COMP 551 – Applied Machine Learning
Lecture 4: Linear classification

Instructor: Joelle Pineau (*jpineau@cs.mcgill.ca)*

Class web page: *www.cs.mcgill.ca/~jpineau/comp551*

Today's Quiz

1. What is meant by the term *overfitting*? What can cause overfitting? How can one avoid overfitting?

2. Which of the following increases the chances of overfitting (assuming everything else is held constant):
   a) Reducing the size of the training set.
   b) Increasing the size of the training set.
   c) Reducing the size of the test set.
   d) Increasing the size of the test set.
   e) Reducing the number of features.
   f) Increasing the number of features.

Evaluation

- Use **cross-validation** for **model selection**.

- **Training set** is used to select a hypothesis *f* from a class of hypotheses *F* (e.g. regression of a given degree).
- **Validation set** is used to compare best *f* from each hypothesis class across different classes (e.g. different degree regression).
    - Must be untouched during the process of looking for *f* within a class *F*.

- **Test set**: Ideally, a separate set of (labeled) data is withheld to get a true estimate of the generalization error.

  (Often the "validation set" is called "test set", without distinction.)

Validation vs Train error

[From Hastie et al. textbook]

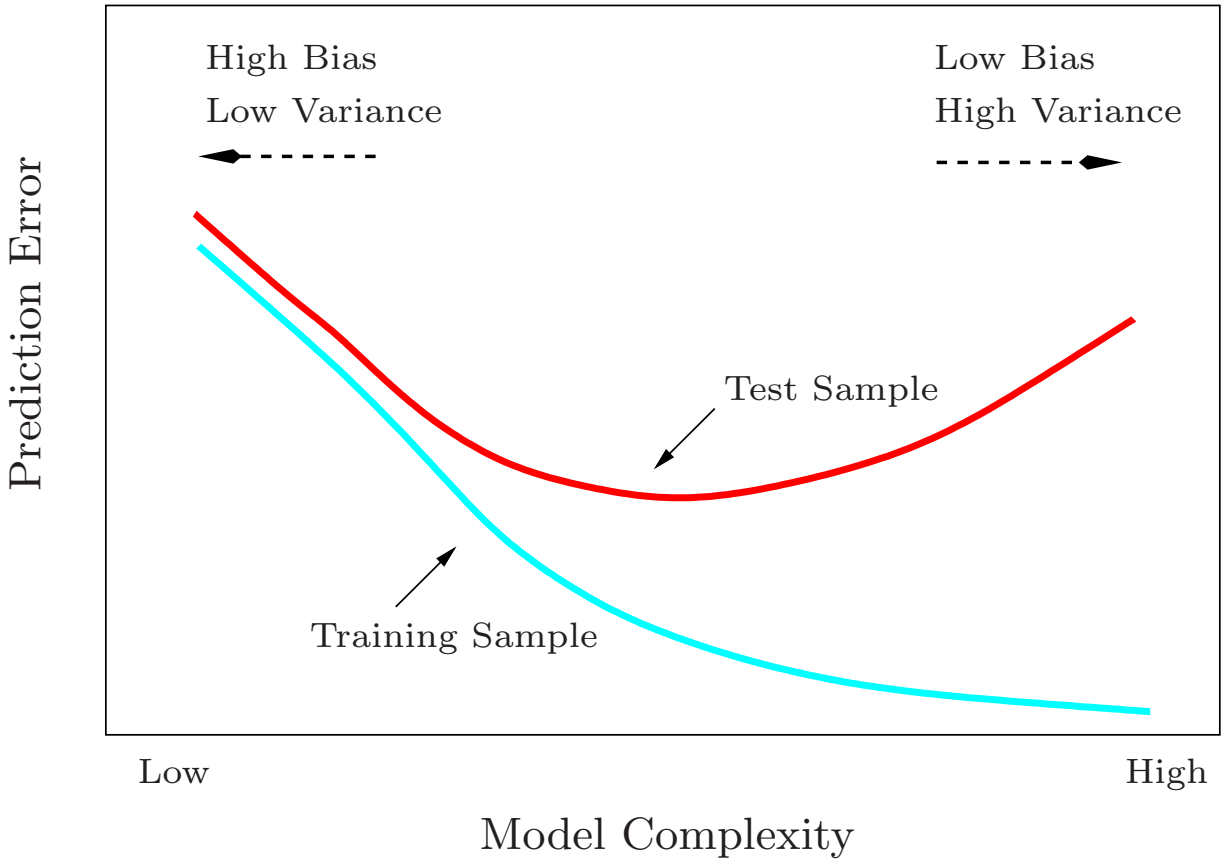


FIGURE 2.11. *Test and training error as a function of model complexity.*

Bias vs Variance

- Gauss-Markov Theorem says:

  The least-squares estimates of the parameters ***w*** have the **smallest variance** among all linear **unbiased** estimates.

- **Insight**: Find lower variance solution, at the expense of some bias.

E.g. Include **penalty** for model complexity in error to reduce overfitting.

$$Err(w) = \sum_{i=1:n} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2 + \lambda \, |model\_size|$$

λ is a hyper-parameter that controls penalty size.

Ridge regression (aka L2-regularization)

- Constrains the weights by imposing a penalty on their size:

$$\hat{\mathbf{w}}^{ridge} = argmin_{\mathbf{w}} \{ \sum_{i=1:n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda \sum_{j=0:m} w_j^2 \}$$

  where λ can be selected manually, or by cross-validation.

- Do a little algebra to get the solution: $\hat{\mathbf{w}}^{ridge} = (X^TX+\lambda I)^{-1}X^TY$

  – The ridge solution is not equivariant under scaling of the data, so typically need to normalize the inputs first.

  – Ridge gives a smooth solution, effectively shrinking the weights, but drives few weights to 0.

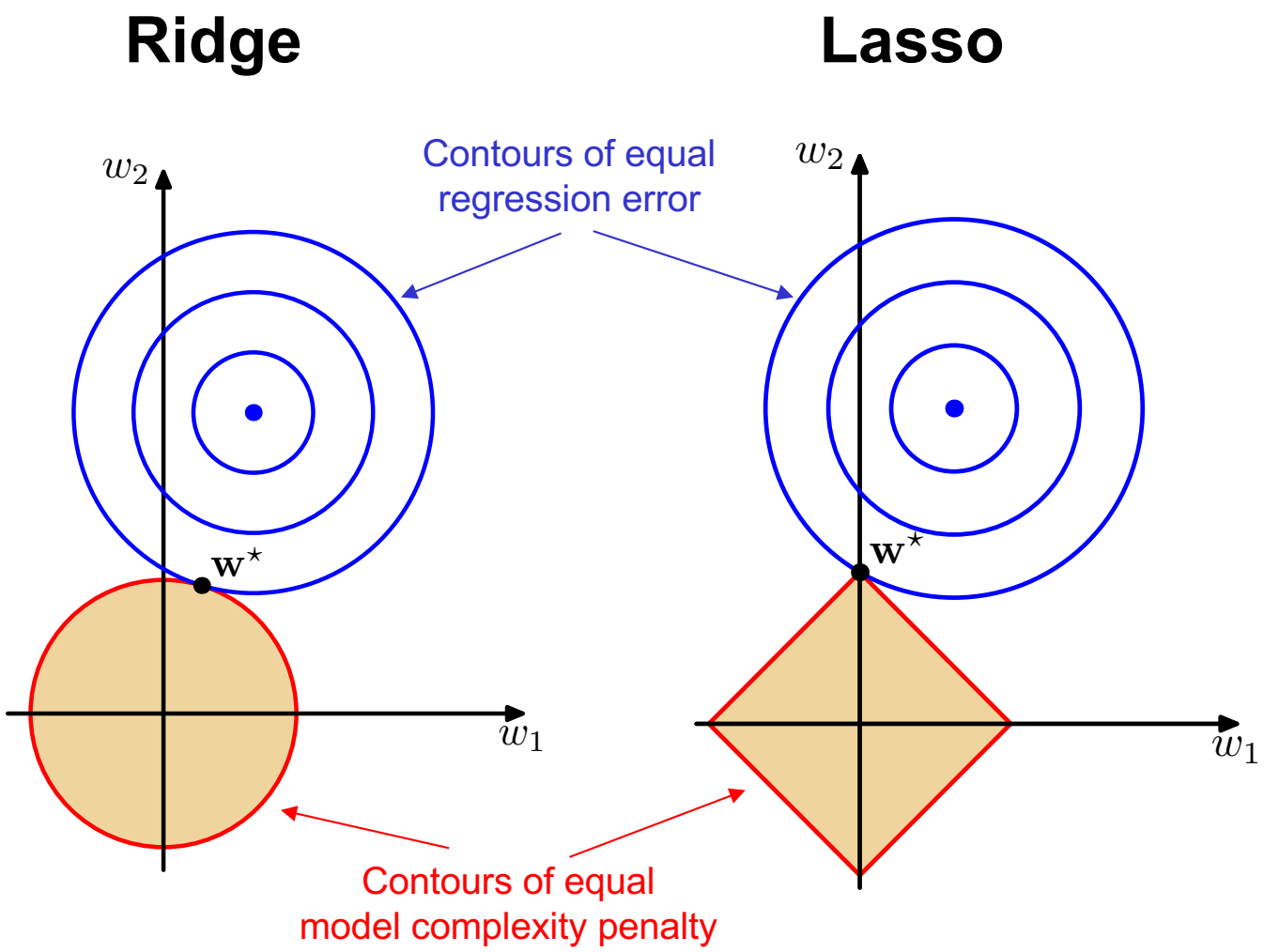
Lasso regression (aka L1-regularization)

- Constrains the weights by penalizing the absolute value of their size:

$$\hat{\mathbf{w}}^{lasso} = argmin_W \{ \sum_{i=1:n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 + \lambda \sum_{j=1:m} |w_j| \}$$

- Now the objective is non-linear in the output y, and there is no closed-form solution. Need to solve a quadratic programming problem instead.
  - More computationally expensive than Ridge regression.
  - Effectively sets the weights of less relevant input features to zero.

Comparing Ridge and Lasso



A quick look at evaluation functions

- We call $L(Y, f_{\mathbf{w}}(x))$ the loss function.
  - Least-square / Mean squared-error (MSE) loss:

$$L(Y, f_{\mathbf{w}}(X)) = \sum_{i=1:n} (y_i - \mathbf{w}^T x_i)^2$$

- Other loss functions?
  - Absolute error loss: $L(Y, f_{\mathbf{w}}(X)) = \sum_{i=1:n} | y_i - \mathbf{w}^T x_i |$
  - 0-1 loss (for classification): $L(Y, f_{\mathbf{w}}(X)) = \sum_{i=1:n} I (y_i \neq f_{\mathbf{w}}(x_i))$

- Different loss functions make **different assumptions**.
  - Squared error loss assumes the data can be approximated by a global linear model with Gaussian noise.

Next: Linear models for classification

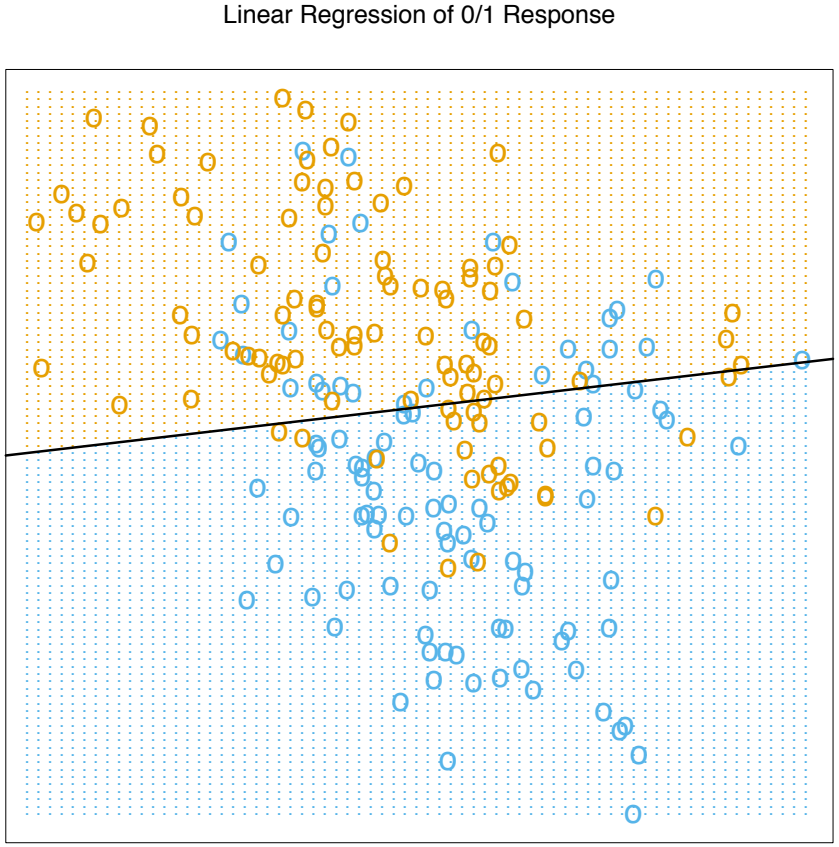


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$*. The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE*.*

Classification problems

Given data set $D=<x_i,y_i>$, $i=1:n$, **with discrete y_i**, find a hypothesis which "best fits" the data.

- If $y_i \in \{0, 1\}$ this is binary classification.
- If y_i can take more than two values, the problem is called multi-class classification.

Fisher's *Iris* Data

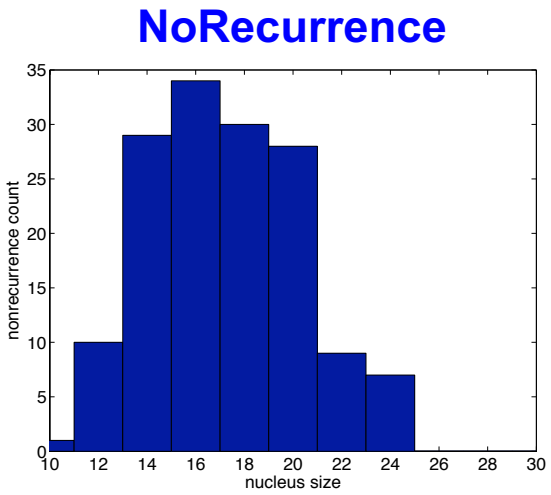
Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*
4.4	2.9	1.4	0.2	*I. setosa*
4.9	3.1	1.5	0.1	*I. setosa*
5.4	3.7	1.5	0.2	*I. setosa*
4.8	3.4	1.6	0.2	*I. setosa*
4.8	3.0	1.4	0.1	*I. setosa*
4.3	3.0	1.1	0.1	*I. setosa*
5.8	4.0	1.2	0.2	*I. setosa*
5.7	4.4	1.5	0.4	*I. setosa*
5.4	3.9	1.3	0.4	*I. setosa*
5.1	3.5	1.4	0.3	*I. setosa*
5.7	3.8	1.7	0.3	*I. setosa*
5.1	3.8	1.5	0.3	*I. setosa*
5.4	3.4	1.7	0.2	*I. setosa*
5.1	3.7	1.5	0.4	*I. setosa*
4.6	3.6	1.0	0.2	*I. setosa*

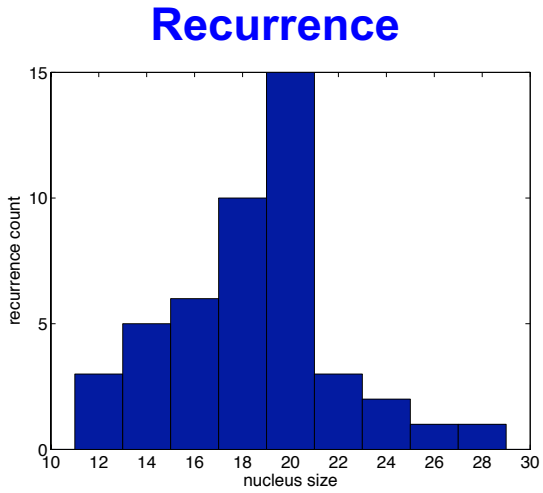
Applications of classification

- Text classification (spam filtering, news filtering, building web directories, etc.)
- Image classification (face detection, object recognition, etc.)
- Prediction of cancer recurrence.
- Financial forecasting.
- Many, many more!

Simple example

- Given “nucleus size”, predict cancer recurrence.
- Univariate input: X = nucleus size.
- Binary output: Y = {NoRecurrence = 0; Recurrence = 1}
- Try: Minimize the least-square error.





Predicting a class from linear regression

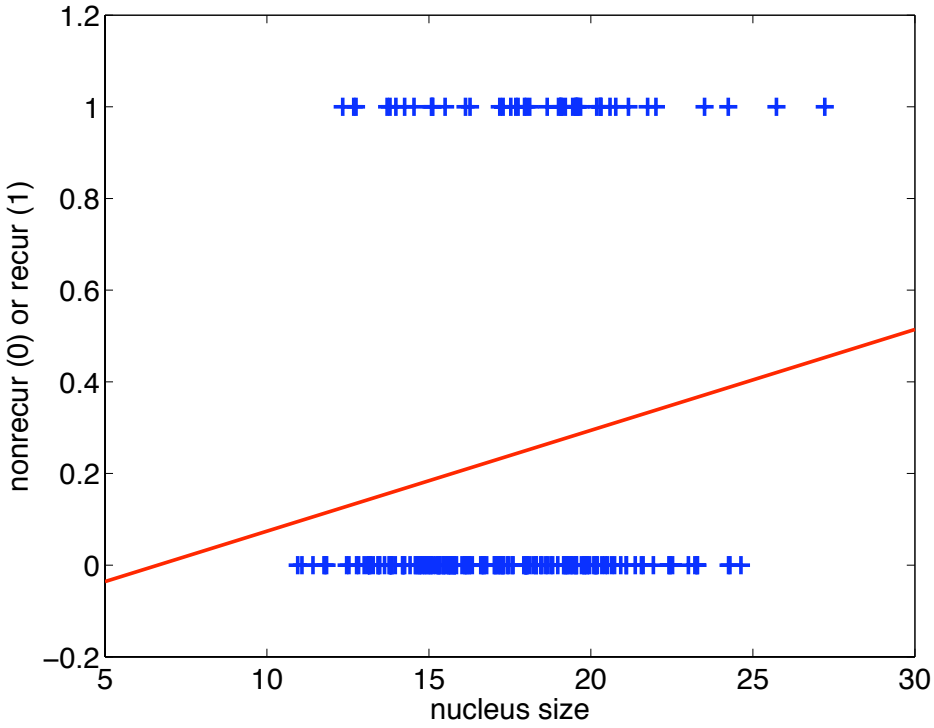
- Here red line is: $Y' = X (X^T X)^{-1} X^T Y$

- How to get a binary output?
  1. Threshold the output:

     { $y <= t$ for NoRecurrence,

     $y > t$ for Recurrence}
  2. Interpret output as probability:

     $y = Pr$ (Recurrence)



- Can we find a better model?

Modeling for binary classification

- Two probabilistic approaches:

1. **Discriminative learning**: Directly estimate *P(y|x)*.

2. **Generative learning**: Separately model *P(x|y)* and *P(y)*. Use Bayes rule, to estimate *P(y|x)*:

$$P(y=1 \mid x) = \frac{P(x \mid y=1)P(y=1)}{P(x)}$$

Probabilistic view of discriminative learning

- Suppose we have 2 classes: $y \in \{0, 1\}$
- What is the probability of a given input x having class $y = 1$?
- Consider Bayes rule:

$$P(y=1 \mid x) = \frac{P(x, y=1)}{P(x)} = \frac{P(x \mid y=1)P(y=1)}{P(x \mid y=1)P(y=1) + P(x \mid y=0)P(y=0)}$$

$$= \frac{1}{1 + \frac{P(x \mid y=0)P(y=0)}{P(x \mid y=1)P(y=1)}} = \frac{1}{1 + \exp(\ln \frac{P(x \mid y=0)P(y=0)}{P(x \mid y=1)P(y=1)})} = \frac{1}{1 + \exp(-a)} = \sigma$$

where $a = \ln \frac{P(x \mid y=1)P(y=1)}{P(x \mid y=0)P(y=0)} = \ln \frac{P(y=1 \mid x)}{P(y=0 \mid x)}$

(By Bayes rule; *P(x) on top and bottom cancels out.)*

- Here σ has a special form, called the **logistic function**

  and a is the log-odds ratio of data being class 1 vs. class 0.

Discriminative learning: Logistic regression

- Idea: Directly model the log-odds with a linear function:

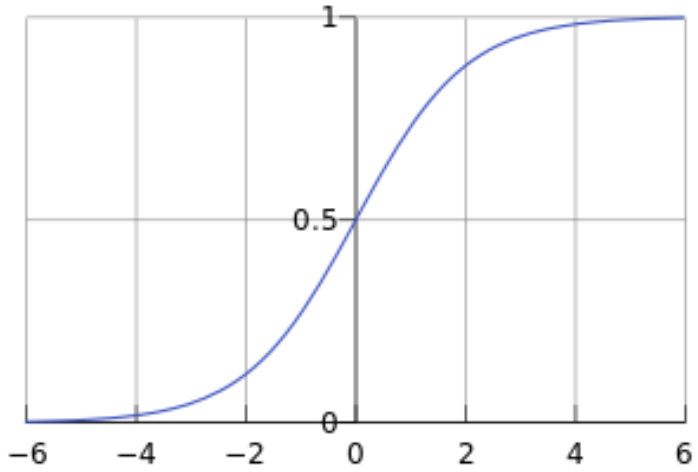
$$a = \ln \frac{P(x \mid y=1)P(y=1)}{P(x \mid y=0)P(y=0)} = w_0 + w_1 x_1 + \ldots + w_m x_m$$

- The **decision boundary** is the set of points for which $a=0$.

- The **logistic** function (= sigmoid curve): $\sigma(\mathbf{w}^T\mathbf{x}) = 1 / (1 + e^{-wTx})$

How do we find the weights?

Need an optimization function.



Fitting the weights

- Recall: $\sigma(\mathbf{w}^T\mathbf{x}_i)$ is the probability that $y_i=1$ (given x_i)

  $1-\sigma(\mathbf{w}^T\mathbf{x}_i)$ be the probability that $y_i = 0$.

- For $y \in \{0, 1\}$, the **likelihood** function, $Pr(x_1,y_1, \ldots, x_n,y_h \mid w)$, is:

  $\prod_{i=1:n} \sigma(\mathbf{w}^T\mathbf{x}_i)^{yi} (1- \sigma(\mathbf{w}^T\mathbf{x}_i))^{(1-yi)}$ (samples are i.i.d.)

- **Goal**: Minimize the **log-likelihood** (also called *cross-entropy error function*):

  $- \sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

$\delta \log(\sigma)/\delta w = 1/\sigma$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\,\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + \ldots$$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

$\delta\sigma/\delta w = \sigma(1-\sigma)$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\,\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + \ldots$$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

$\delta w^T x/\delta w = x$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\,\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + \ldots$$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

$\delta(1-\sigma)/\delta w = (1-\sigma)\sigma(-1)$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\,\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + (1-y_i)(1/(1-\sigma(\mathbf{w}^T\mathbf{x}_i)))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\sigma(\mathbf{w}^T\mathbf{x}_i)(-1)\,\mathbf{x}_i]$$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + (1-y_i)(1/(1-\sigma(\mathbf{w}^T\mathbf{x}_i)))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\sigma(\mathbf{w}^T\mathbf{x}_i)(-1)\mathbf{x}_i]$$

$$= -\sum_{i=1:n} \mathbf{x}_i (y_i (1-\sigma(\mathbf{w}^T\mathbf{x}_i)) - (1-y_i)\sigma(\mathbf{w}^T\mathbf{x}_i))$$

$$= -\sum_{i=1:n} \mathbf{x}_i (y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))$$

Gradient descent for logistic regression

- Error fn: $Err(\mathbf{w}) = -[\sum_{i=1:n} y_i \log(\sigma(\mathbf{w}^T\mathbf{x}_i)) + (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i))]$

- Take the derivative:

$$\partial Err(\mathbf{w})/\partial \mathbf{w} = -[\sum_{i=1:n} y_i (1/\sigma(\mathbf{w}^T\mathbf{x}_i))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i + (1-y_i)(1/(1-\sigma(\mathbf{w}^T\mathbf{x}_i)))(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\sigma(\mathbf{w}^T\mathbf{x}_i)(-1)\mathbf{x}_i]$$

$$= -\sum_{i=1:n} \mathbf{x}_i (y_i (1-\sigma(\mathbf{w}^T\mathbf{x}_i)) - (1-y_i)\sigma(\mathbf{w}^T\mathbf{x}_i))$$

$$= -\sum_{i=1:n} \mathbf{x}_i (y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))$$

- Now apply iteratively: $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \sum_{i=1:n} \mathbf{x}_i (y_i - \sigma(\mathbf{w}_k^T\mathbf{x}_i))$

- Can also apply other iterative methods, e.g. Newton's method, coordinate descent, L-BFGS, etc.

Modeling for binary classification

- Two probabilistic approaches:

1. **Discriminative learning**: Directly estimate *P(y|x)*.

2. **Generative learning**: Separately model *P(x|y)* and *P(y)*. Use Bayes rule, to estimate *P(y|x)*:

$$P(y=1 \mid x) = \frac{P(x \mid y=1)P(y=1)}{P(x)}$$

What you should know

- Basic definition of linear classification problem.
- Derivation of logistic regression.
- Linear discriminant analysis: definition, decision boundary.
- Quadratic discriminant analysis: basic idea, decision boundary.
- LDA vs QDA pros/cons.

- Worth reading further:
  - Under some conditions, linear regression for classification and LDA are the same (Hastie et al., p.109-110).
  - Relation between Logistic regression and LDA (Hastie et al., 4.4.5)

Final notes

You don't yet have a team for Project #1? => Use *myCourses*.

You don't yet have a plan for Project #1? => Start planning!

Feedback on tutorial 1?