Logistic Regression

Required reading:

• Mitchell draft chapter (see course website)

Recommended reading:

• Bishop, Chapter 3.1.3, 3.1.4

• Ng and Jordan paper (see course website)

Machine Learning 10-701

Tom M. Mitchell
Center for Automated Learning and Discovery
Carnegie Mellon University

September 29, 2005

Generative vs. Discriminative Classifiers

Wish to learn f: X → Y, or P(Y|X)

Generative classifiers (e.g., Naïve Bayes):

- Assume some functional form for P(X|Y), P(Y)
  - This is the '*generative*' model
- Estimate parameters of P(X|Y), P(Y) directly from training data
- Use Bayes rule to calculate $P(Y|X= x_i)$

Discriminative classifiers:

- Assume some functional form for P(Y|X)
  - This is the '*discriminative*' model
- Estimate parameters of P(Y|X) directly from training data

- Consider learning f: X $\rightarrow$ Y, where
  - X is a vector of real-valued features, $< X_1 \ldots X_n >$
  - Y is boolean
- We could use a Gaussian Naïve Bayes classifier
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma)$
  - model P(Y) as Bernoulli (π)
- What does that imply about the form of P(Y|X)?

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, $< X_1 \dots X_n >$
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli (π)

Assume σ_{ik} indep of Y

- What does that imply about the form of P(Y|X)?

$$P(Y = 1|X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

Very convenient!

$$P(Y=1|X=<X_1,...X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)} = exp(w_0+\sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = w_0+\sum_i w_i X_i$$

Derive form for P(Y|X) for continuous X_i

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

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

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}}$$

$$P(x \mid y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \ e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$\boxed{\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2)}{2\sigma_i^2} \right)}$$

$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

Very convenient!

$$P(Y=1|X=<X_1,...X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

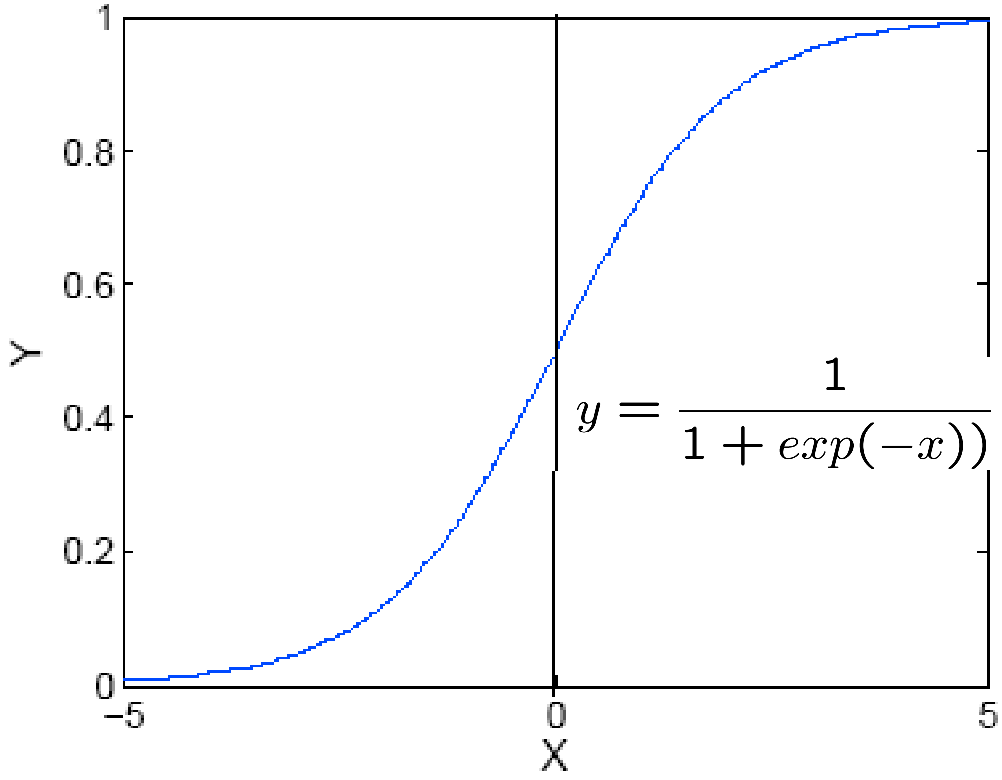
$$\frac{P(Y=0|X)}{P(Y=1|X)} = exp(w_0+\sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln \frac{P(Y=0|X)}{P(Y=1|X)} = w_0+\sum_i w_i X_i$$

Logistic function



$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Logistic regression more generally

- Logistic regression in more general case, where $Y \in \{Y_1 \ldots Y_R\}$: learn R-1 sets of weights

for $k<R$

$$P(Y = y_k|X) = \frac{\exp(w_{k0} + \sum_{i=1}^{n} w_{ki}X_i)}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

for $k=R$

$$P(Y = y_R|X) = \frac{1}{1 + \sum_{j=1}^{R-1} \exp(w_{j0} + \sum_{i=1}^{n} w_{ji}X_i)}$$

Training Logistic Regression: MCLE

- Choose parameters $W=<w_0, \ldots w_n>$ to <u>maximize conditional likelihood</u> of training data

where
$$P(Y=0|X,W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- Training data D = $\{\langle X^1, Y^1 \rangle, \ldots \langle X^L, Y^L \rangle\}$
- Data likelihood = $\prod_l P(X^l, Y^l | W)$
- Data conditional likelihood = $\prod_l P(Y^l | X^l, W)$

$$W \leftarrow \arg\max_W \ \ln \prod_l P(Y^l | X^l, W)$$

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$P(Y = 0|X, W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y = 1|X, W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

Maximizing Conditional Log Likelihood

$$P(Y=0|X,W) = \frac{1}{1+exp(w_0 + \sum_i w_i X_i)}$$

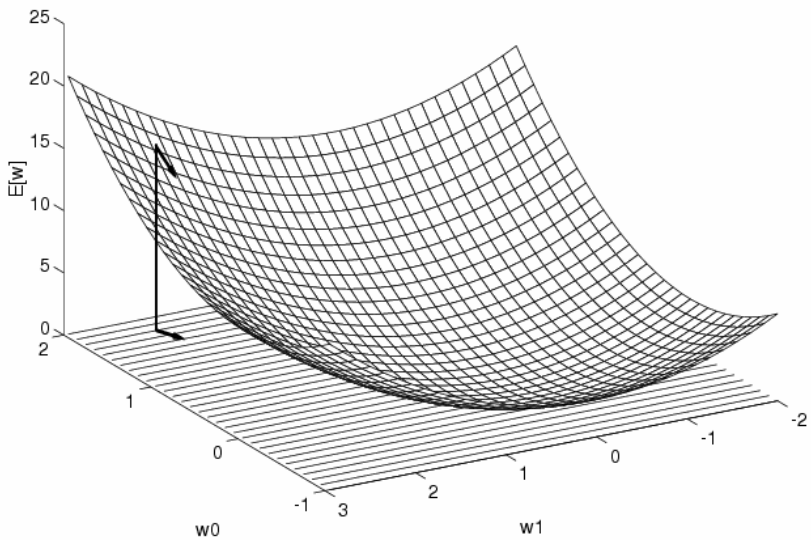
$$P(Y=1|X,W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1+exp(w_0 + \sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &\equiv \ln \prod_l P(Y^l|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

Good news: *l(W)* is concave function of *W*

Bad news: no closed-form solution to maximize *l(W)*

Gradient Descent



Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Maximize Conditional Log Likelihood: Gradient Ascent

$$
\begin{aligned}
l(W) &\equiv \ln \prod_l P(Y^l | X^l, W) \\
&= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))
\end{aligned}
$$

$$
\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1 | X^l, W))
$$

Gradient ascent algorithm: iterate until change < ε

For all i, $$w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1 | X^l, W))$$

repeat

That's all for M(C)LE. How about MAP?

- One common approach is to define priors on W
  - Normal distribution, zero mean, identity covariance
- Helps avoid very large weights and overfitting
- MAP estimate

$$W \leftarrow \arg\max_W \ \ln \ P(W|\{\langle Y^l, X^l \rangle\})$$

$$W \leftarrow \arg\max_W \ \ln \ P(W) \prod_l P(Y^l|X^l, W)$$

MLE vs MAP

- Maximum conditional likelihood estimate

$$W \leftarrow \arg\max_W \ \ln \prod_l P(Y^l|X^l, W)$$

$$\boxed{w_i \leftarrow w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

- Maximum a posteriori estimate

$$W \leftarrow \arg\max_W \ \ln[P(W) \ \prod_l P(Y^l|X^l, W)]$$

$$\boxed{w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))}$$

Generative vs. Discriminative Classifiers

Training classifiers involves estimating $f: X \rightarrow Y$, or $P(Y|X)$

Generative classifiers:

- Assume some functional form for $P(X|Y)$, $P(X)$
- Estimate parameters of $P(X|Y)$, $P(X)$ directly from training data
- Use Bayes rule to calculate $P(Y|X= x_i)$

Discriminative classifiers:

- Assume some functional form for $P(Y|X)$
- Estimate parameters of $P(Y|X)$ directly from training data

Naïve Bayes vs Logistic Regression

Consider Y boolean, X_i continuous, $X=<X_1 \ldots X_n>$

Number of parameters to estimate:

- NB:

- LR:

Naïve Bayes vs Logistic Regression

Consider Y boolean, X_i continuous, $X=<X_1 \ldots X_n>$

Number of parameters:

- NB: 4n +1
- LR: n+1

Estimation method:

- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

G.Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

- Generative and Discriminative classifiers

- Asymptotic comparison (# training examples → infinity)
  - when model assumptions correct
    - GNB, LR produce identical classifiers

  - when model assumptions incorrect
    - LR is less biased – does not assume cond indep.
    - therefore expected to outperform GNB

Naïve Bayes vs. Logistic Regression

- Generative and Discriminative classifiers

- Non-asymptotic analysis (see [Ng & Jordan, 2002])
  - convergence rate of parameter estimates – how many training examples needed to assure good estimates?
    - GNB order log n (where n = # of attributes in X)
    - LR order n

GNB converges more quickly to its (perhaps less helpful) asymptotic estimates

Rate of covergence: logistic regression

[Ng & Jordan, 2002]

Let $h_{Dis,m}$ be logistic regression trained on *m* examples in *n* dimensions. Then with high probability:

$$\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + O(\sqrt{\frac{n}{m} \log \frac{m}{n}})$$

Implication: if we want $\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + \epsilon_0$

for some constant ϵ_0, it suffices to pick order *n* examples

→ Convergences to its asymptotic classifier, in order *n* examples

(result follows from Vapnik's structural risk bound, plus fact that VCDim of *n* dimensional linear separators is *n*)

Rate of covergence: naïve Bayes parameters

[Ng & Jordan, 2002]

Let any $\epsilon_1, \delta > 0$ and any $l \geq 0$ be fixed. Assume that for some fixed $\rho_0 > 0$, we have that $\rho_0 \leq p(y = T) \leq 1 - \rho_0$. Let $m = O((1/\epsilon_1^2)\log(n/\delta))$. Then with probability at least $1-\delta$, after m examples:

1. For discrete inputs, $|\widehat{p}(x_i|y = b) - p(x_i|y = b)| \leq \epsilon_1$, and $|\widehat{p}(y = b) - p(y = b)| \leq \epsilon_1$, for all i, b.

2. For continuous inputs, $|\widehat{\mu}_{i|y=b} - \mu_{i|y=b}| \leq \epsilon_1$, and $|\widehat{\sigma}_i^2 - \sigma_i^2| \leq \epsilon_1$, for all i, b.

Some experiments from UCI data sets

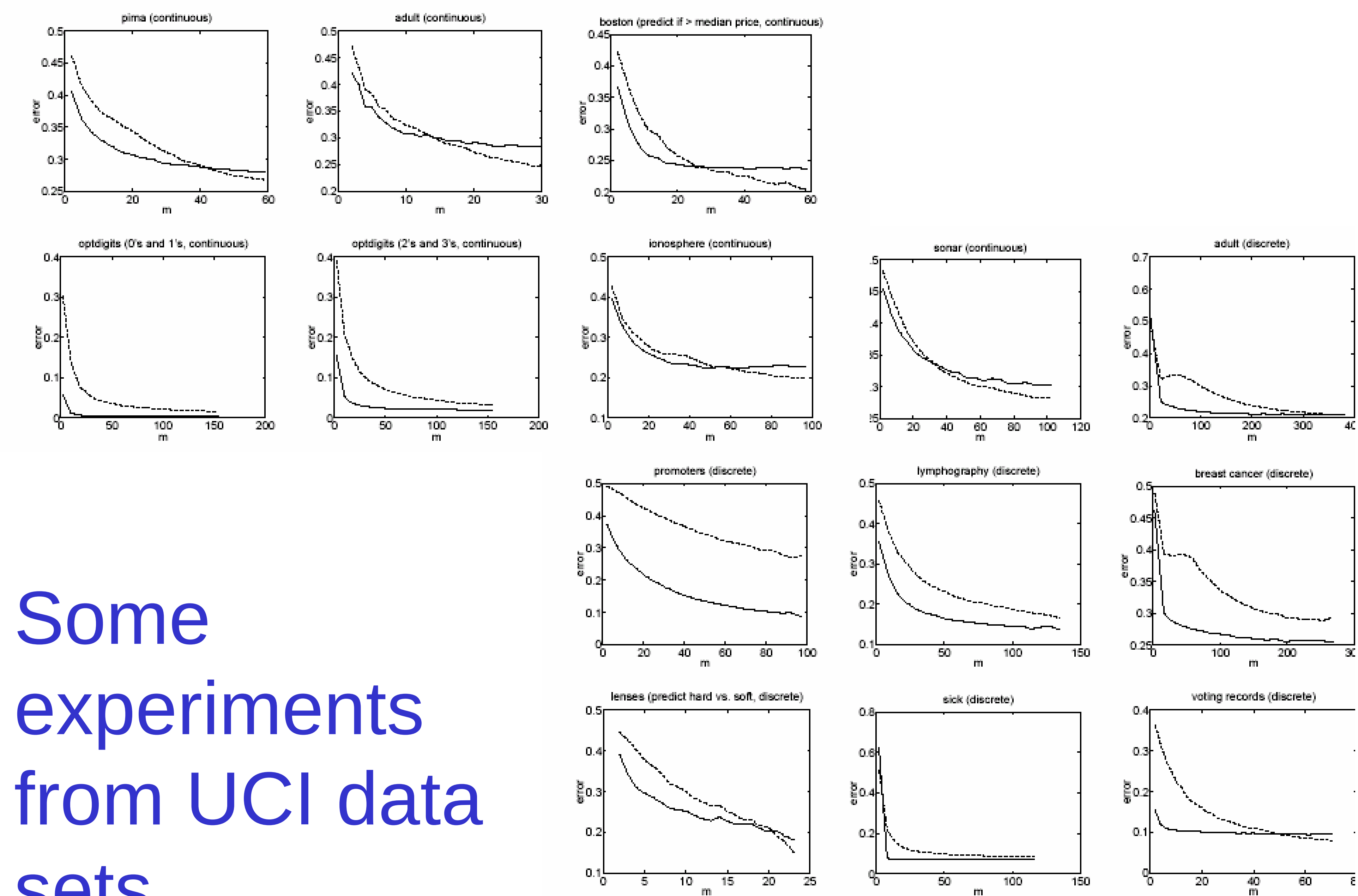


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learnin
repository. Plots are of generalization error vs. m (averaged over 1000 randor
train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

What you should know:

- Logistic regression
  - Functional form follows from Naïve Bayes assumptions
    - For Gaussian Naïve Bayes assuming variance $\sigma_{i,k} = \sigma_i$
    - For discrete-valued Naïve Bayes too
  - But training procedure picks parameters without the conditional independence assumption
  - MLE training: pick W to maximize P(Y | X, W)
  - MAP training: pick W to maximize P(W | X,Y)
    - 'regularization'
    - helps reduce overfitting

- Gradient ascent/descent
  - General approach when closed-form solutions unavailable

- Generative vs. Discriminative classifiers
  - Bias vs. variance tradeoff