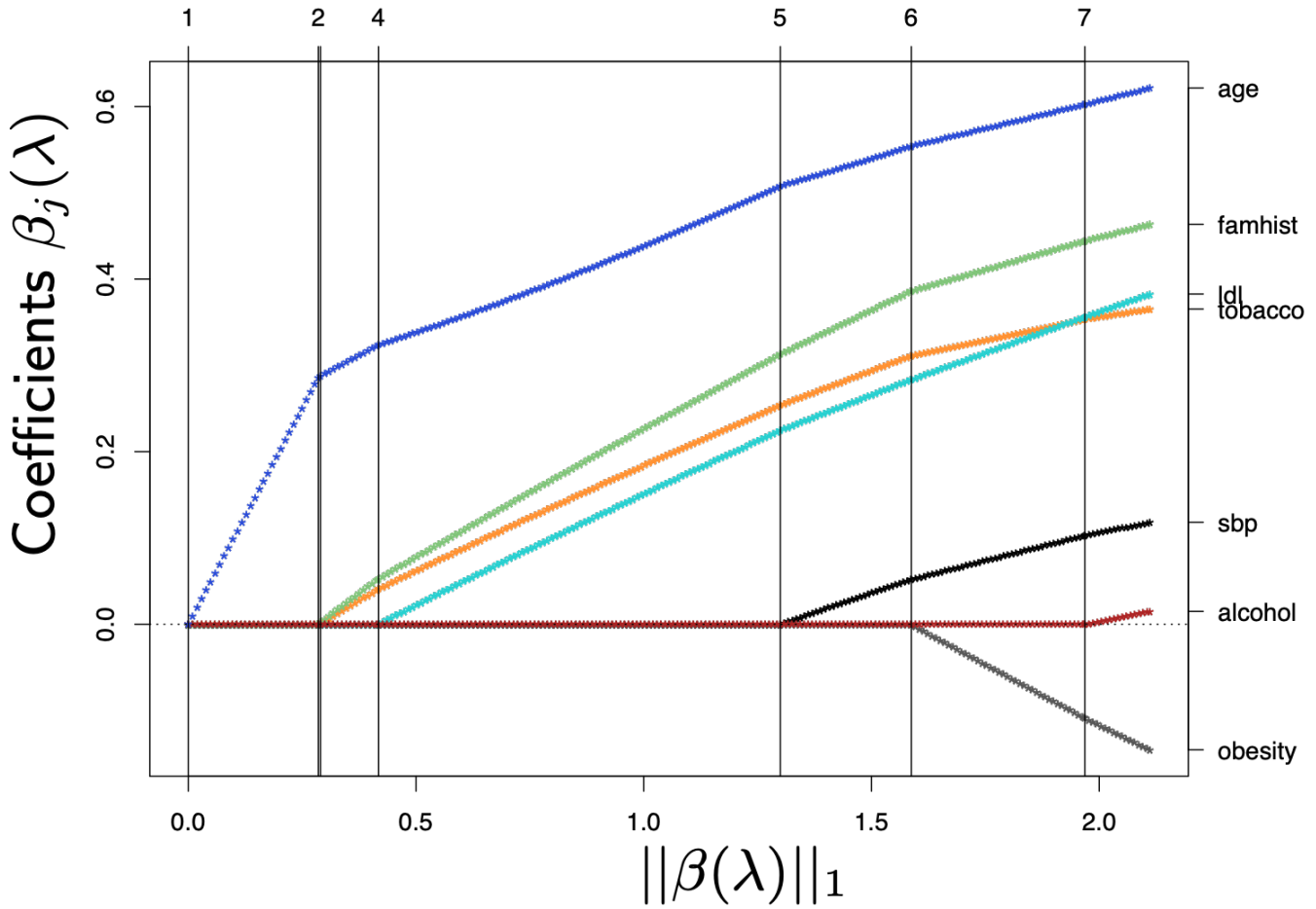


Figure 4.13 (Hastie et al.)

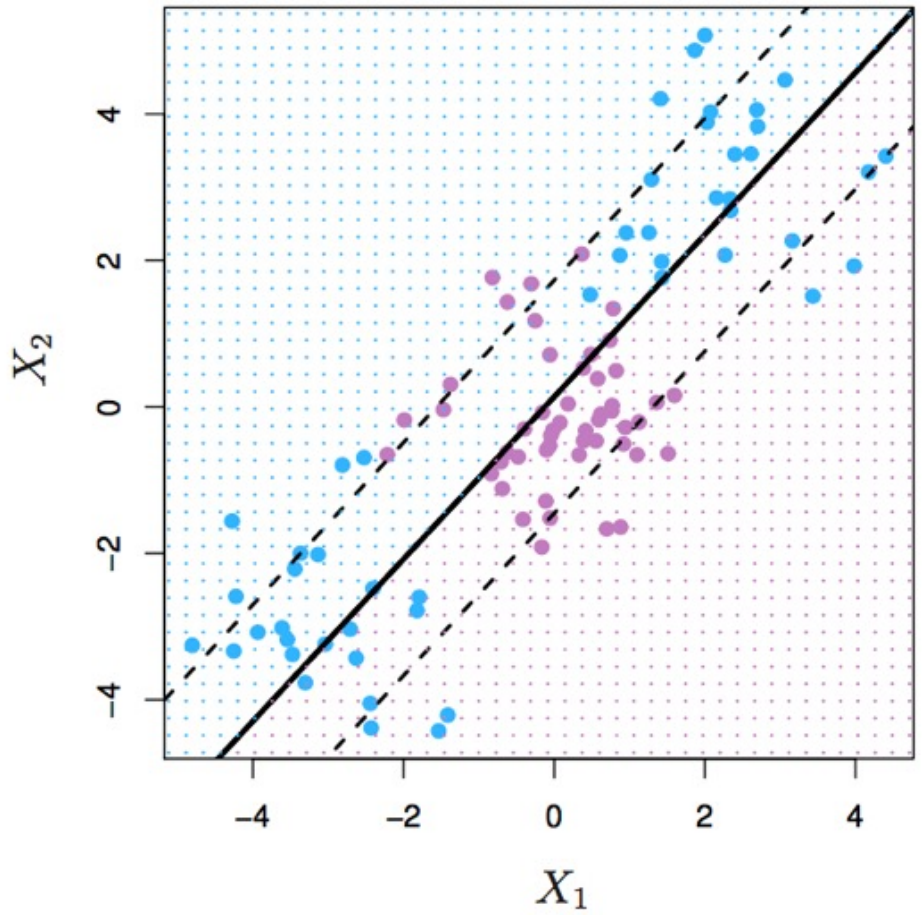
LOGISTIC REGRESSION

- Parameters have useful interpretations — the effect of unit change in a feature is to increase the odds of a response multiplicatively by the factor $\exp \beta_i$
- Supports different features
- Feature scaling recommended, esp. when regularization is used
- Quite robust, well developed

LOGISTIC REGRESSION DISADVANTAGES

- Parametric, but works for entire exponential family of distributions
- Solution not closed form, but still reasonably fast
- Requires large sample size to achieve stable results

HOW TO GET NONLINEAR DECISION BOUNDARIES?



Sometimes a linear boundary won't work.
What should we do in this case?

FEATURE EXPANSION

- Enlarge the space of features by including transformations
  - Example: $x_1^2, x_1^3, x_1 x_2, x_1 x_2^2$
  - Feature space dimension from p to p' where p' > p
- Fit logistic regression on new feature space => non-linear boundaries in original space

GROUP ACTIVITY

EXAMPLE: FEATURE EXPANSION

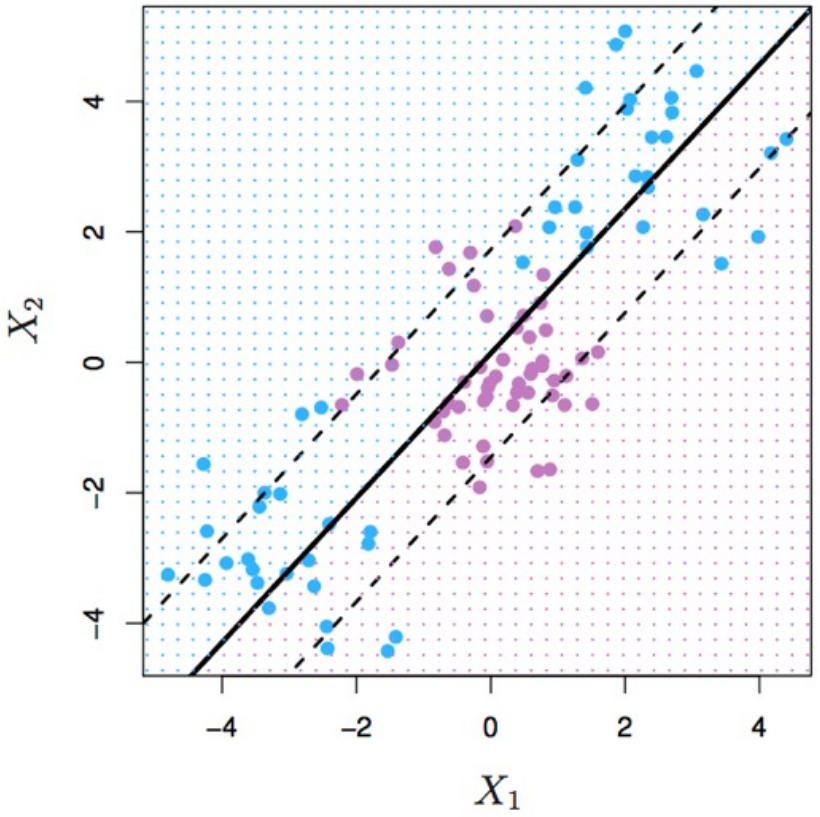
- Feature expansion

$$(x_1, x_2) \rightarrow (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

- Logistic regression coefficients

$$\beta = [-1, 0, 0, 1, 1, 0]$$

- What does this decision boundary look like in (x_1, x_2) plane?



LOGISTIC REGRESSION: SKLEARN

- LogisticRegression
- LogisticRegressionCV

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Colab: https://colab.research.google.com/drive/1_zBn88flEekqYvI1byLzAQEthVTJDHsr?usp=sharing

SAheart.csv on Canvas/Files

SCIKIT-Learn Reference: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html