Recap: Linear Discriminant Functions

- Basic idea
  - Directly encode decision boundary
  - Minimize misclassification probability directly.

- Linear discriminant functions
  \[ y(x) = w^T x + w_0 \]
  - \( w, w_0 \) define a hyperplane in \( \mathbb{R}^2 \).
  - If a data set can be perfectly classified by a linear discriminant, then we call it linearly separable.

Recap: Least-Squares Classification

- Simplest approach
  - Directly try to minimize the sum-of-squares error
    \[ E(w) = \sum_{n=1}^{N} (y(x_n) - t_n)^2 \]
    \[ E_D(W) = \frac{1}{2} \text{trace} \left( (XW - T)^T(XW - T) \right) \]
  - Setting the derivative to zero yields
    \[ W = (X^T X)^{-1} X^T T = (X^T T)^{-1} X^T T \]
  - We then obtain the discriminant function as
    \[ y(x) = W^T x = T^T \left( \frac{x}{x} \right) \]
  - Exact, closed-form solution for the discriminant function parameters.

Recap: Problems with Least Squares

- Least-squares is very sensitive to outliers!
  - The error function penalizes predictions that are “too correct”.

Recap: Generalized Linear Models

- Generalized linear model
  \[ y(x) = g(w^T x + w_0) \]
  - \( g(\cdot) \) is called an activation function and may be nonlinear.
  - The decision surfaces correspond to
    \[ y(x) = \text{const} \iff w^T x + w_0 = \text{const} \]
  - If \( g \) is monotonous (which is typically the case), the resulting decision boundaries are still linear functions of \( x \).

- Advantages of the non-linearity
  - Can be used to bound the influence of outliers and “too correct” data points.
  - When using a sigmoid for \( g(\cdot) \), we can interpret the \( y(x) \) as posterior probabilities.
Recap: Linear Separability
- Up to now: restrictive assumption
  - Only consider linear decision boundaries
- Classical counterexample: XOR

Recap: Extension to Nonlinear Basis Fcts.
- Generalization
  - Transform vector $x$ with $M$ nonlinear basis functions $\phi_j(x)$:
    $y_k(x) = \sum_{j=1}^{M} w_{kj} \phi_j(x) + w_{k0}$
- Advantages
  - Transformation allows non-linear decision boundaries.
  - By choosing the right $\phi$, every continuous function can (in principle) be approximated with arbitrary accuracy.
- Disadvantage
  - The error function can in general no longer be minimized in closed form.
    $\Rightarrow$ Minimization with Gradient Descent

Recap: Gradient Descent
- Iterative minimization
  - Start with an initial guess for the parameter values $w_k(0)$.
  - Move towards a (local) minimum by following the gradient.
- Basic strategies
  - "Batch learning"
    $w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \frac{\partial E(w)}{\partial w_{kj}} |_{w^{(r)}}$
  - "Sequential updating"
    $w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \frac{\partial E_n(w)}{\partial w_{kj}} |_{w^{(r)}}$
    where $E(w) = \sum_{n=1}^{N} E_n(w)$
- Example: Quadratic error function
  $E(w) = \sum (y(x_n; w) - t_n)^2$
- Sequential updating leads to delta rule (=LMS rule)
  $w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta (y_k(x_n; w) - t_{kn}) \phi_j(x_n)$
  $= w_{kj}^{(r)} - \eta \delta_{kn} \phi_j(x_n)$
  where $\delta_{kn} = y_k(x_n; w) - t_{kn}$
  $\Rightarrow$ Simply feed back the input data point, weighted by the classification error.

Recap: Gradient Descent
- Cases with differentiable, non-linear activation function
  $y_k(x) = g(a_k) = g \left( \sum_{j=0}^{M} w_{kj} \phi_j(x_n) \right)$
- Gradient descent (again with quadratic error function)
  $\frac{\partial E_n(w)}{\partial w_{kj}} = \frac{\partial g(a_k)}{\partial w_{kj}} (y_k(x_n; w) - t_{kn}) \phi_j(x_n)$
  $w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \frac{\partial E_n(w)}{\partial w_{kj}} |_{w^{(r)}}$
  $\delta_{kn} = \frac{\partial g(a_k)}{\partial w_{kj}} (y_k(x_n; w) - t_{kn})$

Topics of This Lecture
- Fisher’s linear discriminant (FLD)
  - Classification as dimensionality reduction
  - Linear discriminant analysis
  - Multiple discriminant analysis
  - Applications
- Logistic Regression
  - Probabilistic discriminative models
  - Logistic sigmoid (logit function)
  - Cross-entropy error
  - Gradient descent
  - Iteratively Reweighted Least Squares
- Note on Error Functions
Classification as Dimensionality Reduction

• Classification as dimensionality reduction
  
  - We can interpret the linear classification model as a projection onto a lower-dimensional space.
  - E.g., take the $D$-dimensional input vector $x$ and project it down to one dimension by applying the function $y = W^T x$
  - If we now place a threshold at $y \geq -w_0$, we obtain our standard two-class linear classifier.
  - The classifier will have a lower error the better this projection separates the two classes.

⇒ New interpretation of the learning problem

• Try to find the projection vector $w$ that maximizes the class separation.

Two questions

- How to measure class separation?
- How to find the best projection (with maximal class separation)?

Problems with this approach

1. This expression can be made arbitrarily large by increasing $||w||$.
   ⇒ Need to enforce additional constraint $||w|| = 1$.

2. The criterion may result in bad separation if the clusters have elongated shapes.

Classification as Dimensionality Reduction

• Measuring class separation
  
  - We could simply measure the separation of the class means.
    ⇒ Choose $w$ so as to maximize $(m_2 - m_1) = W^T (m_2 - m_1)$

• Problems with this approach

  1. This expression can be made arbitrarily large by increasing $||w||$.
    ⇒ Need to enforce additional constraint $||w|| = 1$.
    ⇒ This constrained minimization results in $w \propto (m_2 - m_1)$
  2. The criterion may result in bad separation if the clusters have elongated shapes.

Fisher’s Linear Discriminant Analysis (FLD)

• Better idea:
  
  - Find a projection that maximizes the ratio of the between-class variance to the within-class variance:
    $$J(w) = \frac{(m_2 - m_1)^2}{s_1 + s_2}$$
    with $s^2_k = \sum y_n (m_n - m_k)^2$
  
  - Usually, this is written as
    $$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

  - where
    $$S_B = (m_2 - m_1)(m_2 - m_1)^T$$
    $$S_W = \sum_{k=1}^{K} \sum_{x_n \in C_k} (x_n - m_k)(x_n - m_k)^T$$

  - between-class scatter matrix
  - within-class scatter matrix

Multiple Discriminant Analysis

• Generalization to $K$ classes
  
  $$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}$$

  - where
    $$W = [w_1, \ldots, w_K]$$
    $$m = \frac{1}{N} \sum_{n=1}^{N} x_m = \frac{1}{N} \sum_{k=1}^{K} N_k m_k$$
    $$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$$
    $$S_W = \sum_{k=1}^{K} \sum_{x_n \in C_k} (x_n - m_k)(x_n - m_k)^T$$

  - between-class scatter matrix
  - within-class scatter matrix

Image source: C.M. Bishop, 2006

Minimize distance within a class

Maximize distance between classes

Classification as Dimensionality Reduction

Image source: C.M. Bishop, 2006

Classification function:

$$y(x) = W^T x + w_0 \begin{cases} \geq 0 & \text{Class 1} \\ < 0 & \text{Class 2} \end{cases}$$

where $w_0 = -W^T m$
Maximizing $J(W)$

- Generalized eigenvalue problem
  \[ J(W) = |W^T S_i W| \]
  \[ |W^T S_w W| \]
  - The columns of the optimal $W$ are the eigenvectors corresponding to the largest eigenvalues of $S_i w_i = \lambda_i S_w w_i$
  - Defining $v = S_{i}^{-\frac{1}{2}} w_i$, we get $S_{i}^{-\frac{1}{2}} S_w^{-\frac{1}{2}} v = \lambda v$
    which is a regular eigenvalue problem.
  - Solve to get eigenvectors of $v$, then from that of $w$.
  - For the $K$-class case we obtain (at most) $K-1$ projections.

- (i.e. eigenvectors corresponding to non-zero eigenvalues.)

What Does It Mean?

- What does it mean to apply a linear classifier?
  \[ y(x) = w^T x \]
  Weight vector
  Input vector

- Classifier interpretation
  - The weight vector has the same dimensionality as $x$.
  - Positive contributions where $\text{sign}(w_i) = \text{sign}(w_j)$.
  - The weight vector identifies which input dimensions are important for positive or negative classification (large $w_i$) and which ones are irrelevant (near-zero $w_i$).
  - If the inputs $x$ are normalized, we can interpret $w$ as a “template” vector that the classifier tries to match.
  \[ w^T x = |w||x| \cos \theta \]

Example Application: Fisherfaces

- Visual discrimination task
  - Training data:
    \[ C_1: \text{Subjects with glasses} \quad C_2: \text{Subjects without glasses} \]
  - Test:
    - glasses?

Take each image as a vector of pixel values and apply FLD...

Summary: Fisher’s Linear Discriminant

- Properties
  - Simple method for dimensionality reduction, preserves class discriminability.
  - Can use parametric methods in reduced-dim. space that might not be feasible in original higher-dim. space.
  - Widely used in practical applications.

- Restrictions / Caveats
  - Not possible to get more than $K-1$ projections.
  - FLD reduces the computation to class means and covariances.
  - Implicit assumption that class distributions are unimodal and well-approximated by a Gaussian/hyperellipsoid.

Fisherases: Interpretability

- Resulting weight vector for “Glasses/NoGlasses”

Topics of This Lecture

- Fisher’s Linear Discriminant (FLD)
  - Classification as dimensionality reduction
  - Linear discriminant analysis
  - Multiple discriminant analysis
  - Applications

- Logistic Regression
  - Probabilistic discriminative models
  - Logistic sigmoid (logit function)
  - Cross-entropy error
  - Gradient descent
  - Iteratively Reweighted Least Squares

- Noise on Error Functions
**Probabilistic Discriminative Models**

- We have seen that we can write
  \[ p(C_1|x) = \sigma(a) = \frac{1}{1 + \exp(-a)} \]  
  logistic sigmoid function

- We can obtain the familiar probabilistic model by setting
  \[ a = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)} \]

- Or we can use generalized linear discriminant models
  \[ a = w^T x \]
  or \[ a = w^T \phi(x) \]

**Comparison**

- Let’s look at the number of parameters...
  - Assume we have an \( M \)-dimensional feature space \( \phi \).
  - And assume we represent \( p(C_1) \) and \( p(C_2) \) by Gaussians.
  - How many parameters do we need?
    - For the means: \( 2M \)
    - For the covariances: \( MMM+1/2 \)
    - Together with the class priors, this gives \( (M+1)(M+1) \) parameters!
  - How many parameters do we need for logistic regression?
    - \[ p(C_1|\phi) = y(\phi) = \sigma(w^T \phi) \]
    - Just the values of \( w \) as \( M \) parameters.

⇒ **For large \( M \), logistic regression has clear advantages!**

**Logistic Regression**

- Let’s consider a data set \( \{ \phi_n, t_n \} \) with \( n = 1, \ldots, N \), where \( \phi_n = \phi(x_n) \) and \( t_n \in \{0,1\} \), \( t = (t_1, \ldots, t_N)^T \).

- With \( y_n = p(C_1|\phi_n) \), we can write the likelihood as
  \[ p(t|w) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n} \]

- Define the error function as the negative log-likelihood
  \[ E(w) = -\ln p(t|w) \]
  \[ = -\sum_{n=1}^{N} \{ t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \} \]

  - This is the so-called cross-entropy error function.

**Gradient of the Error Function**

- Error function
  \[ E(w) = -\sum_{n=1}^{N} \{ t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \} \]

- Gradient
  \[ \nabla E(w) = -\sum_{n=1}^{N} \left\{ t_n \frac{\partial}{\partial \phi_n} \ln y_n + (1 - t_n) \frac{\partial}{\partial \phi_n} \ln (1 - y_n) \right\} \]
  \[ = -\sum_{n=1}^{N} \left\{ \frac{t_n}{y_n} \phi_n - \frac{1 - t_n}{1 - y_n} \phi_n \right\} \]
  \[ = -\sum_{n=1}^{N} \left\{ \frac{t_n}{y_n} \phi_n - \frac{1 - t_n}{1 - y_n} \phi_n \right\} \]
  \[ = \sum_{n=1}^{N} \{ y_n - t_n \} \phi_n \]

**Probabilistic Discriminative Models**

- In the following, we will consider models of the form
  \[ p(C_1|\phi) = y(\phi) = \sigma(w^T \phi) \]
  with \[ p(C_2|\phi) = 1 - p(C_1|\phi) \]

- This model is called logistic regression.

- Why should we do this? What advantage does such a model have compared to modeling the probabilities?
  \[ p(C_1|\phi) = \frac{p(\phi|C_1)p(C_1)}{p(\phi|C_1)p(C_1) + p(\phi|C_2)p(C_2)} \]

- Any ideas?
Gradient of the Error Function

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

- Does this look familiar to you?
- This is the same result as for the Delta (=LMS) rule
  \[ w^{r+1}_{kj} = w^{r}_{kj} - \eta (y(x_n; w) - t_k_n) \phi_j(x_n) \]
- We can use this to derive a sequential estimation algorithm.
  - However, this will be quite slow...

Newton-Raphson for Least-Squares Estimation

- Let’s first apply Newton-Raphson to the least-squares error function:
  \[ E(w) = \frac{1}{2} \sum_{n=1}^{N} (w^T \phi_n - t_n)^2 \]
  \[ \nabla E(w) = \sum_{n=1}^{N} (w^T \phi_n - t_n) \phi_n - \Phi^T \Phi w - \Phi^T t \]
  \[ H = \nabla^2 E(w) = \sum_{n=1}^{N} \phi_n \phi_n^T = \Phi^T \Phi \quad \text{where} \quad \Phi = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_N^T \end{bmatrix} \]
- Resulting update scheme:
  \[ w^{(r+1)} = w^{(r)} - (\Phi^T \Phi)^{-1} \Phi^T \left( \Phi w^{(r)} - \Phi^T \tau \right) \]
  \[ = (\Phi^T \Phi)^{-1} \Phi^T \tau \quad \text{Closed-form solution!} \]

Iteratively Reweighted Least Squares

- Update equations
  \[ w^{(r+1)} = w^{(r)} - (\Phi^T \Phi)^{-1} \Phi^T (y - t) \]
  \[ = (\Phi^T \Phi)^{-1} \left( \Phi^T \Phi w^{(r)} - \Phi^T y - t \right) \]
  \[ = (\Phi^T \Phi)^{-1} \Phi^T R \Phi w^{(r)} \]
  with \( z = \Phi w^{(r)} - R^{-1}(y - t) \)
- Again very similar form (normal equations)
  - But now with non-constant weighing matrix \( R \) (depends on \( w \)).
  - Need to apply normal equations iteratively.
  - Naturally Reweighted Least-Squares (IRLS)

Newton-Raphson for Logistic Regression

- Now, let’s try Newton-Raphson on the cross-entropy error function:
  \[ E(w) = -\sum_{n=1}^{N} \left( t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \right) \]
  \[ \nabla E(w) = \sum_{n=1}^{N} (y_n - t_n) \phi_n = \Phi^T (y - t) \]
  \[ H = \nabla^2 E(w) = \sum_{n=1}^{N} y_n(1 - y_n) \phi_n \phi_n^T = \Phi^T R \Phi \]
  \[ \text{where} \quad R \text{ is an } N \times N \text{ diagonal matrix with } R_{nn} = y_n(1 - y_n). \]
  - The Hessian is no longer constant, but depends on \( w \) through the weighting matrix \( R \).

Summary: Logistic Regression

- Properties
  - Directly represent posterior distribution \( p(\phi | C) \).
  - Requires fewer parameters than modeling the likelihood + prior.
  - Very often used in statistics.
  - It can be shown that the cross-entropy error function is concave.
    - Optimization leads to unique minimum
    - But no closed-form solution exists
    - Iterative optimization (IRLS)
  - Both online and batch optimizations exist
  - There is a multi-class version described in (Bishop Ch.4.3.4).

- Caveat
  - Logistic regression tends to systematically overestimate odds ratios when the sample size is less than ~500.
Topics of This Lecture

- Fisher's linear discriminant (FLD)
  - Classification as dimensionality reduction
  - Linear discriminant analysis
  - Multiple discriminant analysis
  - Applications
- Logistic Regression
  - Probabilistic discriminative models
  - Logistic sigmoid (logit function)
  - Cross-entropy error
  - Gradient descent
  - Iteratively Reweighted Least Squares
- Note on Error Functions

Note on Error Functions

- We have now seen already a number of error functions
  - Ideal misclassification error
  - Quadratic error
  - Cross-entropy error

Error Functions

- Ideal Misclassification Error
  - This is what we would like to optimize.
  - But cannot compute gradients here.

- Quadratic Error
  - Easy to optimize, closed-form solutions exist.
  - But not robust to outliers.

- Cross-Entropy Error
  - Minimizer of this error is given by posterior class probabilities.
  - Concave error function, unique minimum exists.
  - But no closed-form solution, requires iterative estimation.

References and Further Reading

- More information on Linear Discriminant Functions can be found in Chapter 4 of Bishop’s book (in particular Chapter 4.1 - 4.3).

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006