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

Bayesian Features

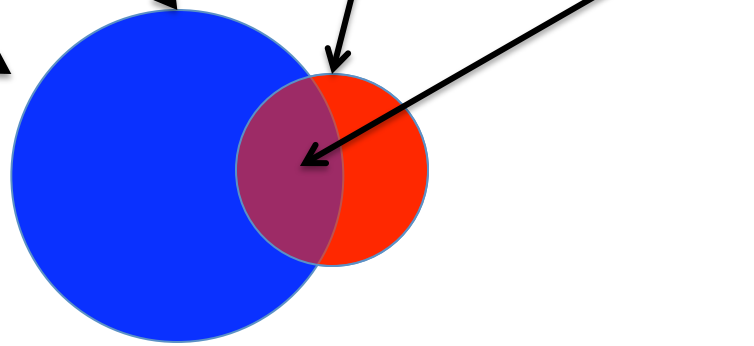
- **Adjust probability that a given hypothesis is correct vs. eliminating hypotheses**
- **Can easily incorporate prior knowledge about probabilities**
- **Can classify according to multiple weighted *h***
- **When computationally intractable, can still provide standard**

Basic Probability Rules

- **Sum rule**
  - ***P*(Glasses) = 0.3**
  - ***P*(Sweater) = 0.1**
  - ***P*(Glasses ∨ Sweater) = ?**

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

$^{7}/_{20} = 0.35$ $^{3}/_{10}$ $^{1}/_{10}$ $^{1}/_{20}$

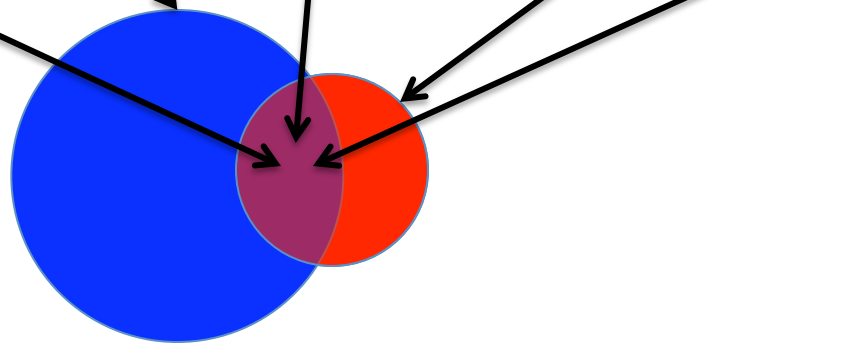


Basic Probability Rules

- **Product rule**
  - ***P*(Glasses) = 0.3**
  - ***P*(Sweater) = 0.1**
  - ***P*(Glasses ∧ Sweater) = ?**

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

$^1/_{20} = 0.05$ $^3/_{10}$ $^1/_6$ $^1/_{10}$ $^1/_2$

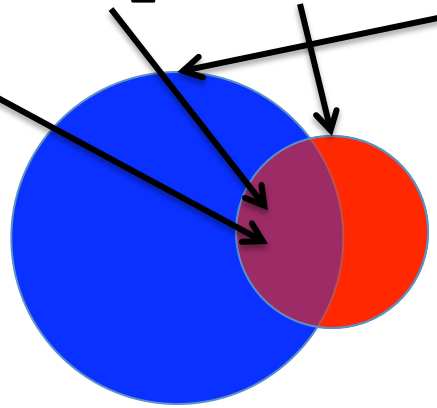


Basic Probability Rules

- **Bayes Theorem**
  - ***P*(Glasses) = 0.3**
  - ***P*(Sweater) = 0.1**
  - ***P*(Glasses | Sweater) = ?**

$$P(\mathrm{S}|\mathrm{G}) = P(\mathrm{G}|\mathrm{S})P(\mathrm{S})/P(\mathrm{G})$$

$^1/_6 = 0.167$ $^1/_2$ $^1/_{10}$ $^3/_{10}$



Basic Probability Rules

- **Theorem of total probability**
  - ***P*(Glasses | Sweater) = 0.4; *P*(Sweater) = 0.1**
  - ***P*(Glasses | Shirt) = 0.2; *P*(Shirt) = 0.6**
  - ***P*(Glasses | WetSuit) = 0.0; *P*(WetSuit) = 0.1**
  - ***P*(Glasses | Other) = 0.1; *P*(Other) = 0.2**

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

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

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

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

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**

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

$$= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D|h)P(h)$$

- **Maximum likelihood hypothesis**

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

Naïve Bayes Classifier

- **Assume independence of attributes given the class**

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

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

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

Conditional Independence

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

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

Bayesian Belief Networks

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

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

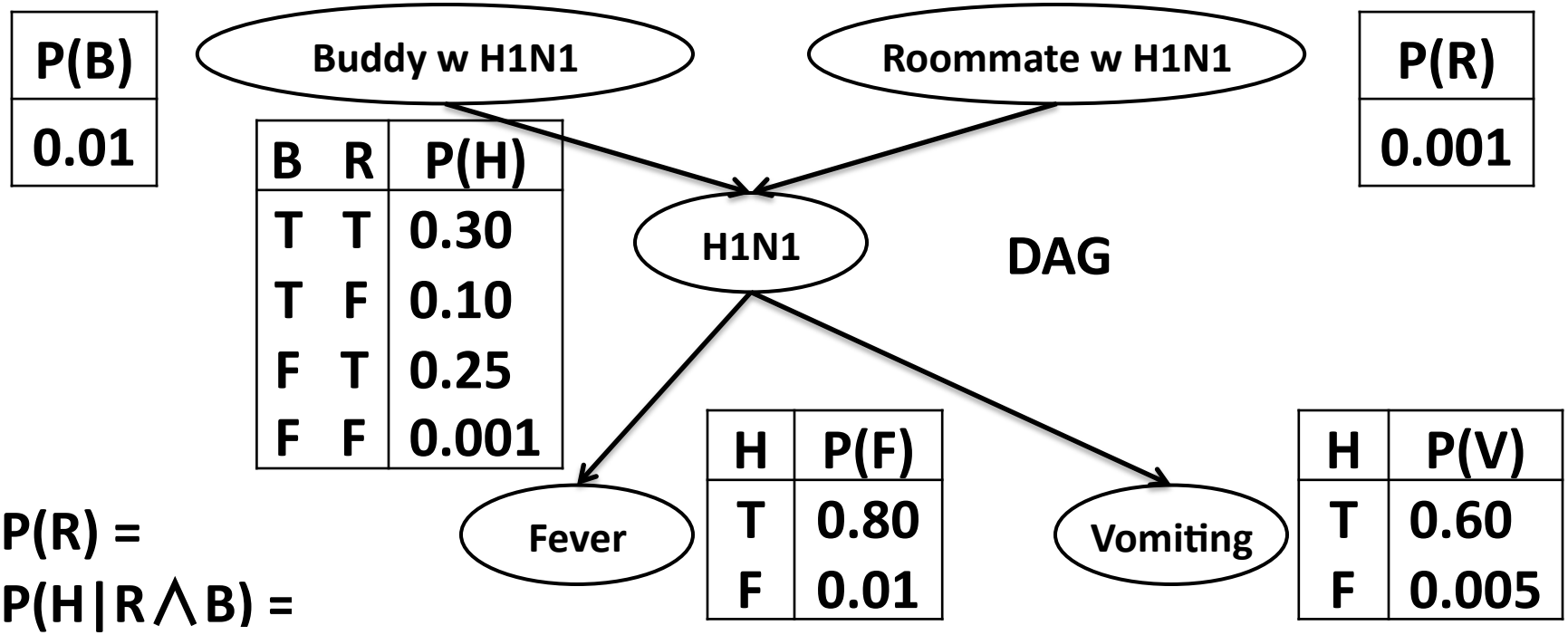
$$P\left(x_d \middle| y\right) P\left(x_{d-1} \middle| x_d, y\right) P\left(x_{d-2} \middle| x_{d-1}, x_d, y\right) \cdots P\left(x_1 \middle| x_2 \cdots x_d, y\right)$$

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

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

ML

Bayesian Network



P(R) =

P(H|R∧B) =

P(H|R) =

P(H) =

P(⊘F) =

P(H|F) =

P(R|F∧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

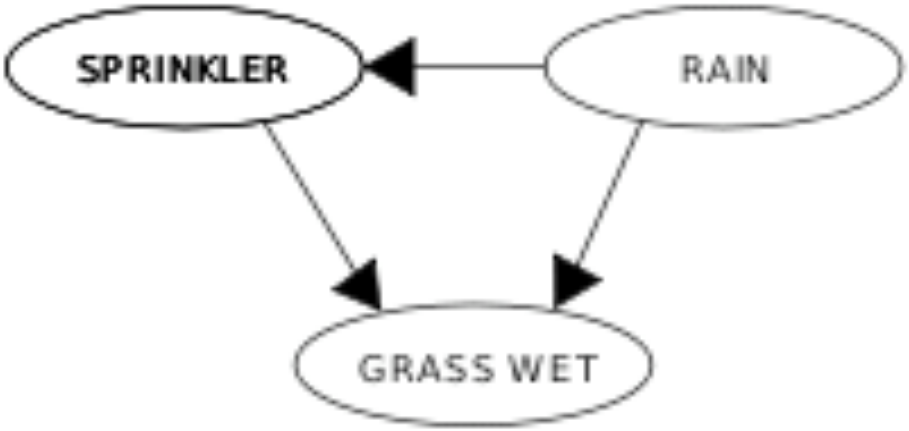
ML

Further Reading

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

Bayesian Belief Networks

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



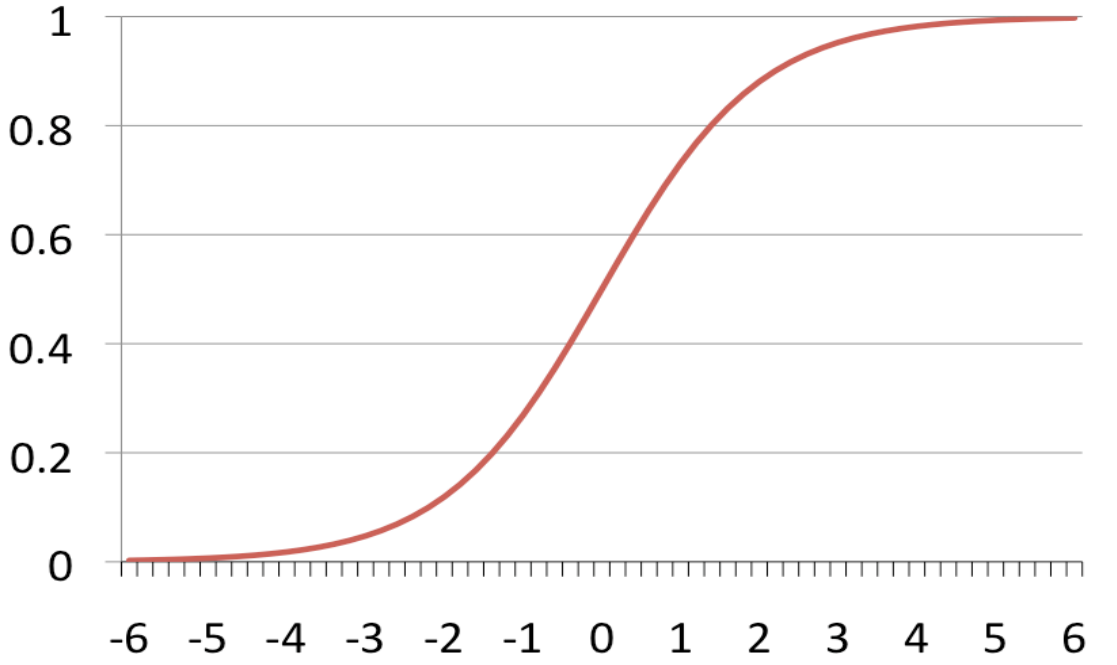
	RAIN	
T	F	
0.2	0.8	

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

Logistic Regression

- AKA Logistic model or Logit model

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



$$g(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

- Interpreted as a probability

Logistic Regression

- **Assume *y* is a Boolean variable**

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

$$g(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

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

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

$$\hat{y} = \mathrm{A} \;\; \text{if} \;\; g(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d > 0$$

Estimating the Parameters

- **Under Gaussian Naïve Bayes assumption**

$$w_j = \frac{\mu_{j\mathrm{A}} - \mu_{j\mathrm{B}}}{\sigma_j^2}$$

$$w_0 = \ln\frac{1-\pi}{\pi} + \sum_j \frac{\mu_{j\mathrm{B}}^2 - \mu_{j\mathrm{A}}^2}{2\sigma_j^2}$$

- **Alternatively, maximize conditional probability**

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

ML

Estimating the Parameters

- **Log conditional probability**

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

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

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

Estimating the Parameters

- **Gradient ascent**

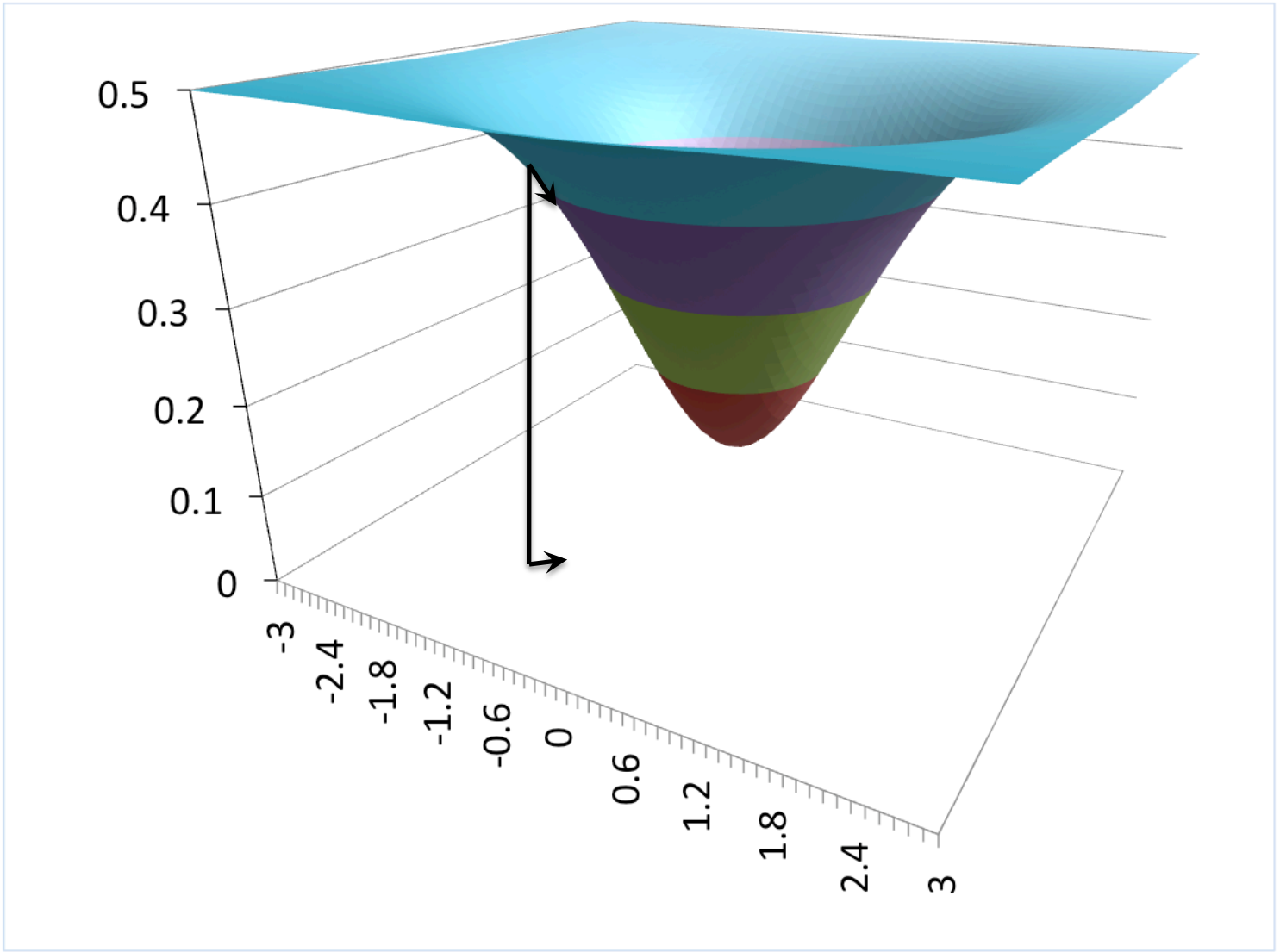
$$\frac{\partial l(\mathbf{w})}{\partial w_j} = \sum_i x_j^{(i)} \left(y^{(i)} - \hat{P}\left(y^{(i)} = 1 \middle| \mathbf{x}^{(i)}, \mathbf{w} \right) \right)$$

- **Parameter update after each instance x**

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

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

Gradient Descent



Gradient Decent Parameter Est

- **Example:**

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

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

$$P\left(y = 1 \middle| \mathbf{x} \right) = \frac{1}{1 + e^{-g(\mathbf{x})}} = \frac{1}{1 + e^{-(0.02 + .07 \cdot 3 + .01 \cdot 0)}} = 0.557$$

Regularization

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

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

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

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

Logit for Non-Boolean *f*(*x*)

- **Gradient ascent rule when there are *K* classes**

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

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

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. Gausian 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. Gausian 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**