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 \( \mathbb{R}^d \).
  - 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
  \[ E(w) = \sum_{n=1}^{N} (y(x_n; w) - t_n)^2 \]
  \[ E_D(W) = \frac{1}{2} \text{Tr} \left\{ (XW - T)(XW - T)^T \right\} \]
  - Setting the derivative to zero yields
  \[ W = (X^T X)^{-1} X^T T = X^T X^{-1} X^T = X^T X^{-1} \]
  - We then obtain the discriminant function as
  \[ y(x) = W^T x = T^T \frac{1}{2} x \]
  \( \Rightarrow \) 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 \( y(x) \) as posterior probabilities.
Recap: Linear Separability

- Up to now: restrictive assumption
  - Only consider linear decision boundaries
- Classical counterexample: XOR

\[
\begin{array}{c|cc}
\times_1 & C_2 & C_1 \\
\hline
C_1 & \bullet & \cdot \\
C_2 & \cdot & \cdot \\
\end{array}
\]

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

Gradient Descent

- Iterative minimization
  - Start with an initial guess for the parameter values \( w_{kj}^{(0)} \).
  - Move towards a (local) minimum by following the gradient.
- Basic strategies
  - “Batch learning” \( w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \frac{\partial E(w)}{\partial w_{kj}} \bigg|_{w^{(r)}} \)
  - “Sequential updating” \( w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \frac{\partial E_n(w)}{\partial w_{kj}} \bigg|_{w_n^{(r)}} \)
  where \( E(w) = \sum_{n=1}^{N} E_n(w) \)

Gradient Descent

- Delta rule (=LMS rule)
  \[
  w_{kj}^{(r+1)} = w_{kj}^{(r)} \eta \left( y_k(x_n; w) - t_{kn} \right) \phi_j(x_n)
  = w_{kj}^{(r)} \eta \delta_{kn} \phi_j(x_n)
  \]
  where \( \delta_{kn} = y_k(x_n; w) - t_{kn} \)

  \( \Rightarrow \) Simply feed back the input data point, weighted by the classification error.

Gradient Descent

- Error function
  \[
  E(w) = \sum_{n=1}^{N} E_n(w) = \frac{1}{2} \sum_{n=1}^{N} \left( \sum_{j=1}^{K} w_{kj} \phi_j(x_n) - t_{kn} \right)^2
  \]
- Gradient descent
  \[
  \frac{\partial E_n(w)}{\partial w_{kj}} = \left( \sum_{j=1}^{M} w_{kj} \phi_j(x_n) - t_{kn} \right) \phi_j(x_n)
  = (y_k(x_n; w) - t_{kn}) \phi_j(x_n)
  \]
  where \( y_k(x_n; w) = g(ak) = g \left( \sum_{i=0}^{M} w_{ki} \phi_i(x_n) \right) \)

  \[ \frac{\partial E_n(w)}{\partial w_{kj}} = \frac{\partial g(ak)}{\partial w_{kj}} \left( \sum_{i=0}^{M} w_{ki} \phi_i(x_n) \right) \phi_j(x_n) \]
  \[
  w_{kj}^{(r+1)} = w_{kj}^{(r)} - \eta \delta_{kn} \phi_j(x_n)
  \]
  \[
  \delta_{kn} = \frac{\partial g(ak)}{\partial w_{kj}} \left( \sum_{i=0}^{M} w_{ki} \phi_i(x_n) \right)
  \]
Summary: Generalized Linear Discriminants

• Properties
  - General class of decision functions.
  - Nonlinearity $g(\cdot)$ and basis functions $\phi_i$ allow us to address
    linearly non-separable problems.
  - Shown simple sequential learning approach for parameter
    estimation using gradient descent.
  - Better 2nd order gradient descent approaches available
    (e.g. Newton-Raphson).

• Limitations / Caveats
  - Flexibility of model is limited by curse of dimensionality
    - $g(\cdot)$ and $\phi_i$ often introduce additional parameters.
  - Models are either limited to lower-dimensional input space
    or need to share parameters.
  - Linearly separable case often leads to overfitting.
    - Several possible parameter choices minimize training error.

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.

Classifying 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 $\|w\|_2$.
     ⇒ Need to enforce additional constraint $\|w\|_2 = 1$.
  2. This constrained minimization results in $w \propto (m_2 - m_1)$
  3. The criterion may result in bad separation if the clusters have
     elongated shapes.

Classifying as Dimensionality Reduction

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

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:
    $J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$
    with $s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$
  - Usually, this is written as
    $J(w) = \frac{w^T S_B w}{w^T S_W w}$
    where
    $S_B = (m_2 - m_1) (m_2 - m_1)^T$
    $S_W = \frac{1}{2} \sum_{k=1}^{n} \sum_{n \in C_k} (x_n - m_k) (x_n - m_k)^T$

Image source: C.M. Bishop, 2006
**Fisher’s Linear Discriminant Analysis (FLD)**

- Maximize distance between classes
- Minimize distance within a class
- Criterion: \( J(w) = w^T S_B w \)
  \[ w^T S_W w \]
- The optimal solution for \( w \) can be obtained as:
- Classification function:

**Multiple Discriminant Analysis**

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

**Maximizing J(W)**

- "Rayleigh quotient" \( \Rightarrow \) Generalized eigenvalue problem
  \[ J(W) = \frac{W^T S_B W}{W^T S_W W} \]
  - The columns of the optimal \( W \) are the eigenvectors corresponding to the largest eigenvalues of \( S_B W_i = \lambda_i S_W W_i \)
  - Defining \( v = S_B W \), we get \( \frac{1}{S_W} S_B v = \lambda v \) 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?
  - Classifier interpretation
  - The weight vector has the same dimensionality as \( x \).
  - Positive contributions where \( \text{sign}(x_i) = \text{sign}(w_i) \).
  - 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:
    \( C_1 \): Subjects with glasses
    \( C_2 \): Subjects without glasses
  - Test:
    \( \text{glasses?} \)
    - Take each image as a vector of pixel values and apply FLD.

**Fisherfaces: Interpretability**

- Resulting weight vector for “Glasses/NoGlasses”
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.

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

Proportional Discriminative Models

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

- 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
  \[ a = w^T x \]
  \[ \text{or} \]
  \[ a = w^T \phi(x) \]

Comparison

- Let’s look at the number of parameters...
  - Assume we have an \( M \)-dimensional feature space \( \phi \).
  - And assume we represent \( p(\phi|C_i) \) and \( p(C_i) \) by Gaussians.
  - How many parameters do we need?
    - For the means: \( 2M \)
    - For the covariances: \( M(M+1)/2 \)
    - Together with the class priors, this gives \( M(M+5)/2+1 \) parameters!

- How many parameters do we need for logistic regression?
  \[ p(C_1|\phi) = y(\phi) = \sigma(w^T \phi) \]
  - Just the values of \( w \) \( \Rightarrow \) \( M \) parameters.

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

Logistic Sigmoid

- Properties
  - Definition: \( \sigma(a) = \frac{1}{1 + \exp(-a)} \)
  - Inverse: \( a = \ln \left( \frac{\sigma}{1 - \sigma} \right) \) “logit” function

  - Symmetry property: \( \sigma(-a) = 1 - \sigma(a) \)
  - Derivative: \( \frac{d\sigma}{da} = \sigma(1 - \sigma) \)
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 \( y_n = p(C|\phi_n) \), we can write the likelihood as

\[
p(t|\phi) = \prod_{n=1}^N y_n^{t_n} (1 - y_n)^{1-t_n}
\]

Define the error function as the negative log-likelihood

\[
E(\phi) = -\ln p(t|\phi) = -\sum_{n=1}^N \{ t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \}
\]

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

---

Logistic Regression

- Let's apply Newton-Raphson for Logistic Regression

\[
\phi = \text{softmax}(\mathbf{W}^T \mathbf{X})
\]

Gradient of the Error Function

- Error function

\[
E(\mathbf{W}) = -\sum_{n=1}^N \{ t_n \ln y_n + (1 - t_n) \ln (1 - y_n) \}
\]

- Gradient

\[
\nabla E(\mathbf{W}) = -\sum_{n=1}^N \{ t_n \frac{\partial y_n}{\partial \mathbf{W}} + (1 - t_n) \frac{\partial y_n}{\partial \mathbf{W}} \}
\]

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

A More Efficient Iterative Method...

- Second-order Newton-Raphson gradient descent scheme

\[
\mathbf{W}^{(r+1)} = \mathbf{W}^{(r)} - \mathbf{H}^{-1} \nabla E(\mathbf{W})
\]

where \( \mathbf{H} = \nabla^2 E(\mathbf{W}) \) is the Hessian matrix, i.e. the matrix of second derivatives.

- Properties
  - Local quadratic approximation to the log-likelihood.
  - Faster convergence.

Newton-Raphson for Least-Squares Estimation

- Let's first apply Newton-Raphson to the least-squares error function:

\[
E(\mathbf{W}) = \frac{1}{2} \sum_{n=1}^N (\mathbf{W}^T \phi_n - t_n)^2
\]

\[
\nabla E(\mathbf{W}) = \sum_{n=1}^N (\mathbf{W}^T \phi_n - t_n) \phi_n = \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{t}
\]

\[
\mathbf{H} = \nabla^2 E(\mathbf{W}) = \sum_{n=1}^N \phi_n \phi_n^T = \mathbf{X}^T \mathbf{X}
\]

\[
\phi = \begin{bmatrix} \phi_1 & \cdots & \phi_N \end{bmatrix}
\]

Resulting update scheme:

\[
\mathbf{W}^{(r+1)} = \mathbf{W}^{(r)} - (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{W}^{(r)} - \mathbf{X}^T \mathbf{y})
\]

\[
= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}
\]

Closed-form solution!
Iteratively Reweighted Least Squares

- Update equations
  \[ w^{(r+1)} = w^{(r)} - (\Phi^T R \Phi)^{-1} \Phi^T (y - t) \]
  \[ = (\Phi^T R \Phi)^{-1} \left\{ \Phi^T R \Phi w^{(r)} - \Phi^T (y - t) \right\} \]
  \[ = (\Phi^T R \Phi)^{-1} \Phi^T R z \]
  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.
  \( \Rightarrow \) Iteratively Reweighted Least-Squares (IRLS)

Summary: Logistic Regression

- Properties
  - Directly represent posterior distribution \( y(\phi | C_k) \)
  - 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

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.

Note on Error Functions

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

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).