Recap: Naïve Bayes classifier

- $f(X) = argmax_y P(y|X)$
  $= argmax_y P(X|y)P(y)$
  $= argmax_y \prod_{i=1}^{V} P(x_i|y) P(y)$

Class conditional density — Class prior

#parameters: $|Y| \times V$ — $|Y| - 1$

v.s.

$|Y| \times (2^V - 1)$

Computationally feasible

Logistic Regression

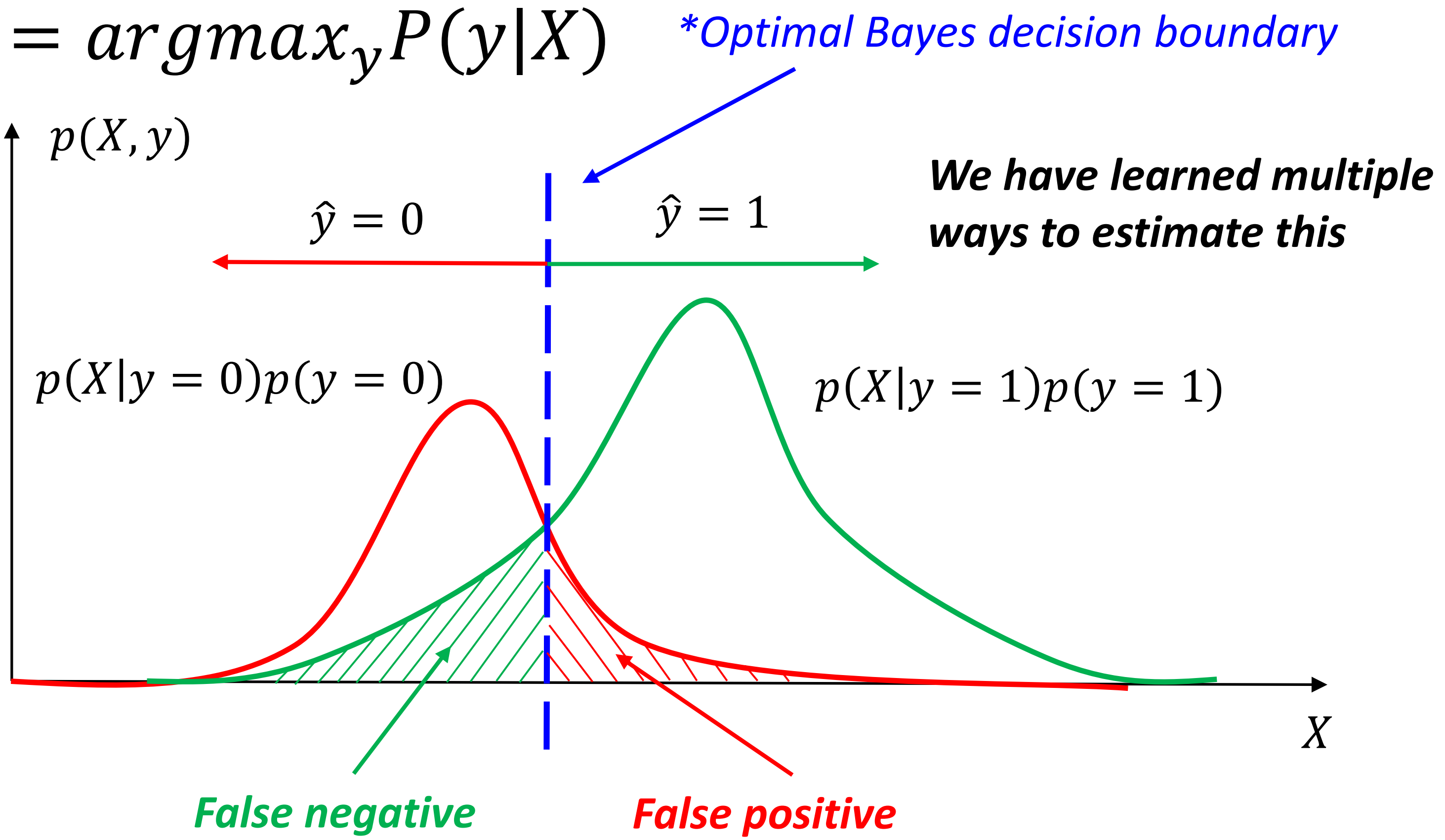
Hongning Wang
CS@UVa

Today's lecture

- Logistic regression model
  – A discriminative classification model
  – Two different perspectives to derive the model
  – Parameter estimation

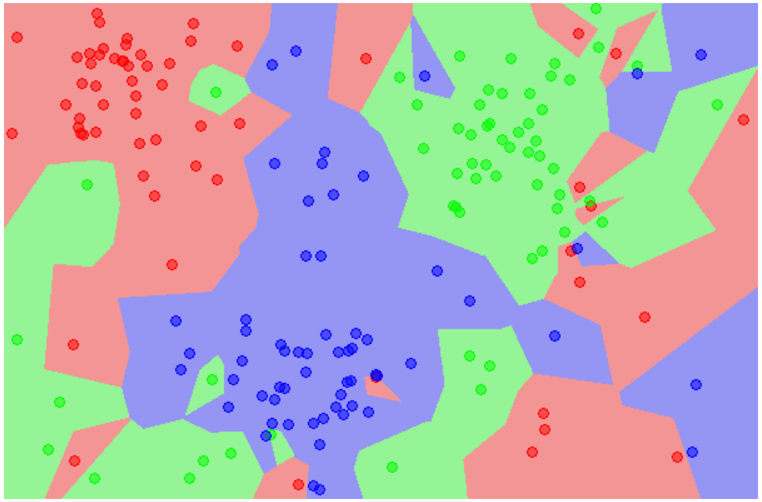
Review: Bayes risk minimization

- Risk – assign instance to a wrong class
  - $y^* = argmax_y P(y|X)$



Instance-based solution

- k nearest neighbors
  - Approximate Bayes decision rule in a subset of data around the testing point



Instance-based solution

- k nearest neighbors
  - Approximate Bayes decision rule in a subset of data around the testing point
  - Let V be the volume of the m dimensional ball around x containing the k nearest neighbors for x, we have

$$p(x)V = \frac{k}{N} \Rightarrow p(x) = \frac{k}{NV} \qquad p(x|y=1) = \frac{k_1}{N_1 V} \qquad p(y=1) = \frac{N_1}{N}$$

Total number of instances

Total number of instances in class 1

With Bayes rule:

$$p(y=1|x) = \frac{\frac{N_1}{N} \times \frac{k_1}{N_1 V}}{\frac{k}{NV}} = \frac{k_1}{k}$$

Counting the nearest neighbors from class1

Generative solution

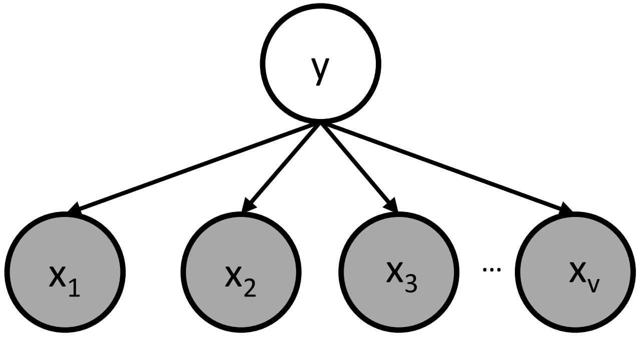
- Naïve Bayes classifier
  - $y^* = argmax_y P(y|X)$

$$= argmax_y P(X|y)P(y)$$

By Bayes rule

$$= argmax_y \prod_{i=1}^{|d|} P(x_i|y)\, P(y)$$

By independence assumption



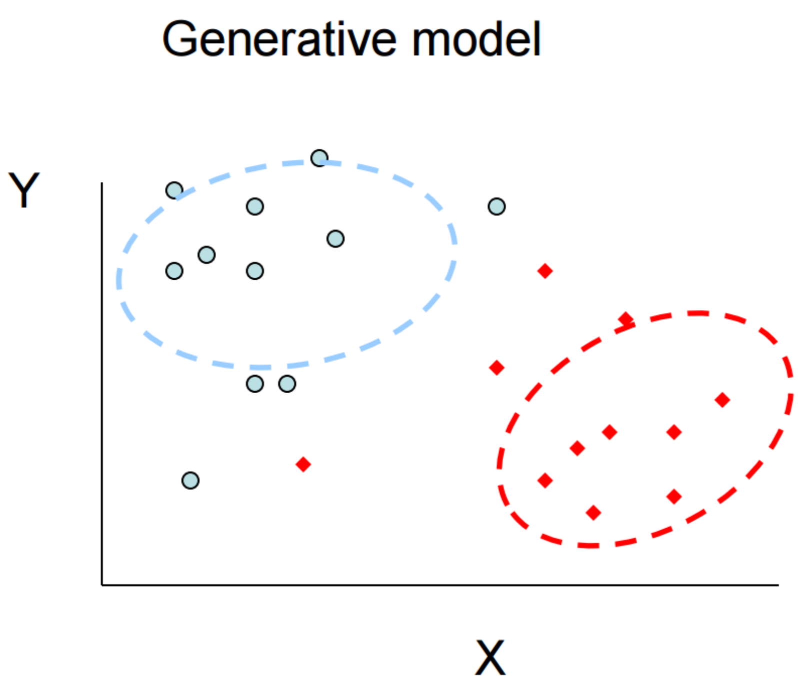
Estimating parameters

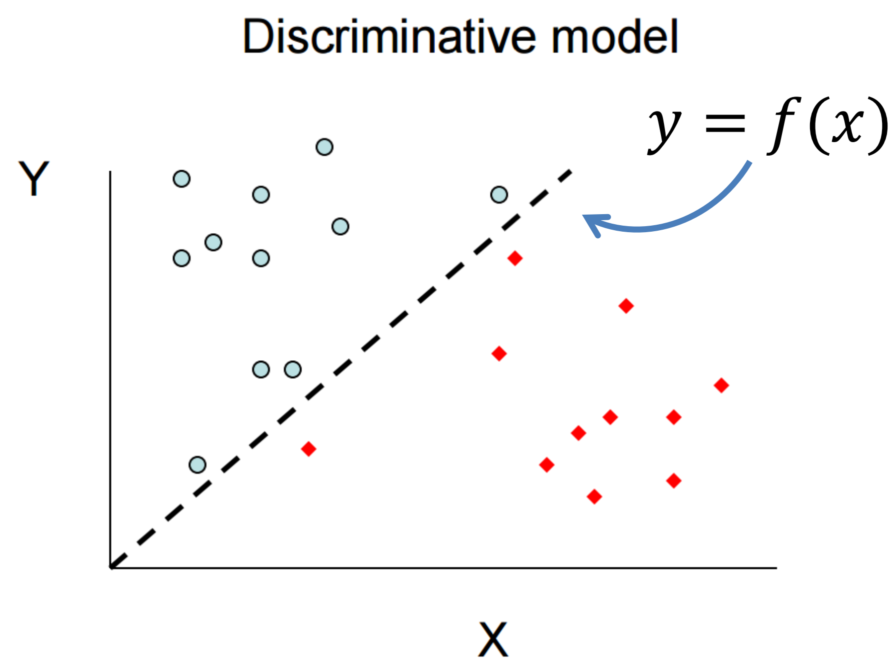
- Maximial likelihood estimator
  - $P(x_i|y) = \frac{\sum_d \sum_j \delta(x_d^j = x_i, y_d = y)}{\sum_d \delta(y_d = y)}$
  - $P(y) = \frac{\sum_d \delta(y_d = y)}{\sum_d 1}$

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious	**Y**
D1	1	1	1	1	0	1	1	1	0	0	0	**1**
D2	1	1	0	0	1	1	1	0	1	0	0	**1**
D3	0	0	0	0	0	1	0	0	0	1	1	**0**

Discriminative v.s. generative models

All instances are considered for probability density estimation





More attention will be put onto the boundary points

Parametric form of decision boundary in Naïve Bayes

- For binary cases
  - $f(X) = sgn(\log P(y=1|X) - \log P(y=0|X))$

$$= sgn\left(\log \frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{|d|} c(x_i, d) \log \frac{P(x_i|y=1)}{P(x_i|y=0)}\right)$$

$$= sgn(w^T \bar{X})$$

Linear regression?

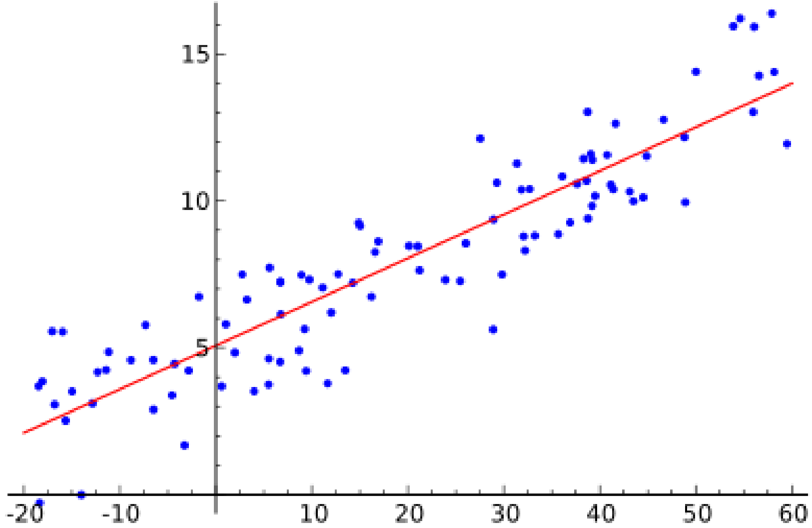
where

$$w = \left(\log \frac{P(y=1)}{P(y=0)}, \log \frac{P(x_1|y=1)}{P(x_1|y=0)}, \dots, \log \frac{P(x_v|y=1)}{P(x_v|y=0)}\right)$$

$$\bar{X} = (1, c(x_1, d), \dots, c(x_v, d))$$

Regression for classification?

- Linear regression
  - $y \leftarrow w^T X$
  - Relationship between a <u>scalar</u> dependent variable y and one or more explanatory variables

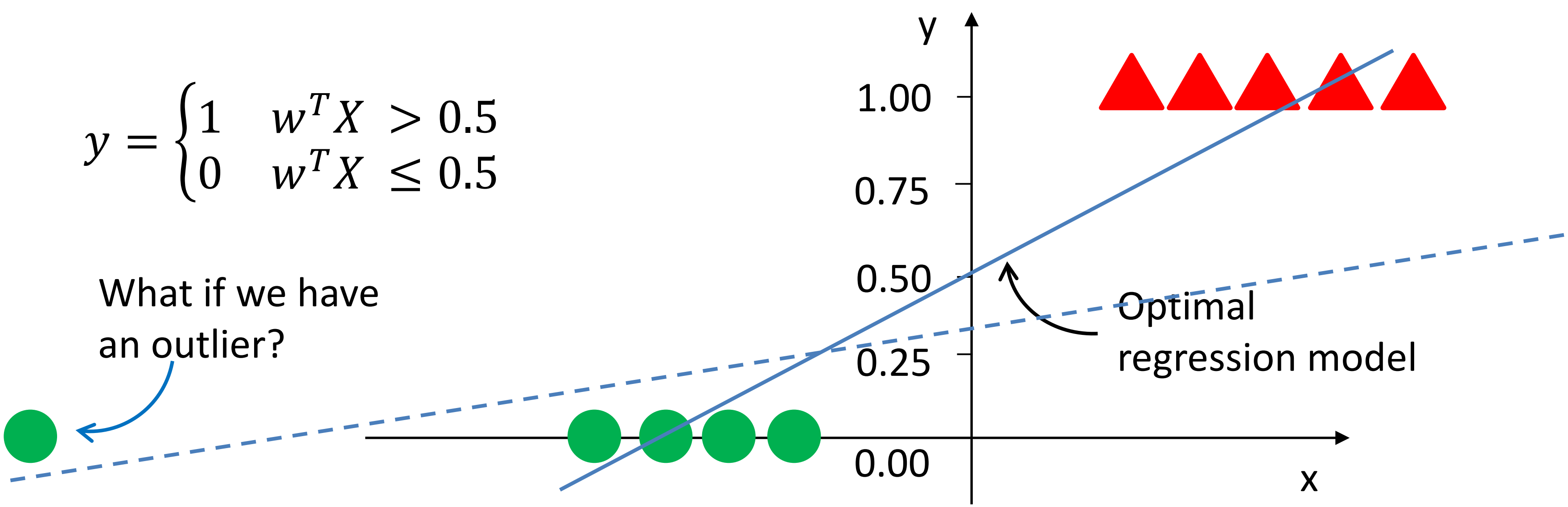


Regression for classification?

- Linear regression
  - $y \leftarrow w^T X$
  - Relationship between a <u>scalar</u> dependent variable y and one or more explanatory variables

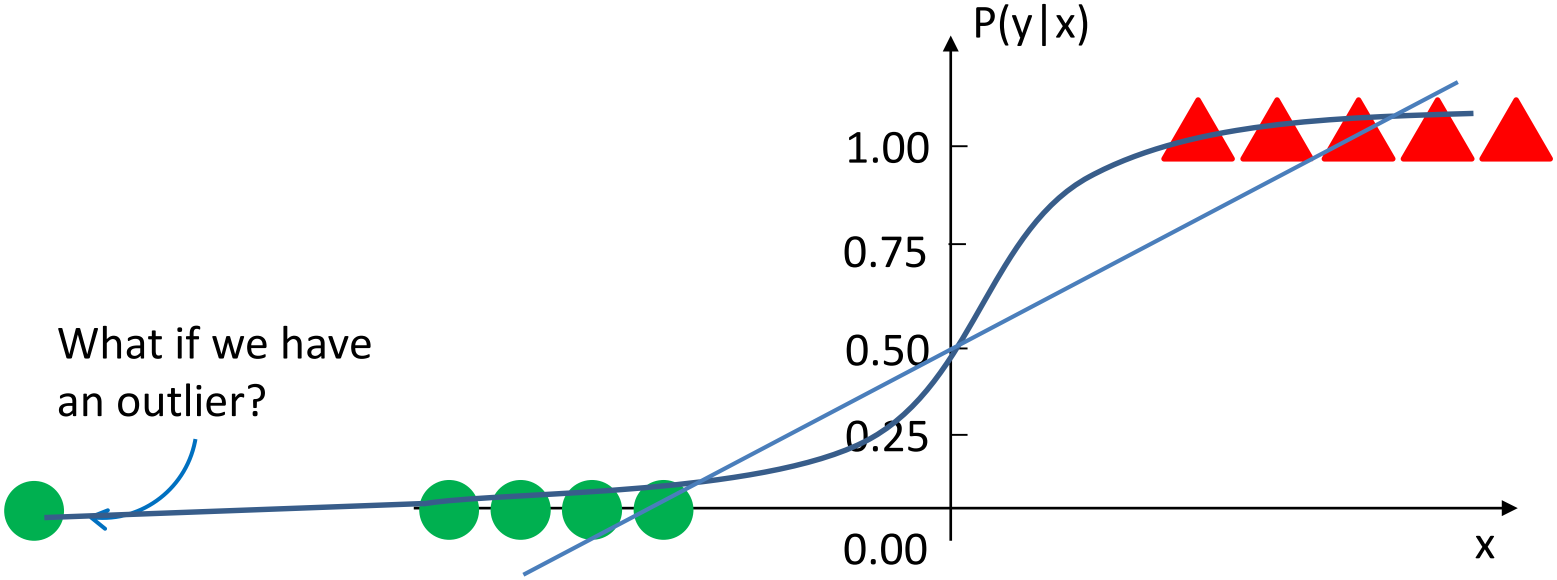
Y is discrete in a classification problem!

$$y = \begin{cases} 1 & w^T X > 0.5 \\ 0 & w^T X \leq 0.5 \end{cases}$$



Regression for classification?

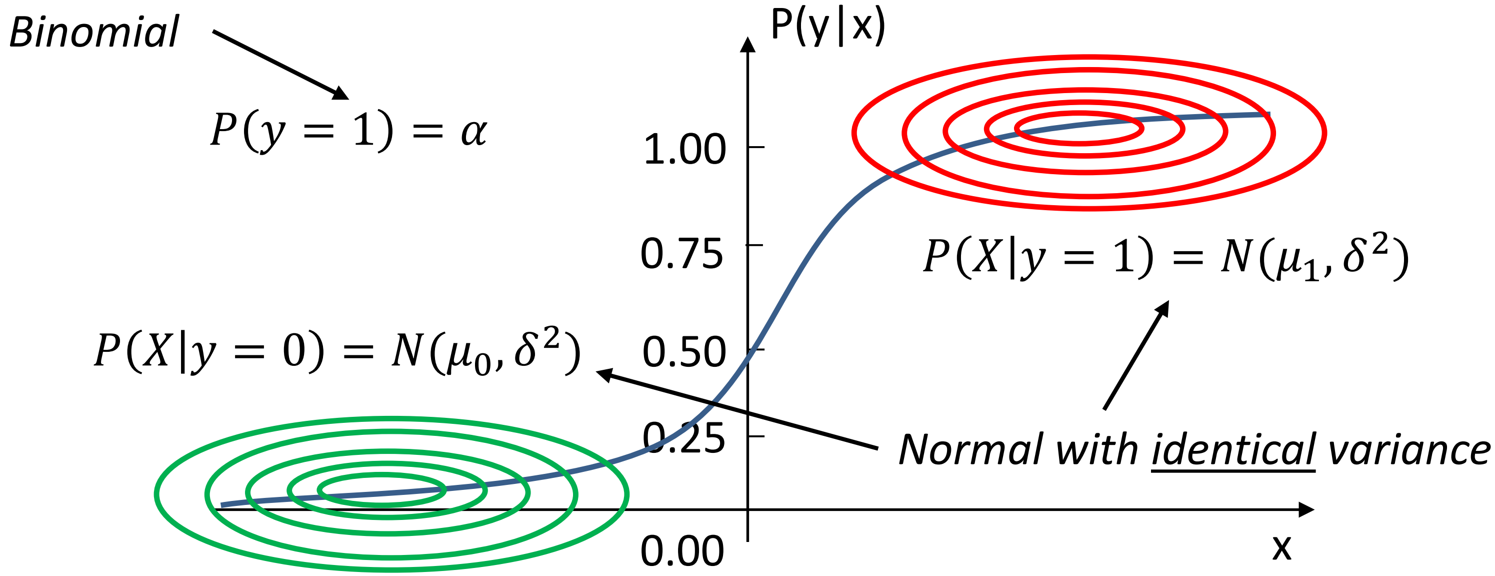
- Logistic regression
  - $p(y|x) = \sigma(w^T X) = \frac{1}{1+\exp(-w^T X)}$ (Sigmoid function)
  - Directly modeling of class posterior



Logistic regression for classification

- Why sigmoid function?
  - $P(y=1|X) = \frac{P(X|y=1)P(y=1)}{P(X|y=1)P(y=1)+P(X|y=0)P(y=0)}$

$$= \frac{1}{1+\frac{P(X|y=0)P(y=0)}{P(X|y=1)P(y=1)}}$$



Logistic regression for classification

- Why sigmoid function?
  - $P(y=1|X) = \frac{P(X|y=1)P(y=1)}{P(X|y=1)P(y=1)+P(X|y=0)P(y=0)}$

$$= \frac{1}{1+\frac{P(X|y=0)P(y=0)}{P(X|y=1)P(y=1)}}$$

$$= \frac{1}{1+\exp\left(-\ln\frac{P(X|y=1)P(y=1)}{P(X|y=0)P(y=0)}\right)}$$

Logistic regression for classification

- Why sigmoid function?

$$P(x|y) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\delta^2}}$$

  - $\ln \frac{P(X|y=1)P(y=1)}{P(X|y=0)P(y=0)} = \ln \frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{V} \ln \frac{P(x_i|y=1)}{P(x_i|y=0)}$

$$= \ln \frac{\alpha}{1-\alpha} + \sum_{i=1}^{V} \left(\frac{\mu_{1i} - \mu_{0i}}{\delta_i^2} x_i - \frac{\mu_{1i}^2 - \mu_{0i}^2}{2\delta_i^2} \right)$$

$$= w_0 + \sum_{i=1}^{V} \frac{\mu_{1i} - \mu_{0i}}{\delta_i^2} x_i$$

$$= w_0 + w^T X$$

$$= \bar{w}^T \bar{X}$$

Origin of the name: logit function

Logistic regression for classification

- Why sigmoid function?
  - $P(y=1|X) = \frac{P(X|y=1)P(y=1)}{P(X|y=1)P(y=1)+P(X|y=0)P(y=0)}$

$$= \frac{1}{1+\frac{P(X|y=0)P(y=0)}{P(X|y=1)P(y=1)}}$$

$$= \frac{1}{1+\exp\left(-\ln\frac{P(X|y=1)P(y=1)}{P(X|y=0)P(y=0)}\right)}$$

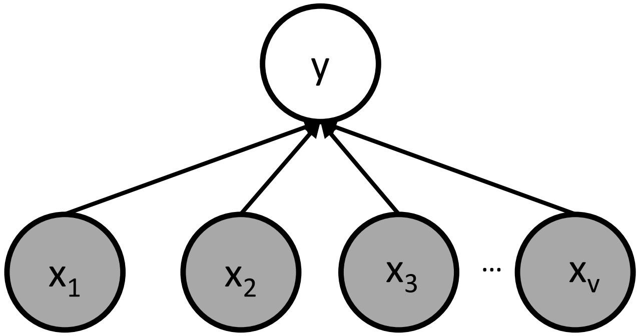
$$= \frac{1}{1+\exp(-\bar{w}^T\bar{X})}$$

Generalized Linear Model

Note: it is still a linear relation among the features!

Logistic regression for classification

- For multi-class categorization
  - $P(y=k|X)=\frac{\exp(w_k^T X)}{\sum_{j=1}^{K}\exp(w_j^T X)}$
  - $P(y=k|X) \propto \exp(w_k^T X)$



Warning: redundancy in model parameters,

When K=2,

$$P(y=1|X)=\frac{\exp(w_1^T X)}{\exp(w_1^T X)+\exp(w_0^T X)}$$
$$=\frac{1}{1+\exp(-(w_1-w_0)^T X)}$$

$\bar{w}$

Logistic regression for classification

- Decision boundary for binary case
  - $\hat{y} = \begin{cases} 1, p(y=1|X) > 0.5 \\ 0, \quad otherwise \end{cases}$

$$p(y=1|X) = \frac{1}{1+\exp(-w^T X)} > 0.5$$

i.f.f.

$$\exp(-w^T X) < 1$$

i.f.f.

$$w^T X > 0$$

  - $\hat{y} = \begin{cases} 1, \quad w^T x > 0 \\ 0, otherwise \end{cases}$ ← ***A linear model!***

Logistic regression for classification

- Decision boundary in general
  - $\hat{y} = argmax_y p(y|X)$

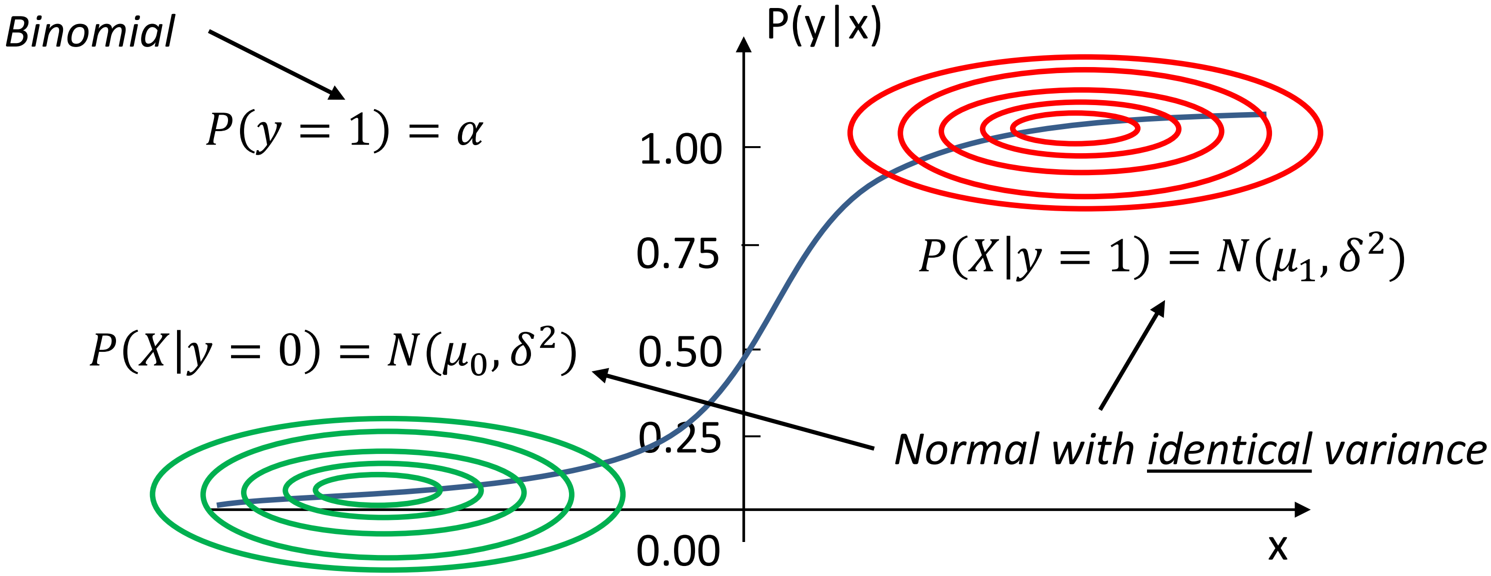
$$= argmax_y \exp(w_y^T X)$$

$$= argmax_y w_y^T X$$

A linear model!

Logistic regression for classification

- Summary
  - $P(y=1|X) = \frac{P(X|y=1)P(y=1)}{P(X|y=1)P(y=1)+P(X|y=0)P(y=0)}$
    $= \frac{1}{1+\frac{P(X|y=0)P(y=0)}{P(X|y=1)P(y=1)}}$



A different perspective

- Imagine we have the following

Documents	**Sentiment**
"happy", "good", "purchase", "item", "indeed"	positive

$$p(x = \text{"happy"}, y = 1) + p(x = \text{"good"}, y = 1) + p(x = \text{"}purchase\text{"}, y = 1)$$
$$+p(x = \text{"item"}, y = 1) + p(x = \text{"}indeed\text{"}, y = 1) = 1$$

Question: find a distribution $p(x, y)$ that satisfies this observation.

Answer1: $p(x = \text{"item"}, y = 1) = 1$, and all the others 0

Answer2: $p(x = \text{"indeed"}, y = 1) = 0.5$, $p(x = \text{"good"}, y = 1) = 0.5$, and all the others 0

We have too little information to favor either one of them.

Occam's razor

- A problem-solving principle
  - "among competing hypotheses that predict equally well, the one with the fewest assumptions should be selected."
    - William of Ockham (1287–1347)
  - Principle of Insufficient Reason: "when one has no information to distinguish between the probability of two events, the best strategy is to consider them equally likely"
    - Pierre-Simon Laplace (1749–1827)

A different perspective

- Imagine we have the following

Documents	Sentiment
“happy”, “good”, “purchase”, “item”, “indeed”	positive

$$p(x = \text{"happy"}, y = 1) + p(x = \text{"good"}, y = 1) + p(x = \text{"}purchase\text{"}, y = 1) \\ + p(x = \text{"item"}, y = 1) + p(x = \text{"}indeed\text{"}, y = 1) = 1$$

Question: find a distribution $p(x, y)$ that satisfies this observation.

As a result, a ***safer*** choice would be:

$$p(x = \text{"}\cdot\text{"}, y = 1) = 0.2$$

Equally favor every possibility

A different perspective

- Imagine we have the following

Observations	Sentiment
"happy", "good", "purchase", "item", "indeed"	positive
30% of time "good", "item"	positive

$$p(x = \text{"happy"}, y = 1) + p(x = \text{"good"}, y = 1) + p(x = "purchase", y = 1)$$
$$+p(x = \text{"item"}, y = 1) + p(x = "indeed", y = 1) = 1$$
$$p(x = \text{"good"}, y = 1) + p(x = \text{"item"}, y = 1) = 0.3$$

Question: find a distribution $p(x, y)$ that satisfies this observation.

Again, a ***safer*** choice would be:

$p(x = "good", y = 1) = p(x = "item", y = 1) = 0.15,$ and all the others $\frac{7}{30}$

Equally favor every possibility

A different perspective

- Imagine we have the following

Observations	Sentiment
"happy", "good", "purchase", "item", "indeed"	positive
30% of time "good", "item"	positive
50% of time "good", "happy"	positive

$$p(x = \text{"happy"}, y = 1) + p(x = \text{"good"}, y = 1) + p(x = \text{"}purchase\text{"}, y = 1)$$
$$+ p(x = \text{"item"}, y = 1) + p(x = \text{"}indeed\text{"}, y = 1) = 1$$

$$p(x = \text{"good"}, y = 1) + p(x = \text{"item"}, y = 1) = 0.3$$

$$p(x = \text{"good"}, y = 1) + p(x = \text{"happy"}, y = 1) = 0.5$$

Question: find a distribution $p(x, y)$ that satisfies this observation.

Time to think about:

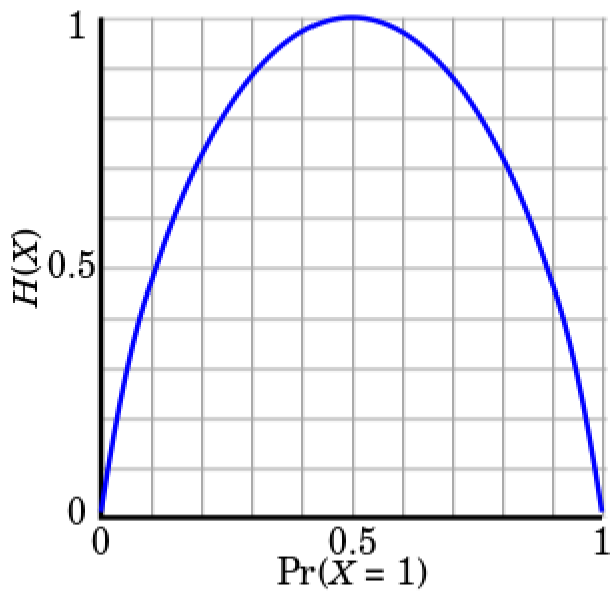
1) what do we mean by equally/uniformly favoring the models?

2) given all these constraints, how could we find the most preferred model?

Maximum entropy modeling

- A measure of uncertainty of random events
  - $H(X) = E[I(X)] = -\sum_{x \in X} P(x) \log P(x)$

Maximized when P(X) is uniform distribution



Question 1 is answered, then how about question 2?

Represent the constraints

- Indicator function
  - E.g., to express the observation that word ‘good’ occurs in a positive document
    - $f(x,y) = \begin{cases} 1 & \text{if } y = 1 \text{ and } x = \text{‘}good\text{’} \\ 0 & \text{otherwise} \end{cases}$
  - Usually referred as feature function

Represent the constraints

- Empirical expectation of feature function over a corpus
  - $E[\tilde{p}(f)] = \sum_{x,y} \tilde{p}(x,y)\, f(x,y)$

    where $\tilde{p}(x,y) = \frac{c(f(x,y))}{N}$ *i.e., frequency of observing $f(x,y)$ in a given collection.*
- Expectation of feature function under a given statistical model
  - $E[p(f)] = \sum_{x,y} \tilde{p}(x)\, p(y|x) f(x,y)$

Empirical distribution of x in the same collection. (pointing to $\tilde{p}(x)$)

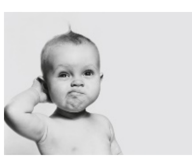
Model's estimation of conditional distribution. (pointing to $p(y|x)$)

Represent the constraints

- When a feature is important, we require our preferred statistical model to accord with it
  - $C := \{p \in P | E[p(f_i)] = E[\tilde{p}(f_i)], \forall i \in \{1,2,\dots,n\}\}$
  - $E[p(f_i)] = E[\tilde{p}(f_i)]$

$$\Rightarrow \sum_{x,y} \tilde{p}(x,y)\, f_i(x,y) = \sum_{x,y} \tilde{p}(x)\, p(y|x)\, f_i(x,y)$$

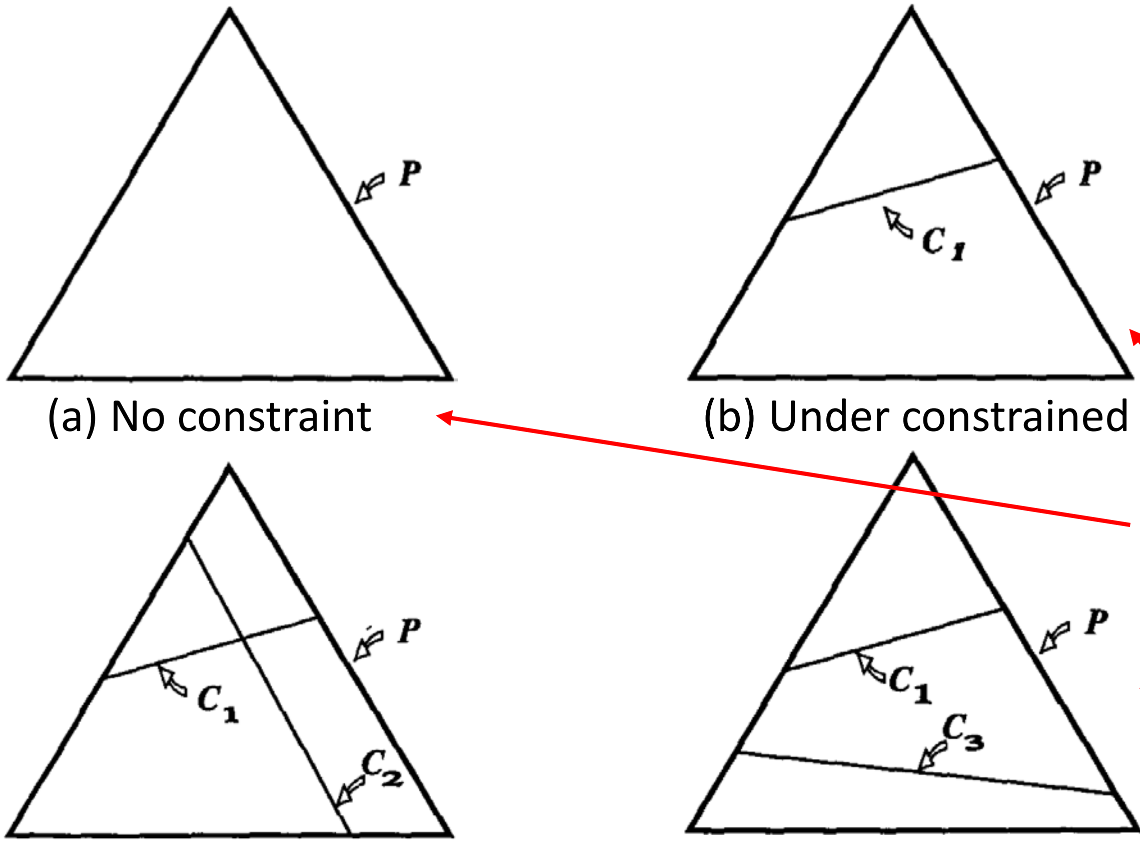
We only need to specify this in our preferred model!



Is Question 2 answered?

Represent the constraints

- Let's visualize this



(a) No constraint

(b) Under constrained

(c) Feasible constraint

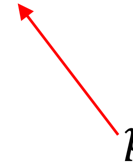
(d) Over constrained

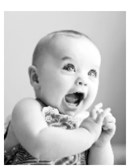
How to deal with these situations?

Maximum entropy principle

- To select a model from a set C of allowed probability distributions, choose the model $p^* \in C$ with maximum entropy $H(p)$

$$p^* = argmax_{p \in C} H(p)$$





Both questions are answered!

Maximum entropy principle

- Let's solve this constrained optimization problem with Lagrange multipliers

Primal:

$$p^* = argmax_{p \in C} H(p)$$

a strategy for finding the local maxima and minima of a function subject to equality constraints

Lagrangian:

$$L(p, \lambda) = H(p) + \sum_i \lambda_i (p(f_i) - \tilde{p}(f_i))$$

Maximum entropy principle

- Let's solve this constrained optimization problem with Lagrange multipliers

Lagrangian:

$$L(p,\lambda) = H(p) + \sum_i \lambda_i (p(f_i) - \tilde{p}(f_i))$$

Dual:

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

$$\Psi(\lambda) = -\sum_x \tilde{p}(x) \log Z_\lambda(x) + \sum_i \lambda_i \tilde{p}(f_i)$$

Maximum entropy principle

- Let's solve this constrained optimization problem with Lagrange multipliers

Dual:

$$\Psi(\lambda) = -\sum_x \tilde{p}(x) \log Z_\lambda(x) + \sum_i \lambda_i \, \tilde{p}(f_i)$$

where

$$\mathrm{Z}_\lambda = \sum\nolimits_y \exp\left(\sum_i \lambda_i f_i(x, y)\right)$$

Maximum entropy principle

- Let's take a close look at the dual function

$$\Psi(\lambda) = -\sum_{x} \tilde{p}(x) \log Z_{\lambda}(x) + \sum_{i} \lambda_i \tilde{p}(f_i)$$

where

$$\mathrm{Z}_{\lambda} = \sum\nolimits_{y} \exp\left(\sum_{i} \lambda_i f_i(x, y)\right)$$

Maximum entropy principle

- Let's take a close look at the dual function

$$\Psi(\lambda) = -\sum_x \tilde{p}(x) \boxed{\log Z_\lambda(x)} + \sum_x \tilde{p}(x) \boxed{\sum_i \lambda_i \tilde{p}(f_i)}$$

$$= \sum_x \tilde{p}(x) \log \frac{\exp(\sum_i \lambda_i \tilde{p}(f_i))}{Z_\lambda(x)}$$

$$= \sum_x \tilde{p}(x) \log p(y|x)$$

Maximum likelihood estimator!

Maximum entropy principle

- Primal: maximum entropy
  - $p^* = argmax_{p \in C} H(p)$
- Dual: logistic regression
  - $p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp(\sum_i \lambda_i f_i(x,y))$

where $$Z_\lambda = \sum_y \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

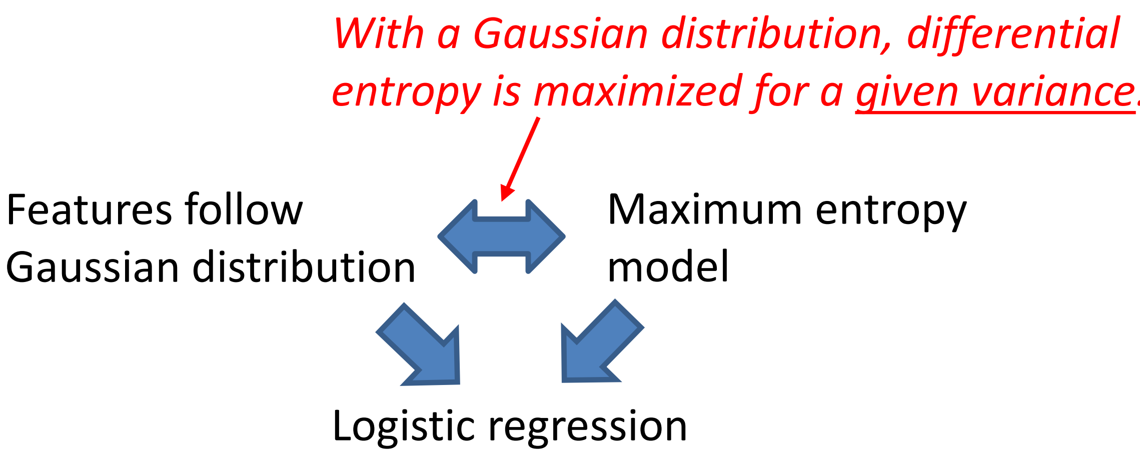
λ^* is determined by $\Psi(\lambda)$

Questions haven't been answered

- Class conditional density
  - Why it should be Gaussian with equal variance?
- Model parameters
  - What is the relationship between w and λ?
  - How to estimate them?

Maximum entropy principle

- The maximum entropy model subject to the constraints C has a parametric solution $p_{\lambda}^{*}(y|x)$ where the parameters λ^{*} can be determined by maximizing the likelihood function of $p_{\lambda}(y|x)$ over a training set



Parameter estimation

- Maximum likelihood estimation
  - $L(w) = \sum_{d \in D} y_d \log p(y_d = 1|X_d) + (1 - y_d) \log p(y_d = 0|X_d)$
  - Take gradient of $L(w)$ with respect to w

$$\frac{\partial L(w)}{\partial w} = \sum_{d \in D} y_d \boxed{\frac{\partial \log p(y_d = 1|X_d)}{\partial w}} + (1 - y_d) \boxed{\frac{\partial \log p(y_d = 0|X_d)}{\partial w}}$$

Parameter estimation

- Maximum likelihood estimation
  - $\frac{\partial \log p(y_d=1|X_d)}{\partial w} = -\frac{\partial \log\left(1+\exp(-w^T X_d)\right)}{\partial w}$
    $$= \frac{\exp(-w^T X_d)}{1+\exp(-w^T X_d)} X_d$$
    $$= (1 - p(y_d = 1|X_d))X_d$$
  - $\frac{\partial \log p(y_d=0|X_d)}{\partial w} = (0 - p(y_d = 1|X_d))X_d$

Parameter estimation

- Maximum likelihood estimation
  - $L(w) = \sum_{d \in D} y_d \log p(y_d = 1|X_d) + (1 - y_d) \log p(y_d = 0|X_d)$
  - Take gradient of $L(w)$ with respect to w

$$\frac{\partial L(w)}{\partial w} = \sum_{d \in D} y_d \boxed{\frac{\partial \log p(y_d = 1|X_d)}{\partial w}} + (1 - y_d) \boxed{\frac{\partial \log p(y_d = 0|X_d)}{\partial w}}$$

$$= \sum_{d \in D} y_d \big(1 - p(y_d = 1|X_d)\big) X_d + (1 - y_d) \big(0 - p(y_d = 1|X_d)\big) X_d$$

$$= \sum_{d \in D} \big(y_d - p(y = 1|X_d)\big) X_d$$

Good news: neat format, concave function for w

Bad news: no close form solution

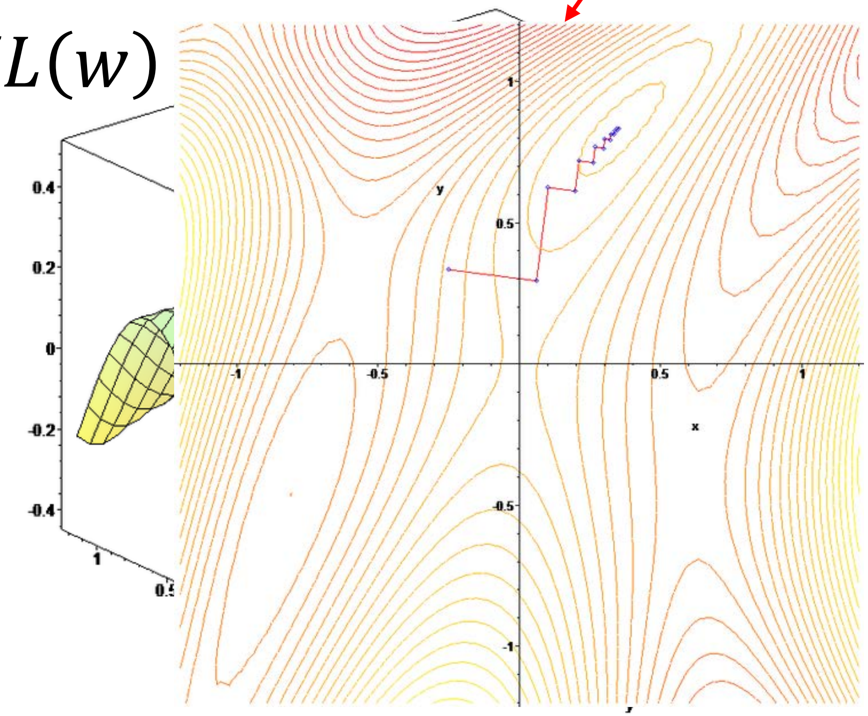
Can be easily generalized to multi-class case

Gradient-based optimization

- Gradient descent
  - $\nabla L(w) = [\frac{\partial L(w)}{\partial w_0}, \frac{\partial L(w)}{\partial w_1}, \dots, \frac{\partial L(w)}{\partial w_V}]$
  - $w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla L(w)$

Iterative updating

Step-size, affects convergence



Parameter estimation

- Stochastic gradient descent

while not converge

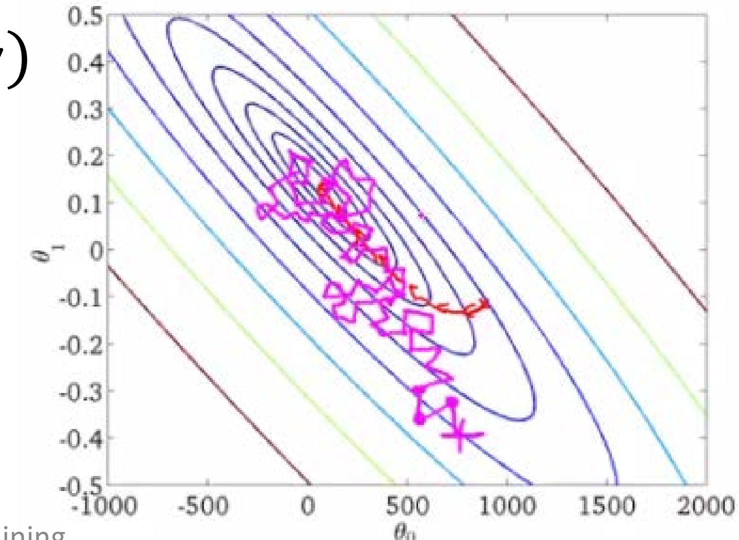
randomly choose $d \in D$

$$\nabla L_d(w) = [\frac{\partial L_d(w)}{\partial w_0}, \frac{\partial L_d(w)}{\partial w_1}, \dots, \frac{\partial L_d(w)}{\partial w_V}]$$

$$w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla L_d(w)$$

$$\eta^{(t+1)} = a\eta^{(t)}$$

Gradually shrink the step-size



Parameter estimation

- Batch gradient descent

while not converge

Compute gradient w.r.t. all training instances

$$\nabla L_D(w) = [\frac{\partial L_D(w)}{\partial w_0}, \frac{\partial L_D(w)}{\partial w_1}, \dots, \frac{\partial L_D(w)}{\partial w_V}]$$

Compute step size $\eta^{(t)}$

$$w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla L_d(w)$$

Line search is required to ensure sufficient decent

First order method

Second order methods, e.g., quasi-Newton method and conjugate gradient, provide faster convergence

Model regularization

- Avoid over-fitting
  - We may not have enough samples to well estimate model parameters for logistic regression
  - Regularization
    - Impose additional constraints over the model parameters
    - E.g., sparsity constraint – enforce the model to have more zero parameters

Model regularization

- L2 regularized logistic regression
  - Assume the model parameter w is drawn from Gaussian: w $\sim N(0, \sigma^2)$
  - $p(y_d, w|X_d) \propto p(y_d|X_d, w)p(w)$
  - $L(w) = \sum_{d \in D}[y_d \log p(y_d = 1|X_d) + (1 - y_d) \log p(y_d = 0|X_d)] - \frac{w^T w}{2\sigma^2}$

L2-norm of w

Generative V.S. discriminative models

Generative

- Specifying joint distribution
  - Full probabilistic specification for all the random variables
- Dependence assumption has to be specified for $p(X|y)$ and $p(y)$
- Flexible, can be used in unsupervised learning

Discriminative

- Specifying conditional distribution
  - Only explain the target variable
- Arbitrary features can be incorporated for modeling $p(y|X)$
- Need labeled data, only suitable for (semi-) supervised learning

Naïve Bayes V.S. Logistic regression

Naive Bayes

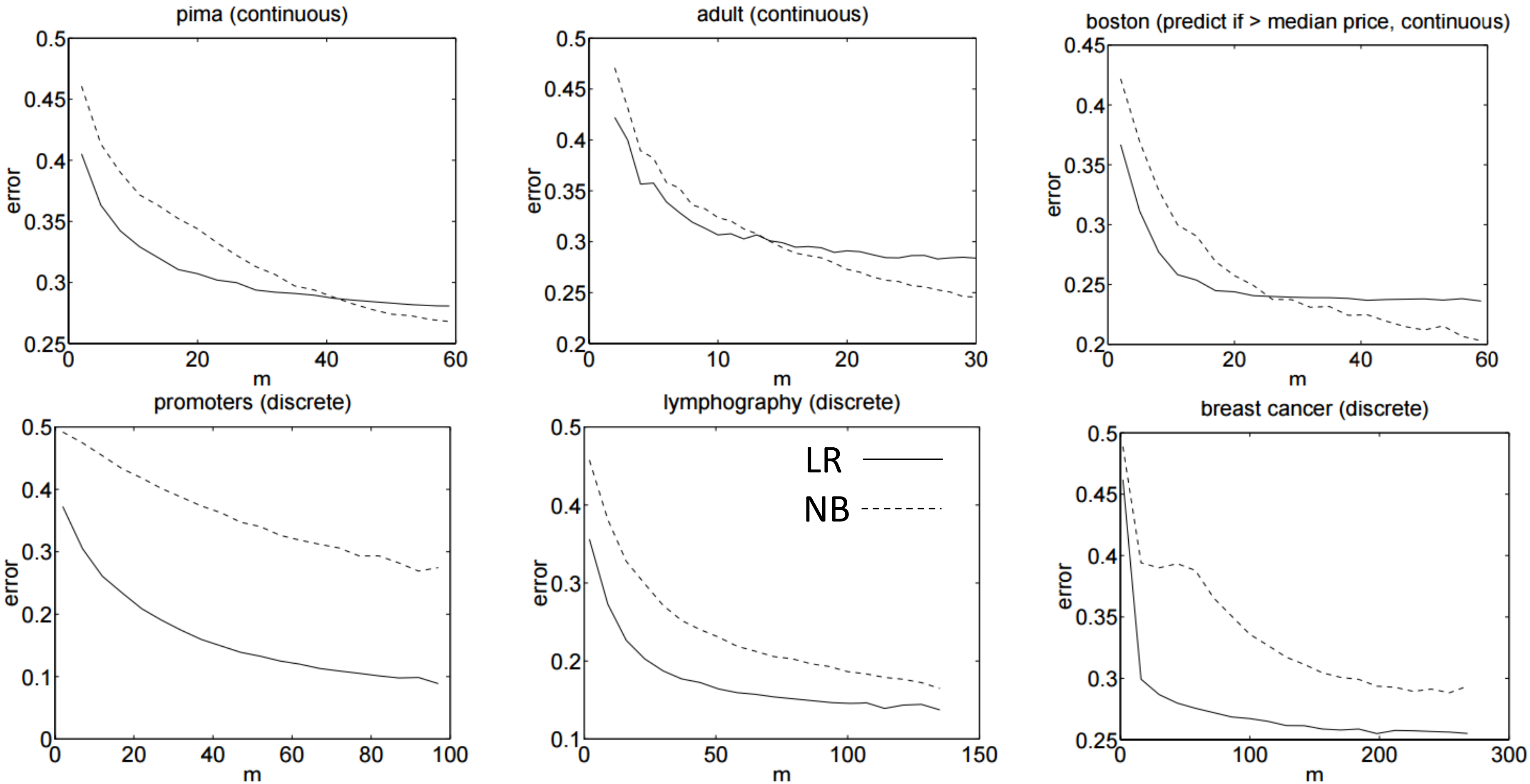
- Conditional independence
  - $p(X|y) = \prod_i p(x_i|y)$
- Distribution assumption of $p(x_i|y)$
- # parameters
  - $k(V+1)$
- Model estimation
  - Closed form MLE
- Asymptotic convergence rate
  - $\epsilon_{NB,n} \leq \epsilon_{NB,\infty} + O(\sqrt{\frac{\log V}{n}})$

Logistic Regression

- No independence assumption
- Functional form assumption of $p(y|X) \propto \exp(w_y^T X)$
- # parameters
  - $(k-1)(V+1)$
- Model estimation
  - Gradient-based MLE
- Asymptotic convergence rate
  - $\epsilon_{LR,n} \leq \epsilon_{LR,\infty} + O(\sqrt{\frac{V}{n}})$

Need more training data

Naïve Bayes V.S. Logistic regression



"On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." – Ng, Jordan NIPS 2002, UCI Data set

What you should know

- Two different derivations of logistic regression
  - Functional form from Naïve Bayes assumptions
    - $p(X|y)$ follows equal variance Gaussian
    - Sigmoid function
  - Maximum entropy principle
    - Primal/dual optimization
  - Generalization to multi-class
- Parameter estimation
  - Gradient-based optimization
  - Regularization
- Comparison with Naïve Bayes

Today's reading

- Speech and Language Processing
  - Chapter 6: Hidden Markov and Maximum Entropy Models
    - 6.6 Maximum entropy models: background
    - 6.7 Maximum entropy modeling