PennState Institute for Computational and Data Sciences

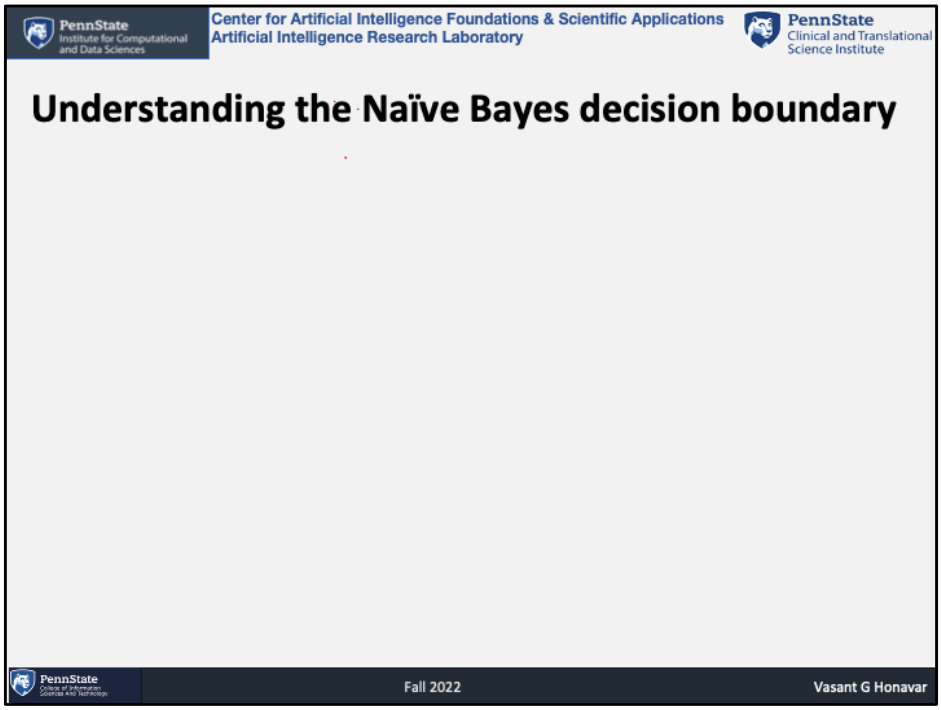
Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory

PennState Clinical and Translational Science Institute

DS 310 Probabilistic Models

Vasant G. Honavar
Dorothy Foehr Huck and J. Lloyd Huck Chair in Biomedical Data Sciences and Artificial Intelligence
Professor of Data Sciences, Informatics, Computer Science and Engineering, Bioinformatics & Genomics, Public Health Sciences and Neuroscience
Director, Center for Artificial Intelligence Foundations and Scientific Applications
Associate Director, Institute for Computational and Data Sciences
Pennsylvania State University

vhonavar@psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu


PennState
Institute for Computational and Data Sciences
Center for Artificial Intelligence Foundations & Scientific Applications
Artificial Intelligence Research Laboratory
PennState
Clinical and Translational Science Institute
Understanding the Naïve Bayes decision boundary
PennState
Fall 2022
Vasant G Honavar

NB decision boundary

$$label = \underset{j}{argmax}\ p(\omega_j) \prod_{i=1}^{n} p(x_i \mid \omega_j)$$

What does the decision boundary for NB look like if the features are binary?

Some math

$$label = \underset{j}{argmax}\; p(\omega_j) \prod_{i=1}^{n} p(x_i \,|\omega_j)$$

$$label = \underset{j}{argmax}\; \log \quad p(\omega_j) + \sum_{i=1}^{n} \log p(x_i|\omega_j)$$

Suppose x_i is binary (0/1)

$$label = \underset{j}{argmax}\; \log p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) + \overline{x_i} \log p(\overline{x_i}|\omega_j)$$

Some more math

$$label = \underset{j}{argmax} \log \quad p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) + \bar{x}_i \log p(\bar{x}_i|\omega_j)$$

$$label = \underset{j}{argmax} \log \quad p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) + (1 - x_i) \log p(\bar{x}_i|\omega_j)$$

(because x_i are binary)

$$label = \underset{\omega_j}{argmax} \log \quad p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) + (1 - x_i) \log(1 - p(x_i|\omega_j))$$

$$label = \underset{j}{argmax} \log p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) - x_i \log\left(1 - p(x_i|\omega_j)\right) + \log\left(1 - p(x_i|\omega_j)\right)$$

And…

$$label = \underset{j}{argmax} \log \quad p(\omega_j) + \sum_{i=1}^{n} x_i \log p(x_i|\omega_j) - x_i \log\left(1 - p(x_i|\omega_j)\right) + \left(\log 1 - p(x_i|\omega_j)\right)$$

$$label = \underset{j}{argmax} \log \quad p(\omega_j) + \sum_{i=1}^{n} \log(1 - p(x_i|\omega_j)) + \sum_{i=1}^{n} x_i \log\left(\frac{p(x_i|\omega_j)}{1 - p(x_i|\omega_j)}\right)$$

What does this look like?

$$label = \underset{j}{argmax} \boxed{\log \quad p(\omega_j) + \sum_{i=1}^{n} \log\left(1 - p(x_i|\omega_j)\right)} + \boxed{\sum_{i=1}^{n} x_i \log\left(\frac{p(x_i|\omega_j)}{1 - p(x_i|\omega_j)}\right)}$$

b_j $\quad\quad$ $\sum_{i=1}^{n} x_i w_{ji}$

$label = argmax_j(b_j + \sum_{i=1}^{n} x_i w_{ji}) = argmax_j \sum_{i=0}^{n} x_i w_{ji}$ where $x_0 = 1$ and $w_{j0} = b_j$

$label = argmax_j(b_j + \sum_{i=1}^{n} x_i w_{ji}) = argmax_j \sum_{i=0}^{n} x_i w_{ji}$ where $x_0 = 1$ and $w_{j0} = b_j$

And

$$label = argmax_j\left(b_j + \sum_{i=1}^{n} x_i\, w_{ji}\right)$$

$$= argmax_j \sum_{i=0}^{n} x_i\, w_{ji} \qquad \text{where } x_0 = 1 \text{ and } w_{j0} = b_j$$

$$= argmax_j \mathbf{w}_j \cdot \mathbf{x} \qquad \text{where } \mathbf{w}_j = \left[w_{j0} \cdots w_{jn}\right]^T$$

Thus, naïve bayes behaves like a linear winner take all classifier (albeit with parameters that have a probabilistic interpretation)!!!

NB as a linear model

$$\log\left(\frac{p(x_i|\,\omega_j)}{1-p(x_i|\omega_j)}\right)$$

How likely this binary feature is to be 1 given the label

How likely this binary feature is to be 0 given the label

- low weights associated with uninformative features
- high weights (positive or negative) associated with important features

Generative Versus Discriminative Models

- The preceding analysis shows that probabilistic generative models have close discriminative cousins
- Is there a systematic relationship between the two?

- Bayesian decision theory revisited
- Generative models
  - Model the probability distributions
- Discriminative models
  - Model decision boundaries
- Relating generative and discriminative models
- Tradeoffs between generative and discriminative models
- Generalizations and extensions

Alternative realizations of the Bayesian recipe

Chef 1: Generative model

Note that $P(\omega_i \mid \mathbf{x}) = \dfrac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})}$

Model $P(\mathbf{x} \mid \omega_1)$, $P(\mathbf{x}|\omega_2)$, $P(\omega_1)$, and $P(\omega_2)$

Using Bayes rule, choose ω_1 if $P(\mathbf{x} \mid \omega_1)P(\omega_1) > P(\mathbf{x}|\omega_2)P(\omega_2)$

Otherwise choose ω_2

Chef 2: Discriminative Model

Model $P(\omega_1 \mid \mathbf{x})$, $P(\omega_2 \mid \mathbf{x})$, or the ratio $\dfrac{P(\omega_1 \mid \mathbf{x})}{P(\omega_2 \mid \mathbf{x})}$ directly

Choose ω_1 if $\dfrac{P(\omega_1 \mid \mathbf{x})}{P(\omega_2 \mid \mathbf{x})} > 1$

Otherwise choose ω_2

Generative vs. Discriminative Classifiers

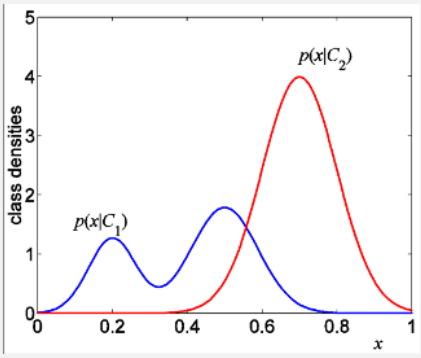
Generative classifiers

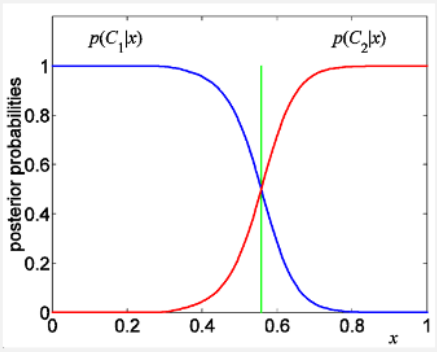
- Assume some functional form for $P(\mathbf{X}|\text{class})$, $P(\text{class})$
- Estimate parameters of $P(\mathbf{X}|\text{class})$, $P(\text{class})$ directly from training data
- Use Bayes rule to calculate $P(\text{class}|\mathbf{X}=\text{x})$

Discriminative classifiers

- Assume some functional form for $P(\text{class}|\mathbf{X})$
- Estimate parameters of $P(\text{class}|\mathbf{X})$ directly from training data

Generative vs. Conditional Models





Which Chef has a better Bayesian Optimal Classifier recipe?

- In theory, in the limit (∞ number of training samples) generative and conditional models produce identical results
- In the limit, the classification produced by the generative model is the same as that produced by the discriminative model
- Given unlimited data, assuming that both approaches select the correct form for the relevant probability distributions or the model for the discriminant function, they will produce identical results (Why?)
- If the assumed form of the probability distributions is incorrect, then it is possible that the generative model might have a higher classification error than the discriminative model (Why?)

How about in practice?

Which Chef has a better Bayesian Optimal Classifier recipe?

In practice

- The error of the classifier that uses the discriminative model can be lower than that of the classifier that uses the generative model (Why?)
  - Naïve Bayes is a generative model
  - A perceptron is a discriminative model, and so is SVM
  - An SVM can outperform Naïve Bayes on classification
- If the goal is classification, it might be useful to consider discriminative models that directly learn the classifier without attempting to solve the harder intermediate problem of modeling the joint probability distribution of features and classes (Vapnik – support vector machines)

From generative to discriminative models

- Assume classes are binary $y \in \{0,1\}$
- Suppose we model the class by a binomial distribution with parameter q $p(y \mid q) = q^{y}(1-q)^{(1-y)}$
- Assume each component x_j of input $\mathbf{X}$ have Gaussian distributions with parameters Θ_j (mean, variance) and are independent given the class

$$p(x, y \mid \Theta) = P(y \mid q)\prod_{j=1}^{n} p(x_j \mid y, \theta_j)$$

$$\text{where } \Theta = (q, \theta_1, \ldots . \theta_n)$$

From generative to discriminative models

$$p(x_j|y=0,\theta_j)=\frac{1}{(2\pi\sigma_j^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma_j^2}(x_j-\mu_{0j})^2\right\}$$

$$p(x_j|y=1,\theta_j)=\frac{1}{(2\pi\sigma_j^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma_j^2}(x_j-\mu_{1j})^2\right\}$$

where $\theta_j=(\mu_{0j},\mu_{1j},\sigma_j)$

(Note : we have assumed that $\forall j\ \sigma_{0j}=\sigma_{1j}=\sigma_j$)

Some reminders

- *V*, C vectors; A matrix; **X** a vector of variables.

$$\|V\|^2 = V^T V \quad \text{(norm)}$$

$$\|V\|_C^2 = (CV)^T (CV) = V^T C^T C V \quad \text{(weighted norm)}$$

$$\|V\|_C^2 = \|V\|^2 \quad \text{if } C^T C = \text{identity matrix}$$

$$\frac{d}{d\mathbf{X}}(A\mathbf{X}) = A$$

$$\frac{d}{d\mathbf{X}}(\mathbf{X}^T A\mathbf{X}) = 2A\mathbf{X} \quad \text{(when A is a symmetric matrix)}$$

$$\frac{d}{dA}(\mathbf{X}^T A\mathbf{X}) = \mathbf{X}^T\mathbf{X}$$

From generative to discriminative models

The calculation of the posterior probability $p(Y=1|x, \Theta)$ is simplified if we use matrix notation

$$p(x \mid y=1,\Theta) = \prod_{j=1}^{j=n} \left(\frac{1}{(2\pi)^{1/2}\sigma_j} \exp\left\{ -\frac{1}{2}\left(\frac{x_1 - \mu_{1j}}{\sigma_j} \right)^2 \right\} \right)$$

$$= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1) \right\}$$

where $\mu_1 = (\mu_{11}....\mu_{1n})^T$; and $\Sigma = diag(\sigma_1^2....\sigma_n^2) = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & . & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix}$

From generative to discriminative models

$$p(y=1 \mid x,\Theta) = \frac{p(x \mid y=1,\Theta)p(y=1 \mid q)}{p(x \mid y=1,\Theta)p(y=1 \mid q) + p(x \mid y=0,\Theta)p(y=0 \mid q)}$$

$$= \frac{q \exp\left\{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right\}}{q \exp\left\{-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1)\right\} + (1-q)\exp\left\{-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right\}}$$

$$= \frac{1}{1+\exp\left\{-\log\left(\frac{q}{1-q}\right) + \frac{1}{2}(x-\mu_1)^T \Sigma^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_0)^T \Sigma^{-1}(x-\mu_0)\right\}}$$

$$= \frac{1}{1+\exp\left\{-\underbrace{(\mu_1-\mu_0)^T \Sigma^{-1}}_{\beta^T} x + \underbrace{\frac{1}{2}(\mu_1-\mu_0)^T \Sigma^{-1}(\mu_1+\mu_0) - \log\left(\frac{q}{1-q}\right)}_{-\gamma}\right\}}$$

$$= \frac{1}{1+\exp\left(-\beta^T x - \gamma\right)}$$

where we have used $A^T DA - B^T DB = (A+B)^T D(A-B)$ for a symmetric matrix D

From generative to discriminative models

$$p(y=1 \mid x,\Theta) = \frac{1}{1+\exp(-\beta^T x-\gamma)}$$

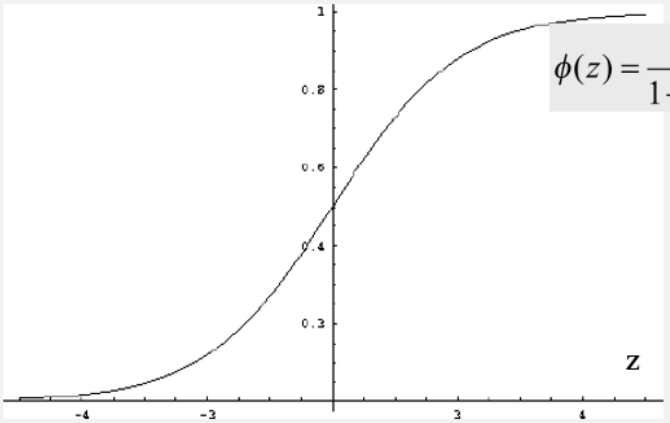
The posterior probability that Y=1 takes the form

where $\phi(z) = \frac{1}{1+e^{-z}}$

$z = \beta^T x + \gamma$ is an affine function of x

Sigmoid or Logistic Function

$$\phi(z)$$

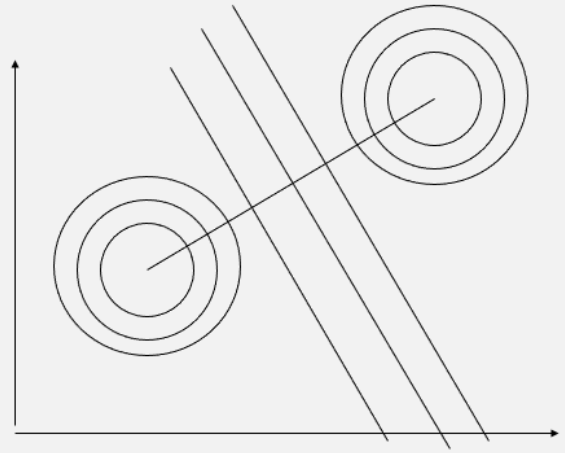


$$\phi(z) = \frac{1}{1+e^{-z}}$$

Implications of the logistic posterior

- Posterior probability of Y is a logistic function of an affine function of x (linear function of x + constant)
- Contours of equal posterior probability are lines in the input space
- $\beta^T x$ is proportional to the projection of x on β and this projection is equal for all vectors x that lie along a line that is orthogonal to β
- Special case
  - variances of Gaussians = 1
  - the contours of equal posterior probability are lines that are orthogonal to the difference vector between the means of the two classes
- Equal posterior for the two classes when $z=0$

Geometric interpretation (diagonal ∑)
Contour plot of the class conditional density functions



Geometric interpretation (diagonal ∑)

$$p(y=1\mid x,\Theta)=\frac{p(x\mid y=1,\Theta)p(y=1\mid q)}{p(x\mid y=1,\Theta)p(y=1\mid q)+p(x\mid y=0,\Theta)p(y=0\mid q)}$$

$$=\frac{1}{1+\exp\left\{\underbrace{-(\mu_1-\mu_0)^T\Sigma^{-1}}_{\beta^T}x+\underbrace{\frac{1}{2}(\mu_1-\mu_0)^T\Sigma^{-1}(\mu_1+\mu_0)-\log\left(\frac{q}{1-q}\right)}_{-\gamma}\right\}}$$

$$=\frac{1}{1+\exp(-\beta^T x-\gamma)}=\frac{1}{1+e^{-z}}$$

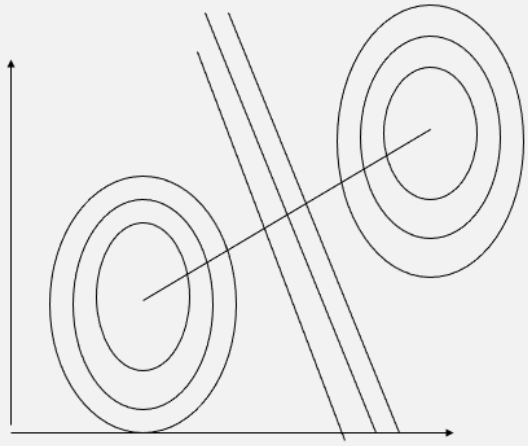
when $q=1\text{-}q$, $z=(\mu_1-\mu_0)^T\Sigma^{-1}\left(x-\frac{(\mu_1+\mu_0)}{2}\right)$

In this case, the posterior probabilities for the two classes are equal when x is equidistant from the two means

Geometric interpretation (diagonal ∑)

- If the prior probabilities of the classes are such that $q > 0.5$ the effect is to shift the logistic function to the left resulting in a larger value for the posterior probability for Y=1 for any given point in the input space.
- $q < 0.5$ results in a shift of the logistic function to the right resulting in a smaller value for the posterior probability for Y=1 (or larger value for the posterior probability for Y=0)

Geometric interpretation (general ∑)



Now the equi-probability contours are still lines in the input space although the lines are no longer orthogonal to the difference in means of the two classes

Generalization to multiple classes – Softmax function

- *Y* is a multinomial variable which takes on one of *K* values

$$q_k = p(y = k \mid q) = p(y^k = 1 \mid q)$$
$$\text{where } (y = k) \equiv (y^k = 1)$$
$$q = (q_1 \quad q_2 \quad \cdots \quad q_K)$$

- As before. $\mathbf{x}$ is a multivariate Gaussian

$$p(\mathbf{x} \mid y_k = 1, \Theta) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right\}$$

where $\boldsymbol{\mu}_k = (\mu_{k1} \ldots \mu_{kn})^T$; and $\forall k \ \ \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$

(covariance matrix is assumed to be same for each class)

Generalization to multiple classes – Softmax function

Posterior probability for class k is obtained via Bayes rule

$$p\left(y^k=1\mid \mathbf{x},\Theta\right)=\frac{p\left(\mathbf{x}\mid y^k=1,\Theta\right)p\left(y^k=1\mid q\right)}{\sum_{l=1}^{K}p\left(\mathbf{x}\mid y^k=1,\Theta\right)p\left(y^k=1\mid q\right)}$$

$$=\frac{q_k\exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_k\right)\right\}}{\sum_{l=1}^{K}q_l\exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_l\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_l\right)\right\}}$$

$$=\frac{\exp\left\{\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\mathbf{x}-\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k+\log q_k\right\}}{\sum_{l=1}^{K}\exp\left\{\boldsymbol{\mu}_l^T\boldsymbol{\Sigma}^{-1}\mathbf{x}-\frac{1}{2}\boldsymbol{\mu}_l^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l+\log q_l\right\}}$$

Generalization to multiple classes – Softmax function

We have shown that

$$p\left(y^k = 1 \mid \mathbf{x}, \Theta\right) = \frac{\exp\left\{\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log q_k\right\}}{\sum_{l=1}^{K} \exp\left\{\boldsymbol{\mu}_l^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_l^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l + \log q_l\right\}}$$

Defining parameter vectors and augmenting the input Vector $\mathbf{x}$ by adding a constant input of 1 we have

$$\boldsymbol{\beta}_k = \begin{bmatrix} -\frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log q_k \\ \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \end{bmatrix}$$

$$p\left(y^k = 1 \mid \mathbf{x}, \Theta\right) = \frac{e^{\boldsymbol{\beta}_k^T \mathbf{x}}}{\sum_{l=1}^{K} e^{\boldsymbol{\beta}_l^T \mathbf{x}}} = \frac{e^{\langle \boldsymbol{\beta}_k, \mathbf{x} \rangle}}{\sum_{l=1}^{K} e^{\langle \boldsymbol{\beta}_l, \mathbf{x} \rangle}}$$

Generalization to multiple classes – Softmax function

$$p\left(y^k = 1 \mid \mathbf{x}, \Theta\right) = \frac{e^{\boldsymbol{\beta}_k^T \mathbf{x}}}{\sum_{l=1}^{K} e^{\boldsymbol{\beta}_l^T \mathbf{x}}} = \frac{e^{\langle \boldsymbol{\beta}_k, \mathbf{x} \rangle}}{\sum_{l=1}^{K} e^{\langle \boldsymbol{\beta}_l, \mathbf{x} \rangle}}$$

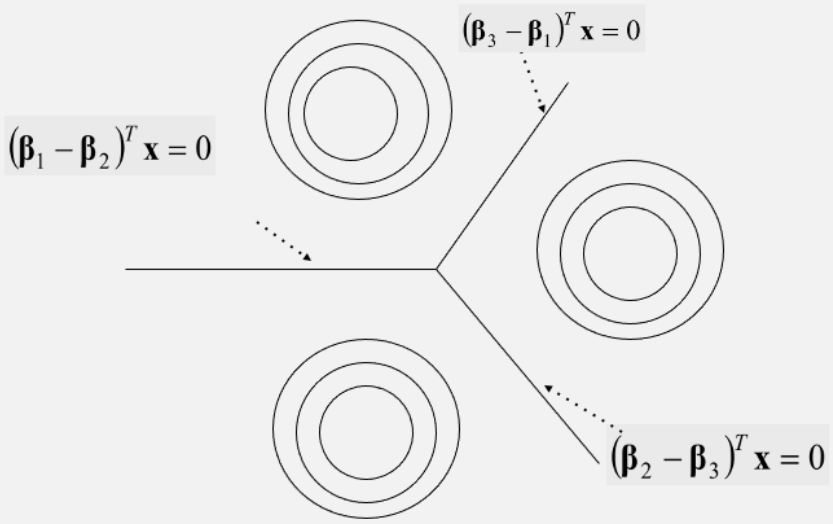
corresponds to the decision rule:

$$h(\mathbf{x}) = \arg\max_j \; p\left(y^k = 1 \mid \mathbf{x}, \Theta\right) = \arg\max_j \; e^{\langle \boldsymbol{\beta}_j, \mathbf{x} \rangle} = \arg\max_j \; \langle \boldsymbol{\beta}_j, \mathbf{x} \rangle$$

Consider the ratio of posterior prob. for classes k and $j \neq k$

$$\frac{p\left(y^k = 1 \mid \mathbf{x}, \Theta\right)}{p\left(y^j = 1 \mid \mathbf{x}, \Theta\right)} = \frac{e^{\langle \boldsymbol{\beta}_k, \mathbf{x} \rangle}}{\sum_{l=1}^{K} e^{\langle \boldsymbol{\beta}_l, \mathbf{x} \rangle}} \frac{\sum_{l=1}^{K} e^{\langle \boldsymbol{\beta}_l, \mathbf{x} \rangle}}{e^{\langle \boldsymbol{\beta}_j, \mathbf{x} \rangle}} = \frac{e^{\langle \boldsymbol{\beta}_k, \mathbf{x} \rangle}}{e^{\langle \boldsymbol{\beta}_j, \mathbf{x} \rangle}} = e^{\langle (\boldsymbol{\beta}_k - \boldsymbol{\beta}_j), \mathbf{x} \rangle}$$

Equi-probability contours of the softmax function



Naïve Bayes classifier with discrete attributes and K classes

q_k= prior probability of class k

η_{kji}= probability that x_j (the jth component of $\mathbf{x}$) takes the ith value in its domain when $\mathbf{x}$ belongs to class k.

$$p(\mathbf{x},\mathbf{y}\mid\Theta)=p(\mathbf{y}\mid q)\prod_{j=1}^{n}p(x_j\mid\mathbf{y},\theta_j)$$

$$q_k=p(y^k=1\mid q)$$

$$\eta_{kji}=p(x_j^i=1\mid y^k=1,\eta)$$

$$p(y^k=1\mid\mathbf{x},\eta)=\frac{q_k\prod_j\prod_i(\eta_{kji})^{x_j^i}}{\sum_l q_l\prod_j\prod_i(\eta_{kji})^{x_j^i}}$$

Naïve Bayes classifier with discrete attributes and K classes

$$q_k = p\left(y^k = 1 \mid q\right), \quad \eta_{kji} = p\left(x_j^i = 1 \mid y^k = 1, \eta\right)$$

$$p\left(y^k = 1 \mid \mathbf{x}, \eta\right) = \frac{q_k \prod_j \prod_i \left(\eta_{kji}\right)^{x_j^i}}{\sum_l q_l \prod_j \prod_i \left(\eta_{kji}\right)^{x_j^i}}$$

$$= \frac{\exp\left\{\log q_k + \sum_{j=1}^{n} \sum_{i=1}^{N_j} x_j^i \log \eta_{kji}\right\}}{\sum_{l=1}^{K} \exp\left\{\log q_l + \sum_{j=1}^{n} \sum_{i=1}^{N_j} x_j^i \log \eta_{lji}\right\}}$$

$$= \frac{e^{\boldsymbol{\beta}_k^T \mathbf{x}}}{\sum_{l=1}^{K} e^{\boldsymbol{\beta}_l^T \mathbf{x}}} = \frac{e^{\langle \boldsymbol{\beta}_k, \mathbf{x} \rangle}}{\sum_{l=1}^{K} e^{\langle \boldsymbol{\beta}_l, \mathbf{x} \rangle}}$$

From generative to discriminative models

- A curious fact about all of the generative models we have considered so far is that
  - The posterior probability of class can be expressed in the form of
    - a logistic function in the case of a binary classifier and
    - a softmax function in the case of a K-class classifier

From generative to discriminative models

- For multinomial and Gaussian class conditional densities (in the case of the latter, with equal but otherwise arbitrary covariance matrices)
  - The contours of equal posterior probabilities of classes are hyperplanes in the input (feature) space.
- The result is a simple linear classifier analogous to the perceptron (for binary classification) or winner-take-all network (for K-ary classification)
- Next, we see that these results hold for a more general class of distributions

The exponential family of distributions

The exponential family is specified by

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\}}$$

where $\boldsymbol{\eta}$ is a parameter vector and $A(\boldsymbol{\eta})$, $h(\mathbf{x})$ and $G(\mathbf{x})$ are appropriately chosen functions.

- Gaussian, Binomial, and multinomial (and many other) distributions belong to the exponential family

The Bernoulli distribution belongs to the exponential family

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\left\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\right\}}$$

Bernoulli distribution with success rate q is given by

$$p(x \mid q) = q^x (1-q)^{1-x} = \exp\left\{\log\left(\frac{q}{1-q}\right)x + \log(1-q)\right\}$$

We can see that Bernoulli distribution belongs to the exponential family by choosing

$$\eta = \log\left(\frac{q}{1-q}\right);\ G(x) = x;\ h(x) = 1$$
$$A(\eta) = -\log(1-q) = \log(1+e^{\eta})$$

The Gaussian distribution belongs to the exponential family

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\}}$$

Univariate Gaussian distribution can be written as

$$p(x \mid \mu, \sigma^2) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$= \frac{1}{(2\pi)^{1/2}} \exp\left\{\frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{1}{2\sigma^2}\mu^2 - \ln\sigma\right\}$$

We see that Gaussian distribution belongs to the exponential family by choosing

$$\eta = \begin{bmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{bmatrix}; \quad A(\eta) = \frac{\mu^2}{2\sigma^2} + \ln\sigma$$

$$G(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}; \quad h(x) = \frac{1}{(2\pi)^{1/2}}$$

The multinomial distribution belongs to the exponential family

The exponential family which is given by

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\left\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\right\}}$$

where $\boldsymbol{\eta}$ is a parameter vector and $A(\boldsymbol{\eta})$, $h(\mathbf{x})$ and $G(\mathbf{x})$ are appropriately chosen functions – can be shown to include several additional distributions such as the multinomial, the Poisson, the Gamma, the Dirichlet, among others.

Exercise: Show that the multinomial distribution belongs to the exponential family.

From generative to discriminative models

- In the case of the generative models we have seen
- The posterior probability of class can be expressed in the form of
  - a logistic function in the case of a binary classifier and
  - a softmax function in the case of a K-class classifier
- The contours of equal posterior probabilities of classes are
  - hyperplanes in the input (feature) space yielding
    - a linear classifier for binary classification) or
    - a winner-take-all network (for K-ary classification).

From generative to discriminative models

- We just showed that the probability distributions underlying the generative models considered belong to the exponential family
- What can we say about the classifiers when the underlying generative models are distributions from the exponential family?

Discriminative classifier when each class is modeled by a distribution from the exponential family

Consider Binary classification task with density for class 0 and class 1 parameterized by η_0 and η_1. Further assume $G(\mathbf{x})$ is a linear function of $\mathbf{x}$ (before augmenting $\mathbf{x}$ with a 1)

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\}}$$

$$p(y=1 \mid \mathbf{x}, \boldsymbol{\eta}) = \frac{p(\mathbf{x} \mid y=1, \boldsymbol{\eta}) p(y=1 \mid q)}{p(\mathbf{x} \mid y=1, \boldsymbol{\eta}) p(y=1 \mid q) + p(\mathbf{x} \mid y=0, \boldsymbol{\eta}) p(y=0 \mid q)}$$

$$= \frac{\exp\left\{\boldsymbol{\eta}_1^T G(\mathbf{x}) - A(\boldsymbol{\eta}_1)\right\} h(\mathbf{x}) q_1}{\exp\left\{\boldsymbol{\eta}_1^T G(\mathbf{x}) - A(\boldsymbol{\eta}_1)\right\} h(\mathbf{x}) q_1 + \exp\left\{\boldsymbol{\eta}_0^T G(\mathbf{x}) - A(\boldsymbol{\eta}_0)\right\} h(\mathbf{x}) q_0}$$

$$p(y=1 \mid \mathbf{x}, \boldsymbol{\eta}) = \frac{1}{1 + \exp\left\{-(\boldsymbol{\eta}_0 - \boldsymbol{\eta}_1)^T G(\mathbf{x}) - A(\boldsymbol{\eta}_0) + A(\boldsymbol{\eta}_1) + \log \frac{q_0}{q_1}\right\}}$$

Note that this is a logistic function of a linear function of $\mathbf{x}$

Discriminative classifier when each class is modeled by a distribution from the exponential family

Consider K-ary classification task; Suppose $G(\mathbf{x})$ is a linear function of $\mathbf{x}$

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) e^{\{\boldsymbol{\eta}^T G(\mathbf{x}) - A(\boldsymbol{\eta})\}}$$

$$p(y^k = 1 \mid \mathbf{x}, \boldsymbol{\eta}) = \frac{\exp\{\boldsymbol{\eta}_k{}^T G(\mathbf{x}) - A(\boldsymbol{\eta}_k)\} q_k}{\sum_{l=1}^{K} \exp\{\boldsymbol{\eta}_l{}^T G(\mathbf{x}) - A(\boldsymbol{\eta}_l)\} q_l}$$

$$= \frac{\exp\{\boldsymbol{\eta}_k{}^T G(\mathbf{x}) - A(\boldsymbol{\eta}_k) + \log q_k\}}{\sum_{l=1}^{K} \exp\{\boldsymbol{\eta}_l{}^T G(\mathbf{x}) - A(\boldsymbol{\eta}_l) + \log q_l\}}$$

which is a softmax function of a linear function of $\mathbf{x}$!!

Summary

- A variety of class conditional densities all yield the same logistic-linear or softmax-linear (with respect to parameters) form for the posterior probability
- In practice, choosing a class conditional density can be difficult – especially in high dimensional spaces – e.g., multi-variate Gaussian where the covariance matrix grows quadratically in the number of dimensions!
- The invariance of the functional form of the posterior probability with respect to the choice of the distribution is good news!
- It is not necessary to specify the class conditional density if we can work directly with the posterior – which brings us to discriminative models!

Discriminative Models

- We saw that under fairly general assumptions concerning the underlying generative model, the posterior probability of class given **x** can be expressed in the form of a logistic function of an affine or polynomial (in the simplest case, linear) function of **x** in the case of a binary classification task.

$$P(y=1 \mid \mathbf{x}) = \frac{1}{1+e^{-\langle \mathbf{w}, G(\mathbf{x}) \rangle}} = \frac{1}{1+e^{-\eta(\mathbf{x})}} = \mu(\mathbf{x})$$

where $\eta(\mathbf{x}) = \mathbf{w}^T G(\mathbf{x}) = \langle \mathbf{w}, G(\mathbf{x}) \rangle$

In the discriminative setting, we simply assume this form and proceed without regard to details of the underlying generative model

Reminder – Expectation and conditional expectation

$$E(Y) = \sum_y yP(y)$$

$$E(Y \mid x) = \sum_y yP(y|x)$$

Discriminative Models

When Y is a binary random variable, that the posterior probability of $Y=1$ is same as the conditional expectation of y given $\mathbf{x}$:

$$E(y \mid \mathbf{x}) = 1 \cdot P(y=1 \mid \mathbf{x}) + 0 \cdot P(y=0 \mid \mathbf{x})$$
$$= P(y=1 \mid \mathbf{x}) = \mu(\mathbf{x}) = (\mu(\mathbf{x}))^{y} (1-\mu(\mathbf{x}))^{1-y}$$

where $$\mu(\mathbf{x}) = \frac{1}{1+e^{-\eta(\mathbf{x})}} = \frac{1}{1+e^{-\mathbf{w}^T G(\mathbf{x})}}$$

Hence estimating $\mathrm{P}(Y=1|\mathbf{x})$ is equivalent to performing logistic regression

Some Properties of the Logistic Function

$$\mu = \frac{1}{1+e^{-\eta}}; \quad \eta = \log\left(\frac{\mu}{1-\mu}\right)$$

$$\frac{d\eta}{d\mu} = \frac{d}{d\mu}\log\left(\frac{\mu}{1-\mu}\right) = \left(\frac{1-\mu}{\mu}\right)\frac{d}{d\mu}\left(\frac{\mu}{1-\mu}\right) = \left(\frac{1-\mu}{\mu}\right)\left(\frac{(1-\mu)\frac{d}{d\mu}(\mu) - \mu\frac{d}{d\mu}(1-\mu)}{(1-\mu)^2}\right) = \frac{1}{\mu(1-\mu)}$$

$$\frac{d\mu}{d\eta} = \mu(1-\mu)$$

Maximum likelihood estimation of $\mathbf{w}$

$$D = \{(x_n, y_n); X_n \in Domain(\mathbf{x}); \quad y_n \in \{0,1\}; \; n = 1..N\}$$

$$\eta_n = \mathbf{w}^T x_n; \quad \mu_n = \frac{1}{1+e^{-\eta_n}}$$

Likelihood

$$P(y_1 \ldots y_N \mid x_1 \ldots x_N, \mathbf{w}) = \prod_{n=1}^{N} (\mu_n)^{y_n} (1-\mu_n)^{(1-y_n)}$$

Log likelihood

$$LL(\mathbf{w} : D) = \sum_{n=1}^{N} \{y_n \log \mu_n + (1-y_n) \log(1-\mu_n)\}$$

We need to find $\mathbf{w}$ that maximizes log likelihood

Maximum likelihood estimation of $\mathbf{w}$

$$LL(\mathbf{w}:D) = \sum_{n=1}^{N} \{y_n \log \mu_n + (1-y_n)\log(1-\mu_n)\}$$

$$\frac{\partial LL(\mathbf{w}:D)}{\partial \mathbf{w}} = \sum_{n=1}^{N} \left(\frac{y_n}{\mu_n} - \frac{(1-y_n)}{(1-\mu_n)}\right)\left(\frac{\partial \mu_n}{\partial \eta_n}\right)\left(\frac{\partial \eta_n}{\partial \mathbf{w}}\right)$$

$$= \sum_{n=1}^{N} \left(\frac{y_n - \mu_n}{\mu_n(1-\mu_n)}\right)\mu_n(1-\mu_n)x_n$$

$$= \sum_{n=1}^{N} (y_n - \mu_n)x_n$$

$$G(\mathbf{x}) = \mathbf{x}$$
$$\eta(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$$

Simple gradient ascent learning algorithm

$$\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) + \rho \left.\frac{\partial LL(\mathbf{w}:D)}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}(t)}$$

$$\rho > 0$$

Maximum likelihood estimation of **w**

The momentum trick can also be applied speed up the algorithm

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta\mathbf{w}(t)$$

$$\Delta\mathbf{w}(t) = \rho_t (y_n - \mu_n) x_n \big|_{\mathbf{w}=\mathbf{w}(t)} + \alpha \Delta w_i(t-1) \text{ where } 0 < \alpha < 1$$

$$= \sum_{\tau=0}^{t} \alpha^{t-\tau} \rho_t (y_n - \mu_n) x_n \big|_{w_i = w_i(\tau)}$$

Maximum likelihood estimation of w

- More sophisticated optimization algorithms can be used to maximize the log likelihood function which although not quadratic, is approximately quadratic.
- For details, see standard texts on optimization.
- When the form of the underlying generative model is known, we can initialize the parameter vector **w** based on the maximum likelihood estimates for which often closed form solutions are available and then run a few iterations of gradient ascent to improve classification accuracy.

Multi-class Discriminative Model

Softmax-linear model is the multi-class generalization of the logistic-linear model

$$p\left(y^k = 1 \mid \mathbf{x}\right) = \frac{e^{\boldsymbol{\theta}_k^T \mathbf{x}}}{\sum_{l=1}^{K} e^{\boldsymbol{\theta}_l^T \mathbf{x}}}$$

In the discriminative setting, we simply assume this form and proceed without regard to details of the underlying generative model

Multi-class Discriminative Model

Let $$p\left(y^k = 1 \mid x_n\right) = \frac{e^{\boldsymbol{\theta}_k^T x_n}}{\sum_{l=1}^{K} e^{\boldsymbol{\theta}_l^T x_n}} = \mu_n^k$$

Let
$$\eta_n^k = \boldsymbol{\theta}_k^T x_n$$
$$\eta_n = \left[\eta_n^1 \ldots \eta_n^k \ldots \eta_n^K\right]$$
$$\mu_n = \left[\mu_n^1 \ldots \mu_n^k \ldots \mu_n^K\right]$$

Then $$\mu_n^k = \frac{e^{\eta_n^k}}{\sum_{l=1}^{K} e^{\eta_n^l}} \; ; \quad \mu^k = \frac{e^{\eta^k}}{\sum_{l=1}^{K} e^{\eta^l}}$$

Some properties of the softmax function

Softmax-linear function is invertible up to an additive constant.

$$\mu^k = \frac{e^{\eta^k}}{\sum_{l=1}^{K} e^{\eta^l}}$$

$$\eta^k = \log \mu^k + C$$

$$C = \log\left(\sum_{l=1}^{K} e^{\eta^l}\right)$$

Some properties of the softmax function

$$\mu^k = \frac{e^{\eta^k}}{\sum_{l=1}^{K} e^{\eta^l}}; \qquad \eta^k = \log \mu^k + C; \qquad C = \log\left(\sum_{l=1}^{K} e^{\eta^l}\right)$$

$$\frac{\partial \mu^k}{\partial \eta^j} = \frac{\left(\sum_{l=1}^{K} e^{\eta^l}\right) e^{\eta^k} \delta_{kj} - e^{\eta^k} e^{\eta^j}}{\sum_{l=1}^{K} e^{\eta^l}}$$

$$= \frac{e^{\eta^k}}{\sum_{l=1}^{K} e^{\eta^l}} \left(\delta_{kj} - \frac{e^{\eta^j}}{\sum_{l=1}^{K} e^{\eta^l}}\right) = \mu^k \left(\delta_{kj} - \mu^j\right)$$

Maximum likelihood estimation of $\boldsymbol{\theta}$

$$D = \{(x_n, y_n); x_n \in Domain(\mathbf{x}); \quad y_n \in \{y_n^1 ... y_n^K\}; \; n = 1..N;\}$$

$$P(y_n \mid x_n, \boldsymbol{\theta}) = \prod_{k=1}^{K} \left(\mu_n^k\right)^{y_n^k}$$

$$L(\boldsymbol{\theta} : D) = P(y_1 ... y_N \mid x_1 ... x_N, \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\mu_n^k\right)^{y_n^k}$$

$$LL(\boldsymbol{\theta}_1 ... \boldsymbol{\theta}_K : D) = \sum_{n=1}^{N} \sum_{k=1}^{K} y_n^k \log \mu_n^k$$

We need to find parameters that maximize log likelihood

Maximum likelihood estimation of $\boldsymbol{\theta}$

$$\eta_n^k = \boldsymbol{\theta}_k^T x_n$$

$$LL(\boldsymbol{\theta} : D) = \sum_{n=1}^{N}\sum_{k=1}^{K} y_n^k \log \mu_n^k$$

$$\nabla_{\boldsymbol{\theta}_i} LL(\boldsymbol{\theta} : D) = \sum_{n=1}^{N}\sum_{k=1}^{K}\left(\frac{\partial LL(\boldsymbol{\theta} : D)}{\partial \mu_n^k}\right)\left(\frac{\partial \mu_n^k}{\partial \eta_n^i}\right)\left(\frac{\partial \eta_n^i}{\partial \boldsymbol{\theta}_i}\right)$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K} y_n^k \left(\delta_{ik} - \mu_n^i\right) x_n$$

$$= \sum_{n=1}^{N} \left(y_n^i - \mu_n^i\right) x_n$$

Where we have used the chain rule and the fact that $\forall n \quad \left(\sum_{k=1}^{K} y_n^k = 1\right)$

Maximum likelihood estimation of $\boldsymbol{\theta}$

$$\nabla_{\boldsymbol{\theta}_i} LL(\boldsymbol{\theta} : D) = \sum_{n=1}^{N} \left(y_n^i - \mu_n^i \right) x_n$$

Basic gradient ascent update rule is given by

$$\boldsymbol{\theta}_i(t+1) \leftarrow \boldsymbol{\theta}_i(t) + \rho \left. \frac{\partial LL(\boldsymbol{\theta} : D)}{\partial \boldsymbol{\theta}_i} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t)}$$

$$\rho > 0$$

which we can be speed up using the momentum trick as before

Maximum likelihood estimation of $\boldsymbol{\theta}$

The momentum trick provides a simple approach to speeding up the simple gradient ascent algorithm

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \Delta\boldsymbol{\theta}_i(t)$$

$$\Delta\boldsymbol{\theta}_i(t) = \rho \left.\frac{\partial LL(\boldsymbol{\theta}:D)}{\partial \boldsymbol{\theta}_i}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}(t)} + \alpha\Delta\boldsymbol{\theta}_i(t-1) \text{ where } 0<\alpha<1$$

$$= \sum_{\tau=0}^{t} \alpha^{t-\tau} \left.\frac{\partial LL(\boldsymbol{\theta}:D)}{\partial \boldsymbol{\theta}_i}\right|_{\boldsymbol{\theta}_i=\boldsymbol{\theta}_i(\tau)}$$

Maximum likelihood estimation of $\boldsymbol{\theta}$

$$\nabla_{\boldsymbol{\theta}_i} LL(\boldsymbol{\theta} : D) = \sum_{n=1}^{N} \left(y_n^i - \mu_n^i\right) x_n$$

Basic online gradient ascent update rule is given by

$$\boldsymbol{\theta}_i(t+1) \leftarrow \boldsymbol{\theta}_i(t) + \rho_t \left(y_n^i - \mu_n^i\right) x_n$$

which we can speed up using the momentum trick as before

Maximum likelihood estimation of $\mathbf{w}$

The momentum trick can also be applied in the on line version

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) + \Delta\boldsymbol{\theta}_i(t)$$

$$\Delta\boldsymbol{\theta}_i(t) = \rho_t\left(y_n^i - \mu_n^i\right)x_n\Big|_{\boldsymbol{\theta}_i=\boldsymbol{\theta}_i(t)} + \alpha\Delta\boldsymbol{\theta}_i(t-1) \text{ where } 0<\alpha<1$$

$$= \sum_{\tau=0}^{t}\alpha^{t-\tau}\rho_t\left(y_n - \mu_n\right)x_n\Big|_{\boldsymbol{\theta}_i=\boldsymbol{\theta}_i(\tau)}$$

Summary

- For a large class of generative models, the probability distribution of class conditioned on the input can be modeled by the exponential family
- Generative models can perform poorly when the assumed parametric form for the distribution is incorrect
- Discriminative models can perform poorly when the assumed form of $G(\mathbf{x})$ is inappropriate – but it is often easier to choose the form of $G(\mathbf{x})$ than it is to specify the precise form of the generative model
- Discriminative models focus on the classification problem without solving (potentially more difficult) problem of learning the generative model for data
- Estimating the parameters in the discriminative setting requires solving an optimization problem although their generative counterparts have closed form solutions (via sufficient statistics)

Summary

- We can learn classifiers in a discriminative setting using maximum likelihood or maximum a posteriori or Bayesian estimation of parameters
- Discriminative models may overfit the data – use of priors or regularization recommended
- Initializing the discriminative model parameters with estimates based on generative model helps