LINEAR MODELS FOR CLASSIFICATION
Classification: Problem Statement

- In regression, we are modeling the relationship between a continuous input variable \( x \) and a continuous target variable \( t \).
- In classification, the input variable \( x \) is still continuous, but the target variable is discrete.
- In the simplest case, \( t \) can have only 2 values.
Example Problem

- Animal or Vegetable?
Linear models for classification separate input vectors into classes using linear decision boundaries.

Example:

Input vector \( \mathbf{x} \)

Two discrete classes \( C_1 \) and \( C_2 \)
Discriminant Functions

A linear discriminant function \( y(x) = f \left( w^t x + w_0 \right) \) maps a real input vector \( x \) to a scalar value \( y(x) \).

\( f(\cdot) \) is called an activation function.
Outline

- Linear activation functions
  - Least-squares formulation
  - Fisher’s linear discriminant

- Nonlinear activation functions
  - Probabilistic generative models
  - Probabilistic discriminative models
    - Logistic regression
    - Bayesian logistic regression
Two Class Discriminant Function

Let $f(\cdot)$ be the identity:

$y(x) = w^T x + w_0$

$y(x) \geq 0 \rightarrow x$ assigned to $C_1$

$y(x) < 0 \rightarrow x$ assigned to $C_2$

Thus $y(x) = 0$ defines the decision boundary
K > 2 Classes

- Idea #1: Just use $K-1$ discriminant functions, each of which separates one class $C_k$ from the rest. (One-versus-the-rest classifier.)

- Problem: Ambiguous regions
K>2 Classes

- Idea #2: Use $K(K-1)/2$ discriminant functions, each of which separates two classes $C_j$, $C_k$ from each other. (One-versus-one classifier.)
- Each point classified by majority vote.
- Problem: Ambiguous regions
K>2 Classes

- Idea #3: Use K discriminant functions $y_k(x)$
- Use the **magnitude** of $y_k(x)$, not just the sign.

$$y_k(x) = w_k^t x + w_{k0}$$

$x$ assigned to $C_k$ if $y_k(x) > y_j(x) \forall j \neq k$

Decision boundary $y_k(x) = y_j(x) \rightarrow (w_k - w_j)^t x + (w_{k0} - w_{j0}) = 0$

Results in decision regions that are simply-connected and convex.
Learning the Parameters

Method #1: Least Squares

\[ y_k(x) = w_k^t x + w_{k0} \]

\[ \rightarrow y(x) = \tilde{W}^t \tilde{x} \]

where
\[ \tilde{x} = (1, x^t)^t \]

\[ \tilde{W} \] is a \((D + 1) \times K\) matrix whose kth column is \(\tilde{w}_k = (w_0, w_k^t)^t\)
Learning the Parameters

□ Method #1: Least Squares

\[ y(x) = \tilde{W}^T \tilde{x} \]

Training dataset \((x_n, t_n), \quad n = 1, \ldots, N\)

where we use the 1-of-\(K\) coding scheme for \(t_n\)

Let \(T\) be the \(N \times K\) matrix whose \(n^{th}\) row is \(t_n^T\)

Let \(\tilde{X}\) be the \(N \times (D + 1)\) matrix whose \(n^{th}\) row is \(\tilde{x}_n^T\)

We define the error as \(E_D(\tilde{W}) = \frac{1}{2} \text{Tr} \left\{ (\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T) \right\} \)

Setting derivative wrt \(\tilde{W}\) yields:

\[ \tilde{W} = (\tilde{X}'\tilde{X})^{-1} \tilde{X}'T = \tilde{X}'T \]
Fisher’s Linear Discriminant

Another way to view linear discriminants: find the 1D subspace that maximizes the separation between the two classes.

Let \( m_1 = \frac{1}{N_1} \sum_{n \in C_1} x_n \), \( m_2 = \frac{1}{N_2} \sum_{n \in C_2} x_n \)

For example, might choose \( w \) to maximize \( w^t (m_2 - m_1) \), subject to \( \|w\| = 1 \)

This leads to \( w \propto m_2 - m_1 \)

However, if conditional distributions are not isotropic, this is typically not optimal.
Let $m_1 = w^T m_1$, $m_2 = w^T m_2$ be the conditional means on the 1D subspace.

Let $s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$ be the within-class variance on the subspace for class $C_k$.

The Fisher criterion is then $J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$.

This can be rewritten as

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

where

$S_B = (m_2 - m_1)(m_2 - m_1)^T$ is the between-class variance

and

$S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$ is the within-class variance

$J(w)$ is maximized for $w \propto S_W^{-1}(m_2 - m_1)$.
Change coding scheme to
\[ t_n = \frac{N}{N_1} \text{ for } C_1 \]
\[ t_n = -\frac{N}{N_2} \text{ for } C_2 \]

Then one can show that the ML \( w \) satisfies
\[ w \propto S_w^{-1}(m_2 - m_1) \]
Least Squares Classifier

Problem #1: Sensitivity to outliers
Problem #2: Linear activation function is not a good fit to binary data. This can lead to problems.
Outline

- Linear activation functions
  - Least-squares formulation
  - Fisher’s linear discriminant

- Nonlinear activation functions
  - Probabilistic generative models
  - Probabilistic discriminative models
    - Logistic regression
    - Bayesian logistic regression
Probabilistic Generative Models

Consider first $K=2$:

By Bayes' equation, the posterior for class $C_1$ can be written:

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

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

where

$$a = \log \frac{p(x | C_1) p(C_1)}{p(x | C_2) p(C_2)}$$

and $\sigma(a)$ is the logistic sigmoid function.
Let's assume that the input vector $\mathbf{x}$ is multivariate normal, when conditioned upon the class $C_k$, and that the covariance is the same for all classes:

$$p(\mathbf{x} | C_k) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_k)^t \Sigma^{-1} (\mathbf{x} - \mu_k) \right\}$$

Then we have that $p(C_1 | \mathbf{x}) = \sigma \left( \mathbf{w}^t \mathbf{x} + w_0 \right)$

where

$\mathbf{w} = \Sigma^{-1} (\mu_1 - \mu_2)$

$w_0 = -\frac{1}{2} \mu_1^t \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^t \Sigma^{-1} \mu_2 + \log \frac{p(C_1)}{p(C_2)}$

Thus we have a generalized linear model, and the decision surfaces will be hyperplanes in the input space.
Probabilistic Generative Models

This result generalizes to $K > 2$ classes:

$$p(C_k | x) = \frac{p(x | C_k)p(C_k)}{\sum_j p(x | C_j)p(C_j)}$$

$$= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \quad \text{"softmax"}$$

where

$$a_k = \log(p(x | C_k)p(C_k))$$

Then we have that $a_k(x) = w_k^i x + w_{k0}$

where

$$w_k = \Sigma^{-1}\mu_k$$

$$w_{k0} = -\frac{1}{2} \mu_k^i \Sigma^{-1}\mu_k + \log p(C_k)$$
Non-Constant Covariance

- If the class-conditional covariances are different, the generative decision boundaries are in general quadratic.
ML for Probabilistic Generative Model

Let $t_n = 1$ denote Class 1, $t_n = 0$ denote Class 2.

Let $\pi = p(C_1)$ so that $1 - \pi = p(C_2)$

Then the ML estimates for the parameters are:

$$\pi = \frac{N_1}{N_1 + N_2}$$

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n x_n$$

$$\mu_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) x_n$$

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

where

$$S_1 = \frac{1}{N_1} \sum_{n \in C_1} (x_n - \mu_1)(x_n - \mu_1)^t$$

and

$$S_2 = \frac{1}{N_2} \sum_{n \in C_2} (x_n - \mu_2)(x_n - \mu_2)^t$$
Probabilistic Discriminative Models

- An alternative to the generative approach is to model the dependence of the target variable $t$ on the input vector $x$ directly, using the activation function $f$.

- One big advantage is that there will typically be fewer parameters to determine.
Logistic Regression ($K = 2$)

\[ p(C_1 | \phi) = y(\phi) = \sigma(w^t \phi) \]
\[ p(C_2 | \phi) = 1 - p(C_1 | \phi) \]

where $\sigma(a) = \frac{1}{1 - \exp(-a)}$

\[ p(C_1 | \phi) = y(\phi) = \sigma(w^t \phi) \]
Logistic Regression

\[ p(C_1 | \phi) = y(\phi) = \sigma(w^t \phi) \]
\[ p(C_2 | \phi) = 1 - p(C_1 | \phi) \]

where
\[ \sigma(a) = \frac{1}{1 - \exp(-a)} \]

- **Number of parameters**
  - Logistic regression: \( M \)
  - Generative model: \( 2M + M(M+1)/2 + 1 = M(M+5)/2+1 \)
ML for Logistic Regression

\[ p(t \mid w) = \prod_{n=1}^{N} y_n^{t_n} \left(1 - y_n \right)^{1-t_n} \]

where \( t = (t_1, \ldots, t_N)^t \) and \( y_n = p(C_1 \mid \phi_n) \)

We define the error function to be \( E(w) = -\log p(t \mid w) \)

Given \( y_n = \sigma(a_n) \) and \( a_n = w^t \phi_n \), one can show that

\[ \nabla E(w) = \sum_{n=1}^{N} (y_n - t_n) \phi_n \]

Unfortunately, there is no closed form solution for \( w \).
ML for Logistic Regression:

- Iterative Reweighted Least Squares

  Although there is no closed form solution for the ML estimate of $\mathbf{w}$, fortunately, the error function is convex.

  Thus an appropriate iterative method is guaranteed to find the exact solution.

  A good method is to use a local quadratic approximation to the log likelihood function (Newton-Raphson update):

  $$\mathbf{w}^{(new)} = \mathbf{w}^{(old)} - H^{-1} \nabla E(\mathbf{w})$$

  where $H$ is the Hessian matrix of $E(\mathbf{w})$.
ML for Logistic Regression

\[ w^{(new)} = w^{(old)} - H^{-1}\nabla E(w) \]

where \( H \) is the Hessian matrix of \( E(w) \):

\[ H = \Phi^t R \Phi \]

where \( R \) is the \( N \times N \) diagonal weight matrix with \( R_{nn} = y_n (1 - y_n) \)

(Note that, since \( R_{nn} \geq 0 \), \( R \) is positive semi-definite, and hence \( H \) is positive semi-definite. Thus \( E(w) \) is convex.)

Thus

\[ w^{new} = w^{(old)} - \left( \Phi^t R \Phi \right)^{-1} \Phi^t (y - t) \]
ML for Logistic Regression

- Iterative Reweighted Least Squares

\[ p(C_1 | \phi) = y(\phi) = \sigma(w^t \phi) \]
Bayesian Logistic Regression

We can make logistic regression Bayesian by applying a prior over \( w \):

\[
\pi(w) = N(w | m_0, S_0)
\]

- Unfortunately, the posterior over \( w \) will not be normal for logistic regression, and hence we cannot integrate over it analytically.
- This means that we cannot do Bayesian prediction analytically.
- However, there are methods for approximating the posterior that allow us to do approximate Bayesian prediction.
The Laplace Approximation

- In the Laplace approximation, we approximate the log of a distribution by a local, second order (quadratic) form, centred at the mode.
- This corresponds to a normal approximation to the distribution, with
  - mean given by the mode of the original distribution
  - precision matrix given by the Hessian of the negative log of the distribution

![Graphs showing p(z) and -log p(z)]
Bayesian Logistic Regression

- When applied to the posterior over $\mathbf{w}$ in logistic regression, this yields

$$p(\mathbf{w}) = q(\mathbf{w}) = N(\mathbf{w} | \mathbf{w}_{\text{MAP}}, \mathbf{S}_N)$$

where

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \sum_{n=1}^{N} y_n (1 - y_n) \phi_n \phi_n^t$$
Bayesian prediction requires that we integrate out this posterior over $w$:

$$p(C_1 | \phi, t) = \int p(C_1 | \phi, w)p(w | t)dw \approx \int \sigma(w^t \phi)q(w)dw$$

This integral is not tractable analytically. However, approximation of the sigmoid function $\sigma(\cdot)$ by the inverse probit (cumulative normal) function yields an analytical solution:

$$p(C_1 | \phi, t) \approx \sigma(\kappa(\sigma^2_a)\mu_a),$$

where $\mu_a = w^t_{MAP} \phi$, $\sigma^2_a = \phi^t S_N \phi$ and $\kappa(\sigma^2_a) = \left(1 + \pi \sigma^2_a / 8\right)^{-1/2}$.
This last approximation is excellent!