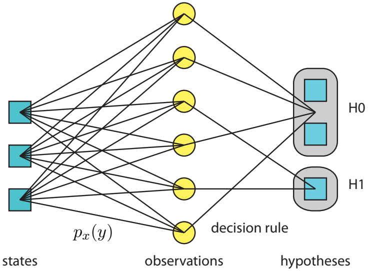
ECE531 Screencast 8.2: Decision Rules

D. Richard Brown III

Worcester Polytechnic Institute

Decision Rules: Main Ideas



- Decision rules: a mapping from observations to hypotheses.
- Deterministic decision rules partition the observation space into subsets $\mathcal{Y}_0, \dots, \mathcal{Y}_{M-1}$ such that

$$y \in \mathcal{Y}_i \quad \Rightarrow \quad \text{decide } \mathcal{H}_i$$

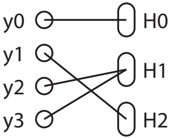
  with $\mathcal{Y}_i \subseteq \mathcal{Y}$, $\mathcal{Y}_i \bigcap \mathcal{Y}_j = \emptyset$ for $i \neq j$, and $\bigcup_i \mathcal{Y}_i = \mathcal{Y}$.
- There are cases where we may want to specify random decision rules.
- There are several ways of specifying decision rules.

Deterministic Decision Rules: Notation 1

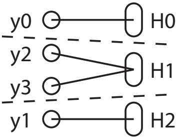
Deterministic decision rule $\delta : \mathcal{Y} \mapsto \mathcal{Z}$

$$\delta(y) \quad = \quad m \text{ means that we decide } \mathcal{H}_m \text{ when we observe } y$$

For example, if we want



or



then we write

$$\delta(y) = \begin{cases} 0 & y = y_0 \\ 1 & y = y_2 \text{ or } y = y_3 \\ 2 & y = y_1 \end{cases}$$

Remarks:

+ work for finite and infinite observations spaces
- not easily generalizable to random decision rules

Deterministic Decision Rules: Notation 2

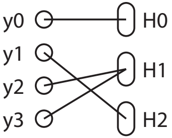
A more general way of specifying deterministic decision rules is $\delta : \mathcal{Y} \mapsto \mathbb{R}^M$ where

$$\delta(y) = [\delta_0(y), \ldots, \delta_{M-1}(y)]^\top$$

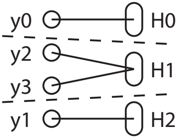
and

$$\text{and } \delta_i(y) = \begin{cases} 1 & \text{if we decide } \mathcal{H}_i \text{ when we observe } y \\ 0 & \text{if we don't decide } \mathcal{H}_i \text{ when we observe } y \end{cases}$$

for $i = 0, \ldots, M-1$. For example, if we want



or



then we write

$$\delta(y) = \begin{cases} [1,0,0]^\top & y = y_0 \\ [0,1,0]^\top & y = y_2 \text{ or } y = y_3 \\ [0,0,1]^\top & y = y_1 \end{cases}$$

Advantage: This notation easily extends to random decision rules.

Decision Matrices

When we have a finite number of possible observations, a convenient way to specify a decision rule is a **decision matrix** $D \in \mathbb{R}^{M \times L}$ with $D_{i\ell} = \delta_i(y_\ell)$. For example

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

This is equivalent to the previous decision rule.

Remarks:

+ easily generalizable to random decision rules, e.g.

$$D = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0.5 \\ 0 & 1 & 0 & 0.5 \end{bmatrix}.$$

+ convenient for generating conditional risk vectors in Matlab
- doesn't work for infinite observations spaces

Random Decision Rules: General Notation

We usually use the notation $\rho_i(y)$ to denote a random decision rule.

This means "when we get observation y, we decide hypothesis i with probability $\rho_i(y)$".

Clearly, for any given y, we must have

$$\sum_{i=0}^{M-1} \rho_i(y) = 1$$

This notation can easily be related to decision matrices, e.g.

$$D = \begin{bmatrix} 0.7 & 0.4 & 0.5 & 0 \\ 0.2 & 0.4 & 0.2 & 0.9 \\ 0.1 & 0.2 & 0.3 & 0.1 \end{bmatrix} \Leftrightarrow \rho_0(y_0) = 0.7,\ \rho_1(y_0) = 0.2, \ldots$$

but, unlike decision matrices, is not limited to finite observation spaces.

This is probably the most general way of specifying decision rules, but it can be notationally cumbersome.

Random Decision Rules: Binary HT Notation

In **binary** hypothesis testing problems, there are only two possible decisions: $\mathcal{H}_0$ and $\mathcal{H}_1$. It is convenient in this case to use the more compact deterministic decision rule notation:

$$\delta(y) = \begin{cases} 1 & \text{if we decide } \mathcal{H}_1 \text{ when we observe } y \\ 0 & \text{if we decide } \mathcal{H}_0 \text{ when we observe } y \end{cases}$$

Since there are only two possibilities, randomized decision rules can be written as

$$\rho(y) = \begin{cases} 1 & \text{if we always decide } \mathcal{H}_1 \text{ when we observe } y \\ \gamma & \text{if we decide } \mathcal{H}_1 \text{ with probability } \gamma \text{ when we observe } y \\ 0 & \text{if we always decide } \mathcal{H}_0 \text{ when we observe } y \end{cases}$$

Advantages and limitations:

+ works for random decision rules
+ work for infinite observations spaces
+ not cumbersome
- only applicable to binary hypothesis testing problems