Discriminant functions

Victor Kitov

Yandex School of Data Analysis

Evaluation

- In machine learning objects, predicted classes, prediction functions,etc. can be assigned:
  - **score, rating** - this should be maximized
  - **loss, cost** - this should be minimized
- We can always transform score$\longleftrightarrow$loss, using:

$$
\begin{aligned}
loss(z) &= -score(z),, \quad ... \\
loss(z) &= \frac{1}{score(z)} \text{ for } score(z) > 0 \\
&\cdots
\end{aligned}
$$

Metrics

- Among each pair of objects x, x' we may define:
  - **distance** $\rho(x, x')$ or $\|x - x'\|$: how much they are different
  - **similarity** $sim(x, x')$: how much they are close to each other
- We can always transform distance$\longleftrightarrow$similarity, using:

$$
\begin{aligned}
\rho(x, x') &= 1 - sim(x, x'), \\
\rho(x, x') &= \frac{1}{sim(z)} \text{ for } sim(z) > 0 \\
&\cdots
\end{aligned}
$$

Definition

- Discriminant functions is the most general way to describe each classifier.
- Each classifier implies a particular set of discriminant functions.

Discriminant functions

- a set of C functions $g_y(x)$, $y = 1, 2 \ldots C$.
- $g_y(x)$ measures the score of class y, given object x.

Usage

Assign x to class having maximum discriminant function value:

$$\widehat{c} = \arg\max_c g_c(x)$$

Examples

- K-NN:

$$g_y(x) = \sum_{k=1}^{K} \mathbb{I}[y_{i(k)} = y]$$

- Linear classifier:

$$g_y(x) = \langle w_y, x \rangle$$

- Nearest centroid:

$$g_y(x) = \rho(x, \mu_y)$$

- Maximum posterior probability classifier:

$$g_y(x) = p(y|x)$$

- Minimum cost classifier:

$$g_y(x) = -\mathcal{L}(y) = -\sum_c p(\omega_c|x)\lambda_{cy}$$

Properties

Discriminant functions are not unique

$g_y(x)$ and $g'_y(x) = F(g(x))$ lead to equivalent classification for any monotonically increasing function $F(x)$.

Binary classification

- For two class case $y \in \{-1, +1\}$ we may define a single discriminant function $g(x) = g_1(x) - g_2(x)$ such that

$$\widehat{y}(x) = \begin{cases} +1, & g(x) \geq 0, \\ -1 & g(x) < 0. \end{cases}$$

Binary classification

- For two class case $y \in \{-1, +1\}$ we may define a single discriminant function $g(x) = g_1(x) - g_2(x)$ such that

$$\widehat{y}(x) = \begin{cases} +1, & g(x) \geq 0, \\ -1 & g(x) < 0. \end{cases}$$

- Boundary between classes: $\{x : g(x) = 0\}$.

Binary classification

- For two class case $y \in \{-1, +1\}$ we may define a single discriminant function $g(x) = g_1(x) - g_2(x)$ such that

$$\widehat{y}(x) = \begin{cases} +1, & g(x) \geq 0, \\ -1 & g(x) < 0. \end{cases}$$

- Boundary between classes: $\{x : g(x) = 0\}$.
- Linear classifier:
    - $g(x) = \langle w_{+1}, y\rangle - \langle w_{-1}, y\rangle = \langle w, y\rangle$
    - $\widehat{y}(x) = sign[g(x)]$

Binary classification: probability calibration

- $g(x)$ - score of positive class, $p(y=+1|x)$-?
- Platt scaling: $p(y=+1|x)=\sigma\left(\theta_0+\theta_1 g(x)\right)$,
  - $\sigma(u)=\frac{1}{1+e^{-u}}$

Binary classification: probability calibration

- Using the property $1-\sigma(z)=\sigma(-z)$:

$$p(y=1|x)=\sigma\left(\theta_0+\theta_1 g(x)\right)$$
$$p(y=-1|x)=1-\sigma(\theta_0+\theta_1 g(x))=\sigma\left(-\theta_0-\theta_1 g(x)\right)$$

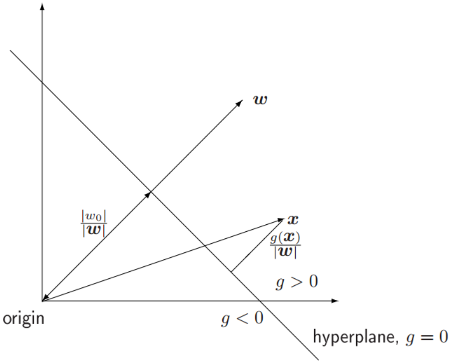
- Thus $p(y|x)=\sigma\left(y(\theta_0+\theta_1 g(x))\right)$
- Estimate θ_0,θ_1 using maximum likelihood:

$$\prod_{n=1}^{N}\sigma\left(y_n(\theta_0+\theta_1 g(x_n))\right)\to\max_{\theta_0,\theta_1}$$

Linear discriminant function

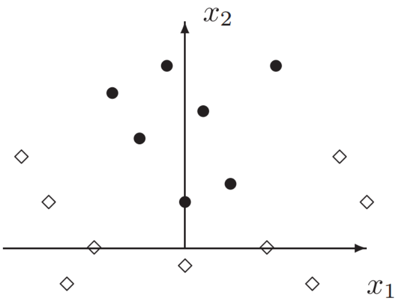
Simplest case - linear discriminant function:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$



Linear discrimination for non-linear cases

The objects below can't be separated with linear boundary.



However, objects may be linearly separated in transformed space: $\phi_1(\boldsymbol{x}) = x_1^2$, $\phi_2(\boldsymbol{x}) = x_2$.

Linear discrimination for non-linear cases

Natural way to make non-linear decision boundaries is to apply standard linear discriminant functions with transformed features.

Most well-known examples:

- linear: $\phi_i(\boldsymbol{x}) = x_i$
- polynomial: $\phi_i(\boldsymbol{x}) = x_{k_1}^{s_1} x_{k_2}^{s_2} ... x_{k_q}^{s_q}$
- radial basis functions: $\phi_i(\boldsymbol{x}) = \phi(\|\boldsymbol{x} - \boldsymbol{z_i}\|)$, where $\phi(\cdot)$ is non-increasing function, meaning proximity.
- multi-layer perceptron: $\phi_i(\boldsymbol{x}) = \sigma(\boldsymbol{x}^T \boldsymbol{w}_i + w_{i0})$, where $\sigma(u) = 1/(1 + e^{-u})$ - sigmoid step function.