CS 688 – Pattern Recognition
Lecture 4

Linear Models for Classification

- Probabilistic generative models
- Probabilistic discriminative models

Probabilistic Generative Models

- We now turn to a probabilistic approach to classification
- We'll see how models with linear decision boundaries arise from simple assumptions about the distribution of the data

Generative Approach

- Solve the inference problem of estimating the class-conditional densities $p(\boldsymbol{x} \mid C_k)$ for each class C_k
- Infer the prior class probabilities $p(C_k)$
- Use Bayes' theorem to find the class posterior probabilities:

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(\boldsymbol{x})}$$

where $$p(\boldsymbol{x}) = \sum_k p(\boldsymbol{x} \mid C_k)p(C_k)$$

- Use decision theory to determine class membership for each new input $\boldsymbol{x}$

Probabilistic Generative Models

Two class case:

$$p(C_1 \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_1)p(C_1)}{p(\boldsymbol{x} \mid C_1)p(C_1) + p(\boldsymbol{x} \mid C_2)p(C_2)}$$

$$= \frac{1}{1+e^{-a}} = \sigma(a)$$

$$a = \ln \frac{p(\boldsymbol{x} \mid C_1)p(C_1)}{p(\boldsymbol{x} \mid C_2)p(C_2)}$$

$$\sigma(a) = \frac{1}{1+e^{-a}}$$ **logistic sigmoid function**

Probabilistic Generative Models

K >2 classes:

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{\sum_j p(\boldsymbol{x} \mid C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}$$

$$a_k = \ln p(\boldsymbol{x} \mid C_k)p(C_k)$$

$$\sigma(a) = \frac{e^{a_k}}{\sum_j e^{a_j}}$$ **softmax function**

Probabilistic Generative Models

Lets assume the class-conditional densities are Gaussian with the same covariance matrix:

$$p(\boldsymbol{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)}$$

Two class case first. We can show the following result:

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0)$$

$$\boldsymbol{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T \Sigma \boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T \Sigma \boldsymbol{\mu}_2 + \ln \frac{p(C_1)}{p(C_2)}$$

Probabilistic Generative Models

$$p(C_1 \mid \boldsymbol{x}) = \frac{1}{1+e^{-a}} = \sigma(a)$$

$$a = \ln \frac{p(\boldsymbol{x} \mid C_1)p(C_1)}{p(\boldsymbol{x} \mid C_2)p(C_2)} = \ln \frac{p(\boldsymbol{x} \mid C_1)}{p(\boldsymbol{x} \mid C_2)} + \ln \frac{p(C_1)}{p(C_2)}$$

$$= \ln p(\boldsymbol{x} \mid C_1) - \ln p(\boldsymbol{x} \mid C_2) + \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln \frac{p(C_1)}{p(C_2)}$$

$$[p(\boldsymbol{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)}]$$

Probabilistic Generative Models

$$a =$$

$$= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln \frac{p(C_1)}{p(C_2)}$$

$$= -\frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1$$

$$+ \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}_2 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(C_1)}{p(C_2)}$$

$$= \boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{x} - \boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{x} \underbrace{- \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(C_1)}{p(C_2)}}_{w_0}$$

$$= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\Sigma^{-1}\boldsymbol{x} + w_0 = \boldsymbol{w}^T\boldsymbol{x} + w_0 \quad \boldsymbol{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Probabilistic Generative Models

We have shown:

$$p(C_1 \mid \boldsymbol{x}) = \sigma\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right)$$

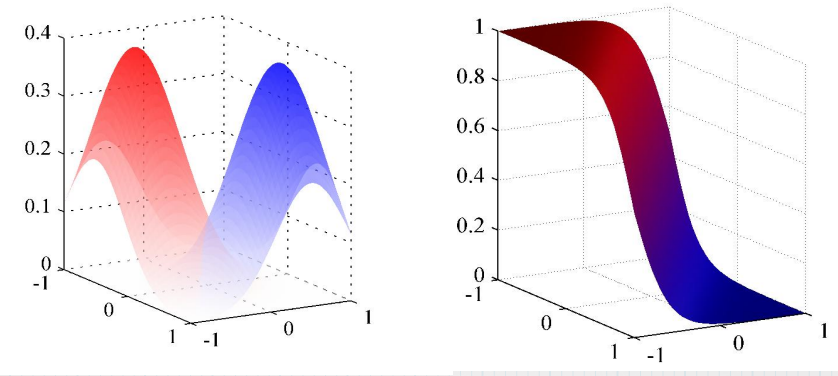
$$w = \Sigma^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T \Sigma \boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T \Sigma \boldsymbol{\mu}_2 + \ln\frac{p(C_1)}{p(C_2)}$$

Decision boundary: $p(C_1 \mid \boldsymbol{x}) = p(C_2 \mid \boldsymbol{x}) = 0.5$

$$\Rightarrow \frac{1}{1+e^{-\left(\boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{w}_0\right)}} = 0.5 \Rightarrow \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{w}_0 = 0$$

Probabilistic Generative Models



Probabilistic Generative Models

K **>2 classes**:

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{\sum_j p(\boldsymbol{x} \mid C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}$$

$$a_k = \ln p(\boldsymbol{x} \mid C_k)p(C_k)$$

We can show the following result:

$$p(C_k \mid x) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T x + w_{j0}}}$$

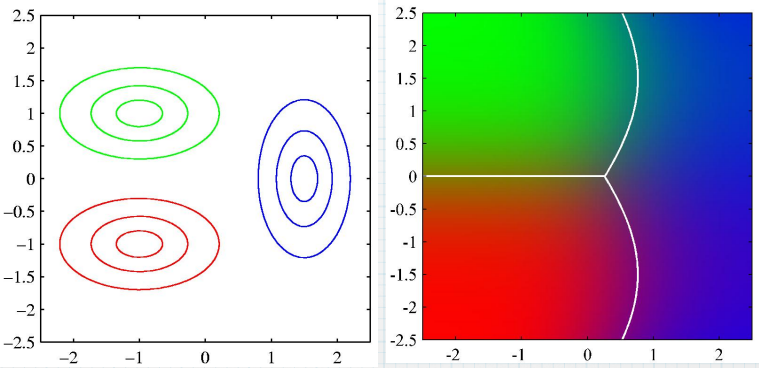
$$\boldsymbol{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k \quad w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T \Sigma^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$$

Probabilistic Generative Models

The cancellation of the quadratic terms is due to the assumption of shared covariances.

If we allow each class conditional density to have its own covariance matrix, the cancellations no longer occur, and we obtain a quadratic function of $\boldsymbol{x}$, i.e. a **quadratic discriminant**.

Probabilistic Generative Models



Maximum likelihood solution

We have a parametric functional form for the class-conditional densities:

$$p(\boldsymbol{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)}$$

We can estimate the parameters and the prior class probabilities using maximum likelihood.

- Two class case with shared covariance matrix.
- Training data:

$$\{\boldsymbol{x}_n, t_n\} \; n = 1, \cdots, N$$

$$t_n = 1 \; denotes \; class \; C_1 \qquad t_n = 0 \; denotes \; class \; C_2$$

$$\textbf{\textit{Priors :}} \; p(C_1) = \pi, \quad p(C_2) = (1-\pi)$$

Maximum likelihood solution

$\{\boldsymbol{x}_n, t_n\}\ n = 1, \cdots, N$

$t_n = 1$ *denotes class* C_1 $\quad t_n = 0$ *denotes class* C_2

Priors : $p(C_1) = \pi, \quad p(C_2) = (1-\pi)$

For a data point $\boldsymbol{x}_n$ from class C_1 we have $t_n = 1$ and therefore

$$p(\boldsymbol{x}_n, C_1) = p(C_1)p(\boldsymbol{x}_n \mid C_1) = \pi N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma)$$

For a data point $\boldsymbol{x}_n$ from class C_2 we have $t_n = 0$ and therefore

$$p(\boldsymbol{x}_n, C_2) = p(C_2)p(\boldsymbol{x}_n \mid C_2) = (1-\pi)N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_2, \Sigma)$$

Maximum likelihood solution

$\{\boldsymbol{x}_n, t_n\}\ n = 1, \cdots, N$

$t_n = 1$ *denotes class* C_1 $\quad t_n = 0$ *denotes class* C_2

Priors : $p(C_1) = \pi, \quad p(C_2) = (1-\pi)$

$p(\boldsymbol{x}_n, C_1) = p(C_1)p(\boldsymbol{x}_n \mid C_1) = \pi N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma)$

$p(\boldsymbol{x}_n, C_2) = p(C_2)p(\boldsymbol{x}_n \mid C_2) = (1-\pi)N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_2, \Sigma)$

Assuming observations are drawn independently, we can write the likelihood function as follows:

$$p(\mathrm{t} \mid \pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N} p(t_n \mid \pi, \mu_1, \mu_2, \Sigma) =$$

$$\prod_{n=1}^{N} \left[p(\mathrm{x}_n, C_1)\right]^{t_n} \left[p(\mathrm{x}_n, C_2)\right]^{1-t_n} =$$

$$\prod_{n=1}^{N} \left[\pi N(\mathrm{x}_n \mid \mu_1, \Sigma)\right]^{t_n} \left[(1-\pi)N(\mathrm{x}_n \mid \mu_2, \Sigma)\right]^{1-t_n} \qquad \mathrm{t} = (t_1, \cdots, t_N)^T$$

Maximum likelihood solution

We want to find the values of the parameters that maximize the likelihood function, i.e., fit a model that best describes the observed data.

$$p(\mathrm{t} \mid \pi, \mu_1, \mu_2, \Sigma) =$$

$$\prod_{n=1}^{N} \left[\pi N(\mathrm{x}_n \mid \mu_1, \Sigma)\right]^{t_n} \left[(1-\pi) N(\mathrm{x}_n \mid \mu_2, \Sigma)\right]^{1-t_n}$$

As usual, we consider the log of the likelihood:

$$\ln p(\boldsymbol{t} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) =$$

$$\sum_{n=1}^{N} \left[t_n \ln \pi + t_n \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) + (1-t_n) \ln(1-\pi) + (1-t_n) \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_2, \Sigma)\right]$$

Maximum likelihood solution

$$\ln p(\boldsymbol{t} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) =$$

$$\sum_{n=1}^{N} \left[t_n \ln \pi + t_n \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) + (1-t_n) \ln(1-\pi) + (1-t_n) \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_2, \Sigma)\right]$$

We first maximize the log likelihood with respect to π . The terms that depend on π are $\sum_{n=1}^{N} \left[t_n \ln \pi + (1-t_n)(1-\pi)\right]$

$$\frac{\partial}{\partial \pi} \sum_{n=1}^{N} \left[t_n \ln \pi + (1-t_n) \ln(1-\pi)\right] = \frac{1}{\pi} \sum_{n=1}^{N} t_n - \frac{1}{1-\pi} \sum_{n=1}^{N} (1-t_n)$$

$$= \frac{1}{\pi} \sum_{n=1}^{N} t_n - \frac{1}{1-\pi} N + \frac{1}{1-\pi} \sum_{n=1}^{N} t_n = \frac{1}{\pi(1-\pi)} \sum_{n=1}^{N} t_n - \frac{1}{1-\pi} N = 0$$

$$\Rightarrow \pi = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

Maximum likelihood solution

$$\pi_{ML} = \frac{N_1}{N_1 + N_2}$$

Thus, the maximum likelihood estimate of π is the fraction of points in class C_1

The result can be generalized to the multiclass case: the maximum likelihood estimate of $p(C_k)$ is given by the fraction of points in the training set that belong to C_k

Maximum likelihood solution

$$\ln p(\boldsymbol{t} \mid \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) =$$

$$\sum_{n=1}^{N} \left[t_n \ln \pi + t_n \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) + (1 - t_n) \ln(1 - \pi) + (1 - t_n) \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_2, \Sigma) \right]$$

We now maximize the log likelihood with respect to $\boldsymbol{\mu}_1$. The terms that depend on $\boldsymbol{\mu}_1$ are $\sum_{n=1}^{N} \left[t_n \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) \right]$

$$\frac{\partial}{\partial \boldsymbol{\mu}_1} \sum_{n=1}^{N} \left[t_n \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) \right] = \frac{\partial}{\partial \boldsymbol{\mu}_1} \left[-\frac{1}{2} \sum_{n=1}^{N} \left[t_n (\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1) \right] + const \right]$$

$$= \sum_{n=1}^{N} \left[t_n (\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \Sigma^{-1} \right] = 0 \Rightarrow \sum_{n=1}^{N} \left[t_n (\boldsymbol{x}_n - \boldsymbol{\mu}_1) \right] = 0 \Rightarrow \sum_{n=1}^{N} t_n \boldsymbol{x}_n = N_1 \boldsymbol{\mu}_1$$

$$\Rightarrow \boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \boldsymbol{x}_n$$

$$[N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_1, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{\left(-\frac{1}{2} (\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1) \right)}]$$

Maximum likelihood solution

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \boldsymbol{x}_n$$

Thus, the maximum likelihood estimate of $\boldsymbol{\mu}_1$ is the sample mean of all the input vectors $\boldsymbol{x}_n$ assigned to class C_1

By maximizing the log likelihood with respect to $\boldsymbol{\mu}_2$ we obtain a similar result for $\boldsymbol{\mu}_2$

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - t_n)\boldsymbol{x}_n$$

Maximum likelihood solution

Maximizing the log likelihood with respect to Σ we obtain the maximum likelihood estimate Σ_{ML}

$$\Sigma_{ML} = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2$$

$$S_1 = \frac{1}{N_1}\sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{\mu}_1)(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \quad S_2 = \frac{1}{N_2}\sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{\mu}_2)(\boldsymbol{x}_n - \boldsymbol{\mu}_2)^T$$

- Thus: the maximum likelihood estimate of the covariance is given by the weighted average of the sample covariance matrices associated with each of the classes.
- This results extend to K classes.

Probabilistic Discriminative Models

Two-class case: $p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0)$

Multiclass case: $p(C_k \mid \boldsymbol{x}) = \dfrac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T x + w_{j0}}}$

Discriminative approach: use the functional form of the generalized linear model for the posterior probabilities and determine its parameters directly using maximum likelihood.

Probabilistic Discriminative Models

Advantages:

- Fewer parameters to be determined
- Improved predictive performance, especially when the class-conditional density assumptions give a poor approximation of the true distributions.

Probabilistic Discriminative Models

Two-class case:

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0) = y(\boldsymbol{x})$$

$$p(C_2 \mid \boldsymbol{x}) = 1 - p(C_1 \mid \boldsymbol{x})$$

In the terminology of statistics, this model is known as **logistic regression**.

Assuming $\boldsymbol{x} \in \Re^M$ how many parameters do we need to estimate?

$$M + 1$$

Probabilistic Discriminative Models

How many parameters did we estimate to fit Gaussian class-conditional densities (generative approach)?

$$p(C_1) \Rightarrow 1$$

$$2 \ mean \ vectors \Rightarrow 2M$$

$$\Sigma \Rightarrow M + \frac{M^2 - M}{2} = \frac{M^2 + M}{2}$$

$$total = 1 + 2M + \frac{M^2 + M}{2} = O(M^2)$$

Logistic Regression

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0) = y(\boldsymbol{x})$$

We use maximum likelihood to determine the parameters of the logistic regression model.

$\{\boldsymbol{x}_n, t_n\} \quad n = 1, \cdots, N$

$t_n = 1$ *denotes class* $C_1 \qquad t_n = 0$ *denotes class* C_2

We want to find the values of w *that* max*imize the posterior probabilities associated to the observed data*

Likelihood function:

$$L(\boldsymbol{w}) = \prod_{n=1}^{N} P(C_1 \mid \boldsymbol{x}_n)^{t_n} \left(1 - P(C_1 \mid \boldsymbol{x}_n)\right)^{1-t_n} = \prod_{n=1}^{N} y(\boldsymbol{x}_n)^{t_n} \left(1 - y(\boldsymbol{x}_n)\right)^{1-t_n}$$

Logistic Regression

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0) = y(\boldsymbol{x})$$

$$L(\boldsymbol{w}) = \prod_{n=1}^{N} P(C_1 \mid \boldsymbol{x}_n)^{t_n} \left(1 - P(C_1 \mid \boldsymbol{x}_n)\right)^{1-t_n} = \prod_{n=1}^{N} y(\boldsymbol{x}_n)^{t_n} \left(1 - y(\boldsymbol{x}_n)\right)^{1-t_n}$$

We consider the negative logarithm of the likelihood:

$$E(\boldsymbol{w}) = -\ln L(\boldsymbol{w}) = -\ln \prod_{n=1}^{N} y(\boldsymbol{x}_n)^{t_n} \left(1 - y(\boldsymbol{x}_n)\right)^{1-t_n} =$$

$$-\sum_{n=1}^{N} \left(t_n \ln y(\boldsymbol{x}_n) + (1 - t_n) \ln(1 - y(\boldsymbol{x}_n))\right)$$

$$\arg\min_{\boldsymbol{w}} E(\boldsymbol{w})$$

Logistic Regression

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0) = y(\boldsymbol{x})$$

We compute the derivative of the error function with respect to $\boldsymbol{w}$ (gradient):

$$\frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w}) = \frac{\partial}{\partial \boldsymbol{w}} \left[-\sum_{n=1}^{N} \left(t_n \ln y(\boldsymbol{x}_n) + (1 - t_n) \ln(1 - y(\boldsymbol{x}_n)) \right) \right]$$

We need to compute the derivative of the logistic sigmoid function:

$$\frac{\partial}{\partial a} \sigma(a) = \frac{\partial}{\partial a} \frac{1}{1 + e^{-a}} = \frac{e^{-a}}{\left(1 + e^{-a}\right)^2} = \frac{1}{1 + e^{-a}} \frac{e^{-a}}{1 + e^{-a}} =$$

$$\frac{1}{1 + e^{-a}} \left(1 - \frac{1}{1 + e^{-a}} \right) = \sigma(a)(1 - \sigma(a))$$

Logistic Regression

$$p(C_1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x} + w_0) = y(\boldsymbol{x})$$

$$\frac{\partial}{\partial \boldsymbol{w}} E(\boldsymbol{w}) = \frac{\partial}{\partial \boldsymbol{w}} \left[-\sum_{n=1}^{N} \left(t_n \ln y(\boldsymbol{x}_n) + (1 - t_n) \ln(1 - y(\boldsymbol{x}_n)) \right) \right] =$$

$$-\sum_{n=1}^{N} \left(\frac{t_n}{y_n} y_n (1 - y_n) \boldsymbol{x}_n + \frac{(1 - t_n)}{(1 - y_n)} (-y_n)(1 - y_n) \boldsymbol{x}_n \right) =$$

$$-\sum_{n=1}^{N} \left(t_n (1 - y_n) \boldsymbol{x}_n - (1 - t_n) y_n \boldsymbol{x}_n \right) = -\sum_{n=1}^{N} \left(t_n - t_n y_n - y_n + t_n y_n \right) \boldsymbol{x}_n =$$

$$-\sum_{n=1}^{N} (t_n - y_n) \boldsymbol{x}_n = \sum_{n=1}^{N} (y_n - t_n) \boldsymbol{x}_n$$

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - t_n) \boldsymbol{x}_n$$

Logistic Regression

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - t_n)\mathbf{x}_n$$

- The gradient of E at $\boldsymbol{w}$ gives the direction of the steepest increase of E at $\boldsymbol{w}$. We need to minimize E. Thus we need to update $\boldsymbol{w}$ so that we move along the opposite direction of the gradient: $\boldsymbol{w}^{t+1} - \boldsymbol{w}^t \propto -\nabla E(\boldsymbol{w})$

This technique is called ***gradient descent***

- It can be shown that E is a concave function of $\boldsymbol{w}$. Thus, it has a unique minimum.
- An efficient iterative technique exists to find the optimal $\boldsymbol{w}$ parameters (*Newton-Raphson optimization*).

Batch vs. on-line learning

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - t_n)\mathbf{x}_n$$

- The computation of the above gradient requires the processing of the entire training set (*batch* technique)

$$\boldsymbol{w}^{t+1} - \boldsymbol{w}^t \propto -\nabla E(\boldsymbol{w})$$

- If the data set is large, the above technique can be costly;
- For real time applications in which data become available as continuous streams, we may want to update the parameters as data points are presented to us (*on-line* technique).

On-line learning

- After the presentation of each data point n, we compute the contribution of that data point to the gradient (stochastic gradient):

$$\nabla E_n(\boldsymbol{w}) = (y_n - t_n)\boldsymbol{x}_n$$

- The on-line updating rule for the parameters becomes:

$$\boldsymbol{w}^{t+1} - \boldsymbol{w}^t \propto -\nabla E_n(\boldsymbol{w})$$

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \nabla E_n(\boldsymbol{w}) = \boldsymbol{w}^t - \eta(y_n - t_n)\boldsymbol{x}_n$$

$\eta > 0$ *is called learning rate.*

It's value needs to be chosen carefully to ensure convergence

Multiclass Logistic Regression

Multiclass case: $p(C_k \mid \boldsymbol{x}) = \dfrac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T x + w_{j0}}} = y_k(\boldsymbol{x})$

We use maximum likelihood to determine the parameters of the logistic regression model.

$\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$ $n = 1, \cdots, N$

$t_n = (0, \cdots, 1, \cdots, 0)$ *denotes class* C_k

We want to find the values of $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K$ *that* max*imize the posterior probabilities associated to the observed data*

Likelihood function:

$$L(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} P(C_k \mid \boldsymbol{x}_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{x}_n)^{t_{nk}}$$

Multiclass Logistic Regression

$$p(C_k \mid \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}}} = y_k(\boldsymbol{x})$$

$$L(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(\boldsymbol{x}_n)^{t_{nk}}$$

We consider the negative logarithm of the likelihood:

$$E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = -\ln L(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{x}_n)$$

$$\arg\min_{\boldsymbol{w}_j} E(\boldsymbol{w}_j)$$

Multiclass Logistic Regression

$$p(C_k \mid \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}}} = y_k(\boldsymbol{x})$$

$$E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{x}_n)$$

We compute the gradient of the error function with respect to one of the parameter vectors:

$$\frac{\partial}{\partial \boldsymbol{w}_j} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \frac{\partial}{\partial \boldsymbol{w}_j} \left[-\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{x}_n) \right]$$

Multiclass Logistic Regression

$$p(C_k \mid \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}}} = y_k(\boldsymbol{x})$$

$$\frac{\partial}{\partial \boldsymbol{w}_j} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \frac{\partial}{\partial \boldsymbol{w}_j} \left[-\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} y_k(\boldsymbol{x}_n) \right]$$

Thus, we need to compute the derivatives of the softmax function:

$$\frac{\partial}{\partial a_k} y_k = \frac{\partial}{\partial a_k} \frac{e^{a_k}}{\sum_j e^{a_j}} = \frac{e^{a_k} \sum_j e^{a_j} - e^{a_k} e^{a_k}}{\left(\sum_j e^{a_j} \right)^2} = y_k - y_k^2 = y_k(1 - y_k)$$

Multiclass Logistic Regression

$$p(C_k \mid \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}}} = y_k(\boldsymbol{x})$$

$$\frac{\partial}{\partial \boldsymbol{w}_j} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \frac{\partial}{\partial \boldsymbol{w}_j} \left[-\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} y_k(\boldsymbol{x}_n) \right]$$

Thus, we need to compute the derivatives of the softmax function:

$$\text{for } j \neq k, \quad \frac{\partial}{\partial a_j} y_k = \frac{\partial}{\partial a_j} \frac{e^{a_k}}{\sum_j e^{a_j}} = \frac{-e^{a_k} e^{a_j}}{\left(\sum_j e^{a_j} \right)^2} = -y_k y_j$$

Multiclass Logistic Regression

$$\frac{\partial}{\partial a_k} y_k = \frac{\partial}{\partial a_k} \frac{e^{a_k}}{\sum_j e^{a_j}} = \frac{e^{a_k} \sum_j e^{a_j} - e^{a_k} e^{a_k}}{\left(\sum_j e^{a_j} \right)^2} = y_k - y_k^2 = y_k (1 - y_k)$$

$$for \;\; j \neq k, \;\; \frac{\partial}{\partial a_j} y_k = \frac{\partial}{\partial a_j} \frac{e^{a_k}}{\sum_j e^{a_j}} = \frac{-e^{a_k} e^{a_j}}{\left(\sum_j e^{a_j} \right)^2} = -y_k y_j$$

Compact expression:

$$\frac{\partial}{\partial a_j} y_k = y_k (I_{kj} - y_j)$$

where I_{kj} are the elements of the identity matrix

Multiclass Logistic Regression

$$p(C_k \mid \boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}}}{\sum_j e^{\boldsymbol{w}_j^T x + w_{j0}}} = y_k(\boldsymbol{x})$$

$$\frac{\partial}{\partial a_j} y_k = y_k (I_{kj} - y_j)$$

$$\nabla_{\boldsymbol{w}_j} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \frac{\partial}{\partial \boldsymbol{w}_j} \left[- \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\boldsymbol{x}_n) \right] =$$

$$- \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \frac{1}{y_{nk}} y_{nk} (I_{kj} - y_{nj}) \boldsymbol{x}_n = \sum_{n=1}^{N} \sum_{k=1}^{K} (t_{nk} y_{nj} - t_{nk} I_{kj}) \boldsymbol{x}_n =$$

$$\sum_{n=1}^{N} (y_{nj} - t_{nj}) \boldsymbol{x}_n$$

Multiclass Logistic Regression

$$\nabla_{\boldsymbol{w}_j} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \boldsymbol{x}_n$$

- It can be shown that E is a concave function of $\boldsymbol{w}$. Thus, it has a unique minimum.
- For a batch solution, we can use the *Newton-Raphson optimization* technique.
- On-line solution (stochastic gradient descent):

$$\boldsymbol{w}_j^{t+1} = \boldsymbol{w}_j^t - \eta \nabla E_n(\boldsymbol{w}) = \boldsymbol{w}_j^t - \eta (y_{nj} - t_{nj}) \boldsymbol{x}_n$$