

CSCI 5622 Machine Learning

ML Evaluating Hypotheses

DATE	READ	DUE
Today, Sept 14	6.11 + TM	---
Wed, Sept 16	4	Full Proposal
Mon, Sept 21	4	Peer Review 1

www.RodneyNielsen.com/teaching/CSCI5622-F09/

Instructor: Rodney Nielsen

Assistant Professor Adjunct, CU Dept. of Computer Science

Research Assistant Professor, DU, Dept. of Electrical & Computer Engr.

Research Scientist, Boulder Language Technologies

ML **Bayesian Learning**

- **Probabilistic approach to inference**
- **Assumes instances are governed by distribution conditioned on the label**
- **Basis for learning algorithms that directly utilize probabilities**
- **Framework for analyzing other ML algorithms**
- **Among best for some problems**

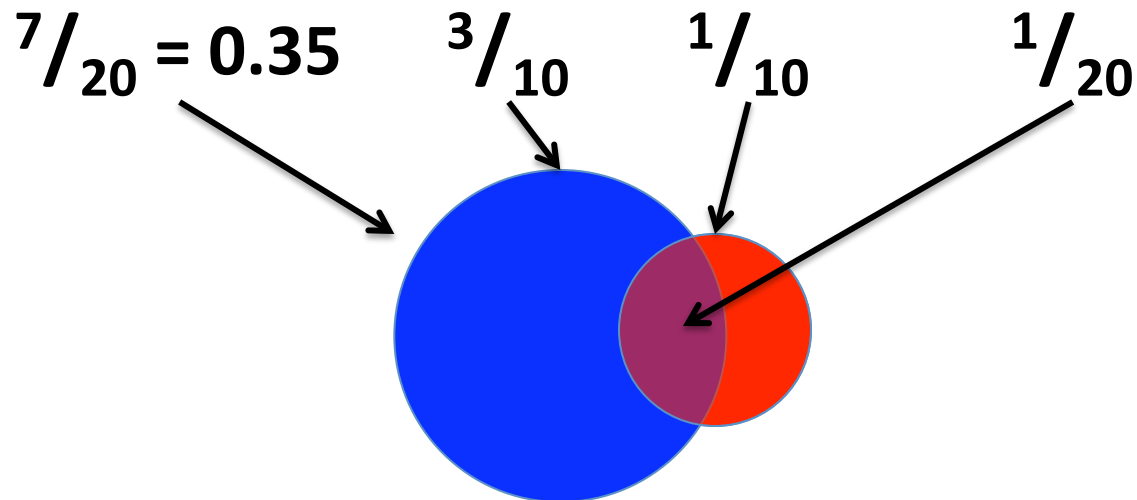
ML Bayesian Features

- **Adjust probability that a given hypothesis is correct vs. eliminating hypotheses**
- **Can easily incorporate prior knowledge about probabilities**
- **Can classify according to multiple weighted h**
- **When computationally intractable, can still provide standard**

Basic Probability Rules

- **Sum rule**
 - $P(\text{Glasses}) = 0.3$
 - $P(\text{Sweater}) = 0.1$
 - $P(\text{Glasses} \vee \text{Sweater}) = ?$

$$P(G \vee S) = P(G) + P(S) - P(G \wedge S)$$

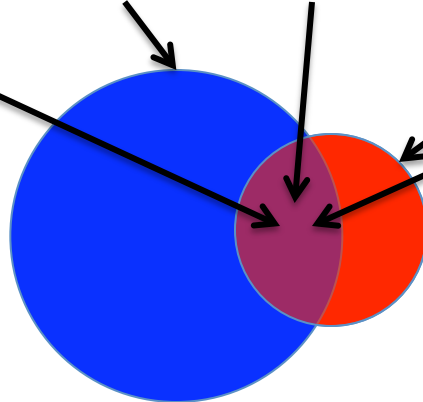


Basic Probability Rules

- **Product rule**
 - $P(\text{Glasses}) = 0.3$
 - $P(\text{Sweater}) = 0.1$
 - $P(\text{Glasses} \wedge \text{Sweater}) = ?$

$$P(G \wedge S) = P(G)P(S|G) = P(S)P(G|S)$$

$$\frac{1}{20} = 0.05 \quad \frac{3}{10} \quad \frac{1}{6} \quad \frac{1}{10} \quad \frac{1}{2}$$



Basic Probability Rules

- **Bayes Theorem**

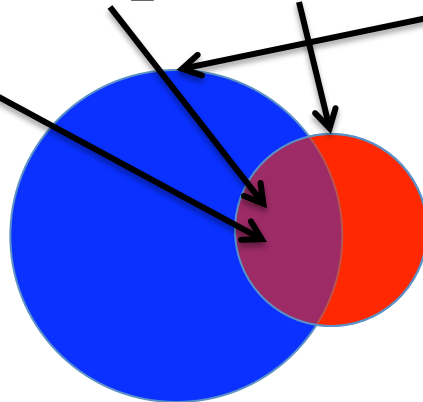
- $P(\text{Glasses}) = 0.3$

- $P(\text{Sweater}) = 0.1$

- $P(\text{Glasses} \mid \text{Sweater}) = ?$

$$P(S|G) = P(G|S)P(S)/P(G)$$

$$\frac{1}{6} = 0.167 \quad \frac{1}{2} \quad \frac{1}{10} \quad \frac{3}{10}$$



Basic Probability Rules

- **Theorem of total probability**

- $P(\text{Glasses} \mid \text{Sweater}) = 0.4$; $P(\text{Sweater}) = 0.1$

- $P(\text{Glasses} \mid \text{Shirt}) = 0.2$; $P(\text{Shirt}) = 0.6$

- $P(\text{Glasses} \mid \text{WetSuit}) = 0.0$; $P(\text{WetSuit}) = 0.1$

- $P(\text{Glasses} \mid \text{Other}) = 0.1$; $P(\text{Other}) = 0.2$

$$P(x \wedge y) = 0; x \neq y; x, y \in \{\text{Sweater}, \text{Shirt}, \text{WetSuit}, \text{Other}\}$$

$$\sum_{x \in \{\text{Sweater}, \text{Shirt}, \text{WetSuit}, \text{Other}\}} P(x) = 1.0$$

$$P(\text{Glasses}) = \sum_{x \in \{\text{Sweater}, \text{Shirt}, \text{WetSuit}, \text{Other}\}} P(\text{Glasses} \mid x) P(x)$$

$$P(\text{Glasses}) = 0.4 \cdot 0.1 + 0.2 \cdot 0.6 + 0.0 \cdot 0.1 + 0.1 \cdot 0.2 = 0.18$$

Bayes Theorem

- Find the most likely hypothesis given:
 - The prior probability of h , the probability of the data given h , and the observed data

$$P(h|\mathbf{D}) = \frac{P(\mathbf{D}|h)P(h)}{P(\mathbf{D})}$$

- $P(h)$ is the prior probability of h
- $P(D)$ is the prior probability of D
- $P(D|h)$: the prob. of D given h (likelihood)
- $P(h|D)$: posterior prob. of h given the data

Bayes Learning

- **Maximum a posteriori hypothesis**

$$\begin{aligned}h_{MAP} &\equiv \arg \max_{h \in H} P(h|D) \\ &= \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg \max_{h \in H} P(D|h)P(h)\end{aligned}$$

- **Maximum likelihood hypothesis**

$$h_{ML} \equiv \arg \max_{h \in H} P(D|h)$$

Naïve Bayes Classifier

- Assume independence of attributes given the class

$$P(x_1, x_2 \cdots x_d | y_j) = \prod_i P(x_i | y_j)$$

$$\hat{y}_{NB} = \arg \max_{y_j \in C} P(y_j) \prod_i P(x_i | y_j)$$

- Now we need only estimate the $P(y_j)$ and the $P(x_i | y_j)$ rather than all possible $P(x_1, x_2 \cdots x_d, | y_j)$

ML Conditional Independence

- x_1 and x_2 are conditionally independent given y , iff:

$$P(x_1|x_2, y) = P(x_1|y); \forall x_1, x_2, y$$

ML Bayesian Belief Networks

- **AKA Bayes Nets**
- **Intermediate approach between**
 - typically intractable attempt to avoid conditional independence assumptions

$$P(x_1, x_2 \cdots x_d | y) =$$

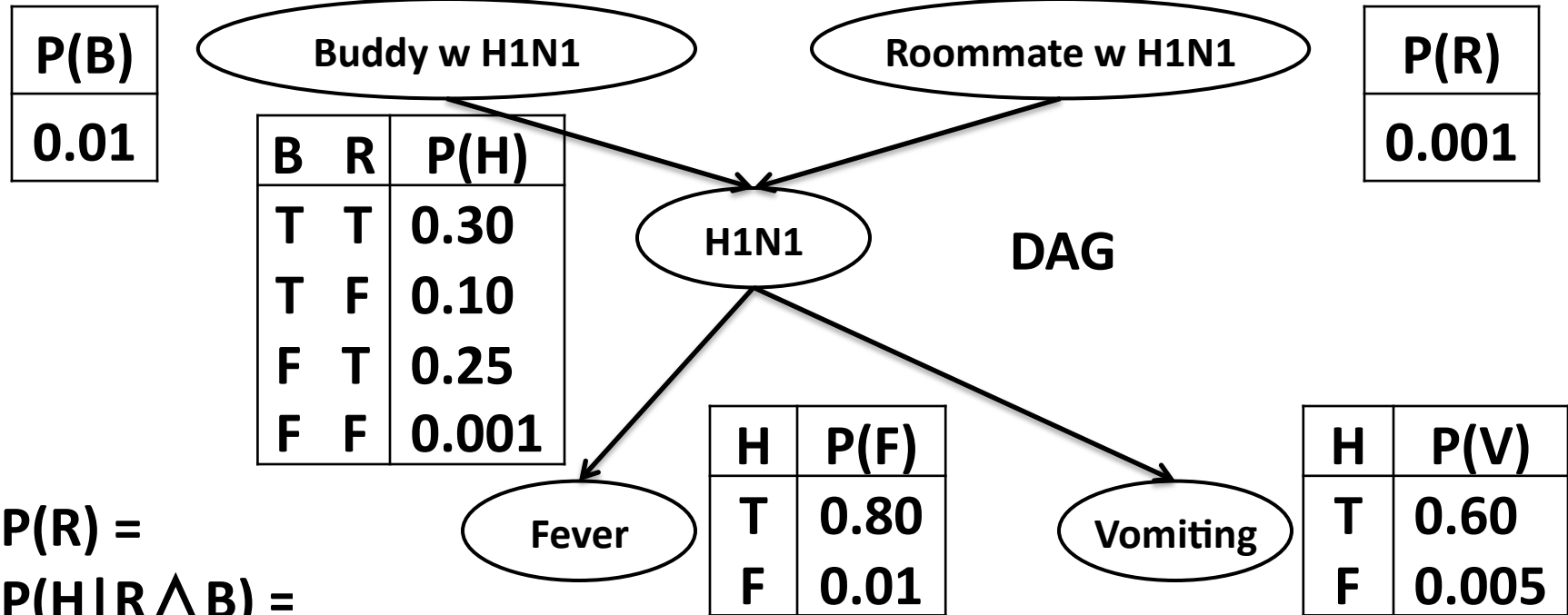
$$P(x_d | y) P(x_{d-1} | x_d, y) P(x_{d-2} | x_{d-1}, x_d, y) \cdots P(x_1 | x_2 \cdots x_d, y)$$

- **Naïve Bayes, which assumes all x_i are independent**

$$P(x_1, x_2 \cdots x_d | y) = P(x_1 | y) P(x_2 | y) \cdots P(x_d | y)$$

ML

Bayesian Network



$$P(R) =$$

$$P(H|R \wedge B) =$$

$$P(H|R) =$$

$$P(H) =$$

$$P(\neg F) =$$

$$P(H|F) =$$

$$P(R|F \wedge V) =$$

ML Building Bayesian Networks

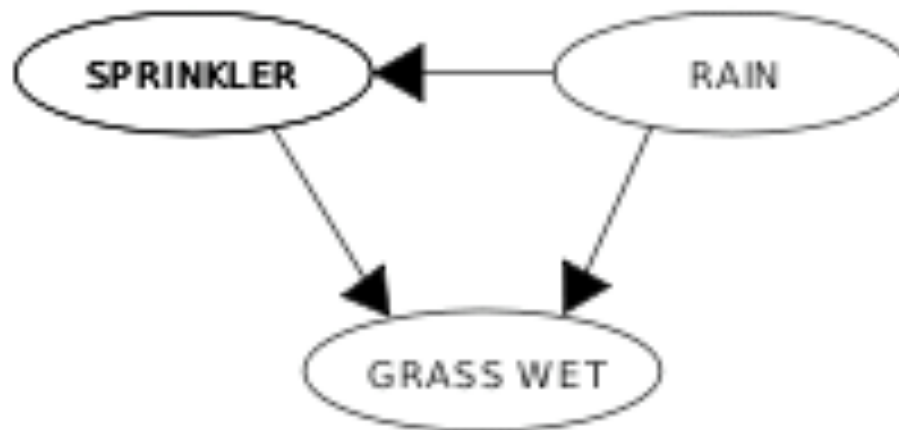
- **The network structure is usually built by hand**
- **If all variables are available in training data, use counts to create the conditional probability tables**
- **Otherwise can learn them by gradient ascent**
- **Methods also exist to automatically create the network structure**

ML **Further Reading**

- **Russell and Norvig. Various chapters and sections of: *Artificial Intelligence A modern approach*.**
- **Bishop. Chpt 8, Pattern Recognition and Machine Learning.**

ML Bayesian Belief Networks

		SPRINKLER	
		T	F
RAIN	F	0.4	0.6
	T	0.01	0.99



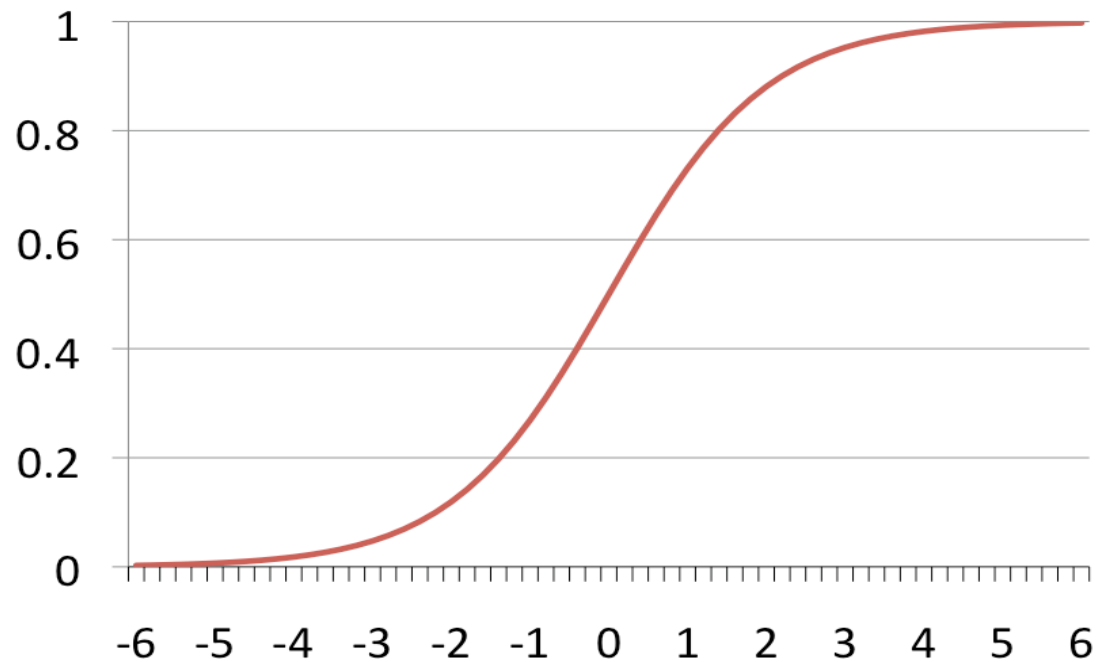
		RAIN	
		T	F
		0.2	0.8

		GRASS WET	
		T	F
SPRINKLER	F	0.0	1.0
	T	0.9	0.1
RAIN	F	0.8	0.2
	T	0.99	0.01

Logistic Regression

- AKA Logistic model or Logit model

$$f(\mathbf{x}) = \frac{1}{1 + e^{-g(\mathbf{x})}}$$



$$g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_dx_d$$

- Interpreted as a probability

ML Logistic Regression

- Assume y is a Boolean variable

$$f(\mathbf{x}) = \frac{1}{1 + e^{-g(\mathbf{x})}} = P(y = A | \mathbf{x})$$

$$g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_dx_d$$

$$\hat{y} = A \text{ if } \frac{1}{2} < \frac{1}{1 + e^{-g(\mathbf{x})}}, \quad 1 + e^{-g(\mathbf{x})} < 2, \quad e^{-g(\mathbf{x})} < 1$$

$$\ln(e^{-g(\mathbf{x})}) = -g(\mathbf{x}) < 0$$

$$\hat{y} = A \text{ if } g(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_dx_d > 0$$

ML Estimating the Parameters

- Under Gaussian Naïve Bayes assumption

$$w_j = \frac{\mu_{jA} - \mu_{jB}}{\sigma_j^2}$$

$$w_0 = \ln \frac{1 - \pi}{\pi} + \sum_j \frac{\mu_{jB}^2 - \mu_{jA}^2}{2\sigma_j^2}$$

- Alternatively, maximize conditional probability

$$\mathbf{w} \leftarrow \arg \max_{\mathbf{w}} \prod_i P(y^{(i)} | \mathbf{x}^{(i)}, \mathbf{w})$$

ML Estimating the Parameters

- **Log conditional probability**

$$\ln P\left(y^{(i)} \mid \mathbf{x}^{(i)}, \mathbf{w}\right) \equiv l(\mathbf{w})$$

$$= \sum_i \left(y^{(i)} \left(w_0 + \sum_{j=1}^d w_j x_j^{(i)} \right) - \ln \left(1 + \exp \left(w_0 + \sum_{j=1}^d w_j x_j^{(i)} \right) \right) \right)$$

- **No closed form solution**
 - Use gradient ascent

ML Estimating the Parameters

- **Gradient ascent**

$$\frac{\partial l(\mathbf{w})}{\partial w_j} = \sum_i x_j^{(i)} \left(y^{(i)} - \hat{P}(y^{(i)} = 1 | \mathbf{x}^{(i)}, \mathbf{w}) \right)$$

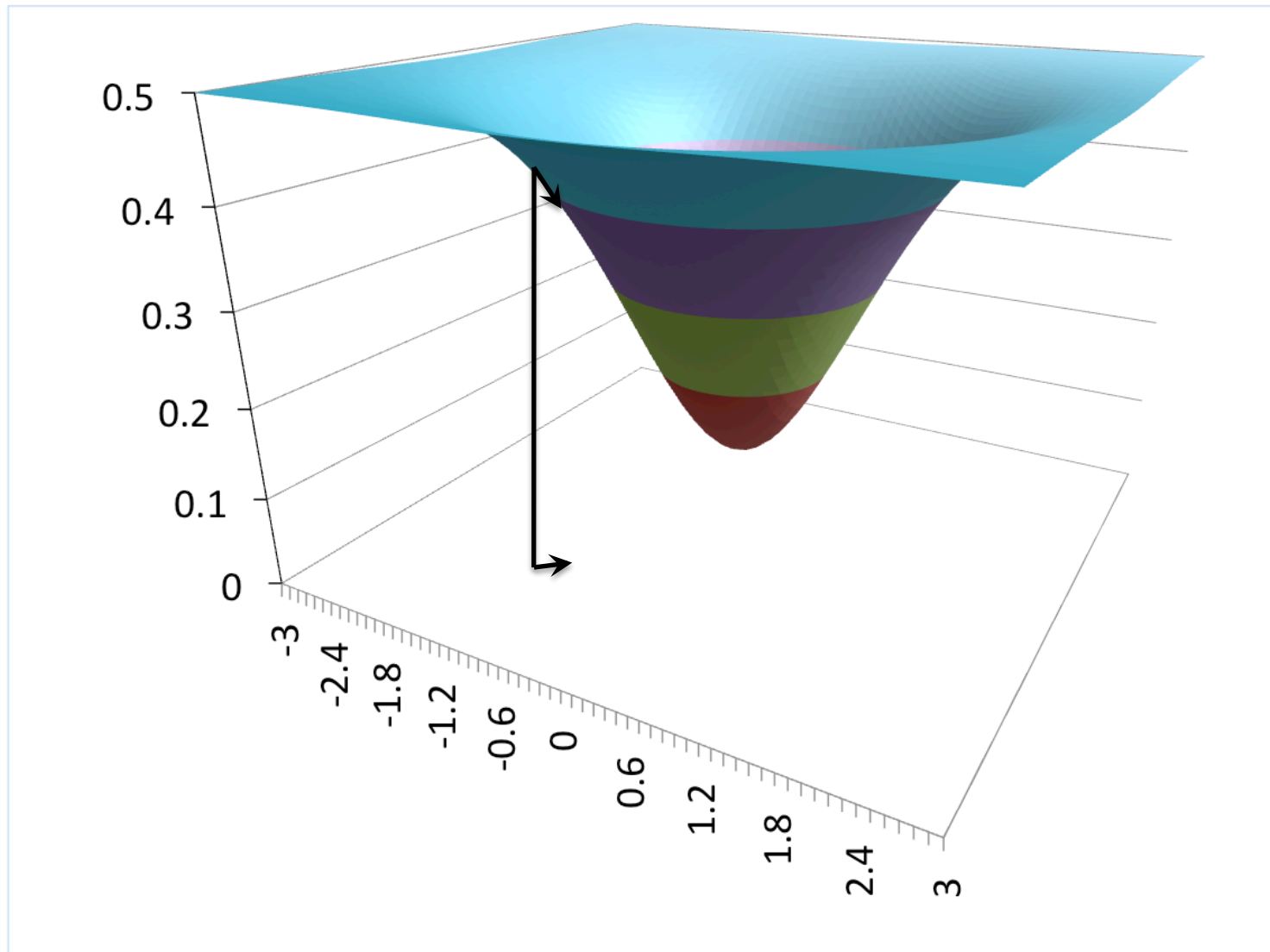
- **Parameter update after each instance \mathbf{x}**

$$w_j \leftarrow w_j + \eta \sum_i x_j^{(i)} \left(y^{(i)} - \hat{P}(y^{(i)} = 1 | \mathbf{x}^{(i)}, \mathbf{w}) \right)$$

– η is a small learning coefficient (step size), ~ 0.01

ML

Gradient Descent



ML Gradient Decent Parameter Est

- **Example:**

$$w_j \leftarrow w_j + \eta \sum_i x_j^{(i)} \left(y^{(i)} - \hat{P}(y^{(i)} = 1 | \mathbf{x}^{(i)}, \mathbf{w}) \right)$$

$$\mathbf{w} = \langle 0.02, 0.07, 0.01 \rangle, \mathbf{x} = \langle 3.0, 0.0 \rangle, y = 1$$

$$P(y = 1 | \mathbf{x}) = \frac{1}{1 + e^{-g(\mathbf{x})}} = \frac{1}{1 + e^{-(0.02 + 0.07 \cdot 3 + 0.01 \cdot 0)}} = 0.557$$

Regularization

- **Technique to avoid overfitting training data**
 - **Use log likelihood function that penalized large w**

$$\mathbf{w} \leftarrow \arg \max_{\mathbf{w}} \sum_i \ln P\left(y^{(i)} \mid \mathbf{x}^{(i)}, \mathbf{w}\right) - \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- **λ determines the general impact of the penalty**

$$w_j \leftarrow w_j + \eta \sum_i x_j^{(i)} \left(y^{(i)} - \hat{P}\left(y^{(i)} = 1 \mid \mathbf{x}^{(i)}, \mathbf{w}\right) \right) - \eta \lambda w_j$$

ML Logit for Non-Boolean $f(x)$

- Gradient ascent rule when there are K classes

$$w_{kj} \leftarrow w_{kj} + \eta \sum_i x_j^{(i)} \left(\delta(y^{(i)} = y_k) - \hat{P}(y^{(i)} = y_k | \mathbf{x}^{(i)}, \mathbf{w}) \right) - \eta \lambda w_{kj}$$

$$\delta(u = v) = 1 \text{ if } u = v, \text{ and } 0 \text{ otherwise}$$

ML Discriminative vs. Generative

- **Logistic Regression → Discriminative classifier**
 - Directly estimates the conditional probability of the class
$$P(y|\mathbf{x})$$

- **Naïve Bayes → Generative classifier**
 - Directly estimates the conditional probability of the data and the prior on the class

$$P(\mathbf{x}|y) \quad P(y)$$

ML Logistic Regression vs. Gaussian NB

- One version of GNB leads to parametric form used in Logistic Regression
- LR parameters w can be expressed in terms of GNB parameters μ and σ
- If GNB assumptions hold, LR and GNB converge to the same classifier as $N \rightarrow \infty$

ML Logistic Regression vs. Gaussian NB

- If GNB assumptions do NOT hold, LR and GNB produce different classifiers
 - Asymptotically Logistic Regression is often better
 - LR does not strictly adhere to conditional independence assumptions $P(x_1 | x_2, y) = P(x_1 | y)$
- Convergence toward asymptotic accuracy is at different rates
 - GNB in $\log(d)$ & LR in d , where d is dimension of x
 - Hence, GNB often better w small N and vice versa