Bayes Decision Theory

How to make decisions in the presence of uncertainty?

History: 2nd World War
Radar for detection aircraft.
Codebreaking: Decryption.

Observed data $x \in X$
State $y \in Y$. Likelihood function

$p(x \mid y)$ — conditional distribution

Example: $y \in \{-1, 1\}$
Salmon / Sea Bass
Airplane / Bird

$p(x \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{1}{2}(x-\mu_y)^2} \quad \text{mean } \mu_y, \quad \text{variance } \sigma_y^2.$

$x_0 \sim \text{lognormal}$. $p(x) \to 0$
Maximum Likelihood (ML)

\[ y_{ML} = \arg \max_y p(x | y) \]

\[ \frac{p(x | y = 1)}{p(x | y = -1)} > 1 \quad \text{decide } y = 1 \\
\text{otherwise } y = -1 \]

Equivalently

\[ \log \frac{p(x | y = 1)}{p(x | y = -1)} > 0 \quad \text{log-likelihood test.} \]

Seems reasonable, but what if birds are more likely than airplanes?

Must take into account the prior probability \( p(y = 1), p(y = -1) \).

Bayes Rule

\[ p(y | x) = \frac{p(x | y) p(y)}{p(x)} \]

Probability of \( y \) conditioned on observation.

\[ \frac{p(y = 1 | x)}{p(y = -1 | x)} > 1 \quad \text{decide } y = 1 \\
\text{otherwise } \quad \text{decide } y = -1 \]

Maximum a Posteriori (MAP)

\[ y_{MAP} = \arg \max_y p(y | x) \]
Another ingredient

what does it cost if you make
a mistake?

i.e. suppose you decide \( y = 1 \), but really \( y = -1 \).
i.e. you may pay a big penalty if you
decide it is a bird when it is a plane.
(Pascal’s Wager: Bet on God)

Putting everything together.

likelihood function \( p(x | y) \) \( x \in X, y \in Y \)

prior \( p(y) \)

decision rule \( \alpha(x) \) \( \alpha(x) \in Y \)

loss function \( L(\alpha(x), y) \) cost of making
decision \( \alpha(x) \) when
true state is \( y \).

\[ L(0, y) = 0, \quad \text{if} \quad \alpha(x) = y \]
\[ L(1, y) = 1, \quad \text{if} \quad \alpha(x) \neq y \]

All wrong answers penalized the same.
(4) **Risk**

The risk of the decision rule \( d(x) \) is the expected loss.

\[
R(d) = \sum_{x,y} L(d(x), y) P(x, y)
\]

(Note integrate \( \int_a^b \) if \( x \) is continuous)

Bayes Decision Theory says

"pick the decision rule \( \hat{d} \) which minimizes the risk",

\[
\hat{d} = \arg\min_{d \in A} R(d), \quad R(\hat{d}) \geq R(d) \quad \forall d \in A.
\]

\( \hat{d} \) is Bayes Decision

\( R(\hat{d}) \) is Bayes Risk.
Bayes Risk

Bayes Risk is the best you can do if:

(a) you know \( p(x|y)p(y) \) \& \( L(\cdot, \cdot) \)

(b) you can compute \( E = \sum_{x,y} R(x,y) \)

(c) you can afford the losses (e.g. gambling, poker)

(d) you make the decision for a sequence of data \( x_1, \ldots, x_n \) with states \( y_1 \ldots y_n \) where each \( (x_i, y_i) \) are independently identically distributed from \( p(x, y) \)

Bad - if you are playing a game against an intelligent opponent (Game Theory)

- if any of the assumptions (a), (b), (c), (d) are wrong.
(6) Better understanding of Bayes Decision Theory. Re-express
\[ R(\alpha) = \sum_x \sum_y L(\alpha(x), y) P(x, y) \]
\[ = \sum_x P(x) \left( \sum_y L(\alpha(x), y) P(y|x) \right) \]
Hence, for each \( x \),
\[ \alpha(x) = \text{arg min}_\alpha \sum_y L(\alpha(x), y) P(y|x) \]

Obtaining MAP & ML as special cases.

If \( y \in \{-1, 1\} \) and the loss function penalizes all errors equally:
\[ L(\alpha(x), y) = 1, \text{ if } \alpha(x) \neq y \]
\[ = 0, \text{ otherwise} \]
\[ y \in \{-1, 1\} \]
Then \[ \alpha(x) = \text{arg max}_\alpha P(y = \alpha(x) | x) \]
\text{MAP estimate.}

If also \( P(y=1) = P(y=-1) \), then
\[ \widehat{\alpha}(x) = \text{arg max}_\alpha \{P(x | y = \alpha(x))\} \text{ ML estimate.} \]
(7) **Examples**

\[ p(x|y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-My)^2}{2\sigma^2}}. \]

\( y \in (-1,1) \quad p(y) = \frac{1}{2}. \)

\[ L(a(x), y) = 1, \text{ if } a(x) \neq y, = 0 \text{ otherwise.} \]

**Bayes Rule**

\[ a(x) = \arg \min_{t \in \{-1,1\}} |x-My| \]

Suppose \( x \) is a vector in two dimensions.

\[ p(x|y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-My)^2}{2\sigma^2}} \]

Gaussian with unequal covariance

Decision Boundary

\[ y = \frac{1}{2} (x-My)^T \Sigma_y^{-1} (x-My) \]
Bayes Decision theory also applies when $y$ is not a binary variable – e.g. $y$ can take $M$ values or $y$ continuous valued.

In this course, usually $y \in \{-1, 1\}$ classifier,

or $y \in \{1, 2, \ldots, M\}$ multi-class classifier

or $y \in (-\infty, +\infty)$ regression.

Bayes decision theory is ideal – but in practice it can be difficult to apply because of the limitations described earlier.
In particular, we usually do not know the distributions \( p(x | y)p(y) \). Instead we have a set of examples \( \{(x_1, y_1), \ldots, (x_N, y_N)\} \).

We can define the empirical risk:

\[
R_{\text{emp}}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} L(\alpha(x_i), y_i).
\]

We assume that the observed data are i.i.d. samples from the (unknown) distribution \( p(x, y) = p(x | y)p(y) \).

Then, as \( N \to \infty \),

\[
R_{\text{emp}}(\alpha) \to R(\alpha) \text{ in } L^2.
\]
(10) This suggests two (extreme) strategies.

**First Strategy:** (Generative)

Use \( \{(x_i, y_i): i = 1 \text{ to } N\} \) to learn the distribution \( p(x|y)p(y) \), then apply Bayes Decision Theory.

\[ \hat{\omega}(x) = \arg\max_{\omega} \sum_y I(y|\omega(x), y) p(y) \]

**Second Strategy:** (Discriminative)

Attempt to estimate \( \hat{\omega}(x) \) directly from the empirical risk \( R_{emp}(\omega) \).

Second strategy is the start of machine learning.

Why estimate the probabilities when you only care about the decision?
Memorization vs Generalization.

(a) Finiteness of Data

Suppose we have $R$ samples with $N$ samples.
You want to learn a rule $h(x)$ that will give good results for data you haven't seen yet.

Assume $\{(x_i, y_i) : i = 1 \ldots N\}$ are samples from unknown distribution $P(x, y)$.

You want to learn a decision rule from $\{(x_i, y_i) : i = 1 \ldots N\}$ that will also apply to other samples from $P(x, y)$.

Vapnik's Theory —

Don't want a rule that works perfectly on $\{(x_i, y_i) : i = 1 \ldots N\}$ but fails to generalize to new (unseen) samples.
\[ (12) \quad \text{Memorization:} \]

Decision Rule: \[ \hat{x} = \text{argmax}_x \text{Rempl}(x) \]

Rempl(\( \hat{x} \)) small, but \( R(x) \) big.

ie. bad for predicting new data.

\[ \text{Generalization:} \]

Want a decision rule \( \hat{x} \) so that

\( \text{Rempl}(\hat{x}) \) is small, but \( R(x) \) is small.

In practice - cross-validation.

Training set \( \{ (x_i, y_i) : i = 1 \text{ to } N \} \) to learn the rule \( \hat{x} \)

Test set \( \{ (x_j, y_j) : j = 1 \text{ to } M \} \) to test the rule \( \hat{x} \).

Choose \( \hat{x} \) so that \( \text{Rempl}(\hat{x}) \) is small on both the training set and test set.

Hence, restrict the possibility of \( \hat{x} \).