Bayes Decision Theory - I

Nuno Vasconcelos
(Ken Kreutz-Delgado)

UCSD
Statistical Learning from Data

• **Goal:** Given a relationship between a feature vector $x$ and a vector $y$, and iid data samples $(x_i,y_i)$, find an approximating function $f(x) \approx y$

  $x \xrightarrow{f(\cdot)} \hat{y} = f(x) \approx y$

• This is called training or learning.

• Two major types of learning:
  – Unsupervised (aka Clustering): only $X$ is known.
  – Supervised (Classification or Regression): both $X$ and target value $Y$ are known during training, only $X$ is known at test time.
Nearest Neighbor Classifier

• The simplest possible classifier that one could think of:
  
  – It consists of assigning to a new, unclassified vector the same class label as that of the closest vector in the labeled training set

  – E.g. to classify the unlabeled point “Red”:
    
    ▪ measure Red’s distance to all other labeled training points
    ▪ If the closest point to Red is labeled “A = square”, assign it to the class A
    ▪ otherwise assign Red to the “B = circle” class

• This works a lot better than what one might expect, particularly if there are a lot of labeled training points
Nearest Neighbor Classifier

• To define this classification procedure rigorously, define:
  – a Training Set \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \)
  – \( x_i \) is a vector of observations, \( y_i \) is the class label
  – a new vector \( x \) to classify

• The Decision Rule is

\[
\text{argmin}_i \quad \text{distance} \\
\text{set} \quad y = y_{i^*} \\
\text{where} \\
i^* = \text{arg min} \quad d(x, x_i) \\
i \in \{1, \ldots, n\}
\]

– \text{argmin} means: “the \( i \) that minimizes the distance”
Metrics

• we have seen some examples:
  – \( \mathbb{R}^d \)
    
    \[ \begin{align*}
    \langle x, y \rangle &= x^T y = \sum_{i=1}^{d} x_i y_i \\
    \|x\| &= \sqrt{x^T x} = \sqrt{\sum_{i=1}^{d} x_i^2} \\
    d(x, y) &= \|x - y\| = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}
    \end{align*} \]
  
  -- Continuous functions
    
    \[ \begin{align*}
    \langle f(x), g(x) \rangle &= \int f(x) g(x) dx \\
    \|f(x)\| &= \sqrt{\int f^2(x) dx} \\
    d(f, g) &= \sqrt{\int [f(x) - g(x)]^2 dx}
    \end{align*} \]
Euclidean distance

• We considered in detail the Euclidean distance

\[ d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \]

• Equidistant points to \( x \)?

\[ d(x, y) = r \iff \sum_{i=1}^{d} (x_i - y_i)^2 = r^2 \]

  – E.g. \((x_1 - y_1)^2 + (x_2 - y_2)^2 = r^2\)

• The equidistant points to \( x \) are on spheres around \( x \)
• Why would we need any other metric?
Inner Products

- **fish example:**
  - **features** are $L =$ fish length, $W =$ scale width
  - **measure** $L$ in meters and $W$ in millimeters
    - **typical $L$:** 0.70m for salmon, 0.40m for sea-bass
    - **typical $W$:** 35mm for salmon, 40mm for sea-bass
  - I have three fish
    - $F_1 = (.7, 35)$  $F_2 = (.4, 40)$  $F_3 = (.75, 37.8)$
    - $F_1$ clearly salmon, $F_2$ clearly sea-bass, $F_3$ looks like salmon
    - yet
      $$d(F_1, F_3) = 2.8 > d(F_2, F_3) = 2.23$$
    - there seems to be something wrong here
  - but if **scale width** is *also* measured in meters:
    - $F_1 = (.7, .035)$  $F_2 = (.4, .040)$  $F_3 = (.75, .0378)$
    - and now
      $$d(F_1, F_3) = .05 < d(F_2, F_3) = .35$$
    - which seems to be right – the units are **commensurate**
Inner Product

• Suppose the scale width is also measured in meters:
  – I have three fish
    ▪ $F_1 = (.7, .035)$  $F_2 = (.4, .040)$  $F_3 = (.75, .0378)$
    ▪ and now
      $$d(F_1, F_3) = .05 < d(F_2, F_3) = 0.35$$
  – which seems to be right

• The problem is that the Euclidean distance depends on the units (or scaling) of each axis
  – e.g. if I multiply the second coordinate by 1,000

$$d'(x, y) = \sqrt{(x_1 - y_1)^2 + 1,000,000(x_2 - y_2)^2}$$

The 2nd coordinates influence on the distance increases 1,000-fold!

• Often “right” units are not clear (e.g. car speed vs weight)
Inner Products

• We need to work with the “right”, or at least “better”, units
• Apply a transformation to get a “better” feature space

\[ x' = Ax \]

• examples:
  – Taking \( A = R \), \( R \) proper and orthogonal, is equivalent to a rotation
  – Another important special case is scaling \( (A = S, \text{ for } S \text{ diagonal}) \)
  \[
  \begin{bmatrix}
  \lambda_1 & 0 & 0 \\
  0 & \ddots & 0 \\
  0 & 0 & \lambda_n
  \end{bmatrix}
  \begin{bmatrix}
  x_1 \\
  \vdots \\
  x_n
  \end{bmatrix}
  =
  \begin{bmatrix}
  \lambda_1 x_1 \\
  \vdots \\
  \lambda_n x_n
  \end{bmatrix}
  
  – We can combine these two transformations by making taking \( A = SR \)
(Weighted) Inner Products

• Thus, in general one can rotate and scale by applying some matrix $A = SR$, to form transformed vectors $x' = Ax$

• What is the inner product in the new space?

$$ (x')^T y' = (Ax)^T Ay = x^T A^T A y $$

• The inner product in the new space is of weighted form in the old space

$$ \langle x', y' \rangle = x^T My $$

• Using a weighted inner product, is equivalent to working in the transformed space
(Weighted) Inner Products

• Can I use any weighting matrix M? – NO!

• **Recall:** an inner product is a bilinear form such that

  i) $\langle x, x \rangle \geq 0$, $\forall x \in \mathcal{H}$

  ii) $\langle x, x \rangle = 0$ if and only if $x = 0$

  iii) $\langle x, y \rangle = \langle y, x \rangle$ for all $x$ and $y$

• From iii), M must be Symmetric since

  $$\langle x, y \rangle = x^T M y = \left( y^T M^T x \right)^T = y^T M^T x \quad \text{and}$$

  $$\langle x, y \rangle = \langle y, x \rangle = y^T M x, \quad \forall x, y$$

• from i) and ii), M must be Positive Definite

  $$\langle x, x \rangle = x^T M x > 0, \quad \forall x \neq 0$$
Positive Definite Matrices

• **Fact:** Each of the following is a necessary and sufficient condition for a real symmetric matrix $A$ to be positive definite:
  i) $x^T A x > 0$, $\forall x \neq 0$
  ii) All eigenvalues, $\lambda_i$, of $A$ are real and satisfy $\lambda_i > 0$
  iii) All upper-left submatrices $A_k$ have strictly positive determinant
  iv) There is a matrix $R$ with independent columns such that $A = R^T R$

• **Note:** from property iv), we see that using a positive definite matrix $A$ to weight an inner product is the same as working in a transformed space.

• **Definition of upper left submatrices:**

\[
A_1 = a_{1,1} \\
A_2 = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \\
A_3 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \\
\vdots
\]
Metrics

• What is a good weighting matrix $M$?
  – Let the data tell us!
  – Use the inverse of the covariance matrix $M = \Sigma^{-1}$

\[
\Sigma = E[(x - \mu)(x - \mu)^T] \\
\mu = E[x]
\]

• Mahalanobis Distance:

\[
d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}
\]

• This distance is adapted to the covariance (“scatter”) of the data and thereby provides a “natural” rotation and scaling for the data
The Multivariate Gaussian

• In fact, for Gaussian data, the Mahalanobis distance tells us all we could statistically know about the data
  – The pdf for a \( d \)-dimensional Gaussian of mean \( \mu \) and covariance \( \Sigma \) is

\[
P_X(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}
\]

  – Note that this can be written as

\[
P_X(x) = \frac{1}{K} \exp \left\{ -\frac{1}{2} d^2 (x, \mu) \right\}
\]

  – I.e. a Gaussian is just the exponential of the negative of the square of the Mahalanobis distance
  – The constant \( K \) is needed only to ensure the density integrates to 1
The Multivariate Gaussian

- Using Mahalanobis = assuming Gaussian data
- Mahalanobis distance:

\[ d^2(x, \mu) = (x - \mu)^T \Sigma^{-1} (x - \mu) \]

Gaussian pdf:

\[ P_x(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]

- Points of high probability are those of small Mahalanobis distance to the center (mean) of a Gaussian density
- This can be interpreted as the right norm for a certain type of space
The Multivariate Gaussian

• Defined by two parameters
  – Mean just shifts center
  – Covariance controls shape
  – in 2D, \( X = (X_1, X_2)^T \)
  \[
  \Sigma = \begin{bmatrix}
  \sigma_1^2 & \sigma_{12} \\
  \sigma_{12} & \sigma_2^2
  \end{bmatrix}
  \]
  – \( \sigma_i^2 \) is variance of \( X_i \)
  – \( \sigma_{12} = \text{cov}(X_1, X_2) \) controls how dependent \( X_1 \) and \( X_2 \) are
  – Note that, when \( \sigma_{12} = 0 \):

\[
P_{X_1,X_2}(x_1, x_2) = P_{X_1}(x_1)P_{X_2}(x_2) \iff X_i \text{ are independent}
\]
The multivariate Gaussian

• this applet allows you to view the impact of the covariance parameters


• note: they show

\[ \rho = \frac{\sigma_{12}}{\sigma_1 \sigma_2} \]

• but since you do not change \( \sigma_1 \) and \( \sigma_2 \) when you are changing \( \rho \), this has the same impact as changing \( \sigma_{12} \)
“Optimal” Classifiers

- Some metrics are “better” than others
- The meaning of “better” is connected to how well adapted the metric is to the properties of the data
- Can we be more rigorous? Can we have an “optimal” metric? What could we mean by “optimal”?
- To talk about optimality we start by defining cost or loss

\[\hat{y} = f(x)\]

- Cost is a real-valued loss function that we want to minimize
- It depends on the true \(y\) and the prediction \(\hat{y}\)
- The value of the cost tells us how good our predictor \(\hat{y}\) is
Loss Functions for Classification

- Classification Problem: loss is function of classification errors
  - What types of errors can we have?
  - Two Types: False Positives and False Negatives
    - Consider a face detection problem
    - If you see these two images and say
      - you have a false-positive
        - false-negative (miss)
  - Obviously, we have similar sub-classes for non-errors
    - true-positives and true-negatives
  - The positive/negative part reflects what we say
  - The true/false part reflects the real classes
Loss functions

- are some errors more important than others?
  - depends on the problem
  - consider a snake looking for lunch
  - the snake likes frogs
  - but dart frogs are highly poisonous
  - the snake must classify each frog it sees
    \[ Y = \{ \text{"dart", "regular"} \} \]
  - the losses are clearly different

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>Frog = dart</th>
<th>Frog = regular</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>(\infty)</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>
Loss functions

• but not all snakes are the same
  – this one is a dart frog predator
  – it can still classify each frog it sees
    \[ Y = \{ \text{“dart”, “regular”} \} \]
  – it actually prefers dart frogs
  – but the other ones are good to eat too

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>Frog = dart</th>
<th>Frog = regular</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>
(Conditional) Risk as Average Cost

• Given a loss function, denote the cost of classifying a data vector \( x \) generated from class \( j \) as \( i \) by

\[
L[j \rightarrow i]
\]

• Conditioned on an observed data vector \( x \), to measure how good the classifier is, \textit{on average}, use the (conditional) expected value of the loss, aka the (conditional) \textit{Risk},

\[
R(x, i) \overset{\text{def}}{=} \mathbb{E}\{L[Y \rightarrow i] \mid x\} = \sum_j L[j \rightarrow i] P_{Y \mid X}(j \mid x)
\]

• This means that the \textit{risk of classifying} \( x \) \textit{as} \( i \) \textit{is equal to}
  – the sum, over all classes \( j \), of the cost of classifying \( x \) as \( i \) when the truth is \( j \) times the \textit{conditional} probability that the true class is \( j \) (where the conditioning is on the observed value of \( x \))
(Conditional) Risk

• Note that:
  – This immediately allows us to define an optimal classifier as the one that minimizes the (conditional) risk
  – For a given observation \( x \), the Optimal Decision is given by

\[
i^*(x) = \arg \min_i R(x, i)
= \arg \min_i \sum_j L[j \rightarrow i] P_{Y|X}(j \mid x)
\]

and it has optimal (minimal) risk given by

\[
R^*(x) = \min_i R(x, i) = \min_i \sum_j L[j \rightarrow i] P_{Y|X}(j \mid x)
\]
(Conditional) Risk

- Back to our example
  - A snake sees this

and makes *probability assessments*

\[ P_{Y|X}(j \mid x) = \begin{cases} 
0 & j = \text{dart} \\
1 & j = \text{regular} 
\end{cases} \]

and computes the *optimal decision*
(Conditional) Risk

• **Info** an *ordinary snake* is presumed to have

Class probabilities *conditioned on* \( x \)

\[
P_{Y|X}(j \mid x) = \begin{cases} 0 & j = \text{dart} \\ 1 & j = \text{regular} \end{cases}
\]

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>dart</th>
<th>regular</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>∞</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

• **The risk of saying “regular” given the observation** \( x \) **is**

\[
\sum_j L[j \to \text{reg}] P_{Y|X}(j \mid x) =
\]

\[
= L[\text{reg} \to \text{reg}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \to \text{reg}] P_{Y|X}(\text{dart} \mid x)
\]

\[
= 0 \times 1 + \infty \times 0 = 0 + 0 = 0
\]
(Conditional) Risk

- Info the ordinary snake has **for the given observation** \( x \)

\[
P_{Y|X}(j | x) = \begin{cases} 0 & j = \text{dart} \\ 1 & j = \text{regular} \end{cases}
\]

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>dart frog</th>
<th>regular frog</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>( \infty )</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

- Risk of saying “dart” **given** \( x \) is

\[
\sum_{j} L[j \rightarrow \text{dart}] P_{Y|X}(j | x) =
\]

\[
= L[\text{reg} \rightarrow \text{dart}] P_{Y|X}(\text{reg} | x) + L[\text{dart} \rightarrow \text{dart}] P_{Y|X}(\text{dart} | x)
\]

\[
= 10 \times 1 + 0 \times 0 = 10 + 0 = 10
\]

- Optimal decision = say “regular”. Snake says “regular” **given** the observation \( x \) and has a good, safe lunch ☺️ (risk = 0)
(Conditional) Risk

- The next time the ordinary snake goes foraging for food
  - It sees this image \( x \)
  - It “knows” that dart frogs can be colorful
  - So it assigns a nonzero probability to this image \( x \) showing a dart frog

\[
P_{Y|X}(j | x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular} 
\end{cases}
\]
(Conditional) Risk

• Info the ordinary snake has given the new measurement $x$

Class probabilities *conditioned on new* $x$

\[
P_{Y|X}(j \mid x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular} 
\end{cases}
\]

Ordinary Snake Losses

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>dart frog</th>
<th>regular frog</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>$\infty$</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

• The risk of saying “regular” given the new observation $x$ is

\[
\sum_{j} L[j \rightarrow \text{reg}] P_{Y|X}(j \mid x) = \\
\begin{align*}
&= L[\text{reg} \rightarrow \text{reg}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \rightarrow \text{reg}] P_{Y|X}(\text{dart} \mid x) \\
&= 0 \times 0.9 + \infty \times 0.1 = \infty
\end{align*}
\]
(Conditional) Risk

• Info the snake has given $x$

$$P_{Y|X}(j \mid x) = \begin{cases} 0.1 & j = \text{dart} \\ 0.9 & j = \text{regular} \end{cases}$$

• Risk of saying “dart” given $x$ is

$$\sum_j L[j \rightarrow \text{dart}] P_{Y|X}(j \mid x) =$$

$$= L[\text{reg} \rightarrow \text{dart}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \rightarrow \text{dart}] P_{Y|X}(\text{dart} \mid x)$$

$$= 10 \times 0.9 + 0 \times 0.1 = 9$$

• The snake decides “dart” and looks for another frog
  – even though this is a regular frog with 0.9 probability

• Note that this is always the case unless $P_{Y|X}(\text{dart} \mid X) = 0$
(Conditional) Risk

• What about the “dart-snake” that can safely eat dart frogs?
  – The dart-snake sees this

\[
P_{Y|x}(j | x) = \begin{cases} 
0 & j = \text{dart} \\
1 & j = \text{regular} 
\end{cases}
\]

and makes probability assessments and computes the optimal decision.
(Conditional) Risk

- Info the dart-snake has given $x$

\[ P_{Y|X}(j | x) = \begin{cases} 0 & j = \text{dart} \\ 1 & j = \text{regular} \end{cases} \]

- Risk of saying "regular" given $x$ is

\[
\sum_j L[j \rightarrow \text{reg}] P_{Y|X}(j | x) =
\]

\[
= L[\text{reg} \rightarrow \text{reg}] P_{Y|X}(\text{reg} | x) + L[\text{dart} \rightarrow \text{reg}] P_{Y|X}(\text{dart} | x)
\]

\[
= 0 \times 1 + 10 \times 0 = 0
\]
(Conditional) Risk

- Info the dart-snake has *given* $x$

$$P_{Y|X}(j \mid x) = \begin{cases} 
0 & j = \text{dart} \\
1 & j = \text{regular} 
\end{cases}$$

- Risk of dart-snake deciding "dart" *given* $x$ is

$$\sum_j L[j \rightarrow \text{dart}] P_{Y|X}(j \mid x) =$$

$$= L[\text{reg} \rightarrow \text{dart}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \rightarrow \text{dart}] P_{Y|X}(\text{dart} \mid x)$$

$$= 10 \times 1 + 0 \times 0 = 10$$

- Dart-snake optimally decides "regular", which is consistent with the $x$-conditional class probabilities
(Conditional) Risk

• Now the dart-snake sees this

\[
P_{Y|X}(j \mid x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular} 
\end{cases}
\]

– Let’s assume that it makes the same probability assignments as the ordinary snake.
(Conditional) Risk

• Info dart-snake has *given new x*

\[
P_{Y|X}(j \mid x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular} 
\end{cases}
\]

Dart-Snake Losses

<table>
<thead>
<tr>
<th>Snake Prediction</th>
<th>Dart Frog</th>
<th>Regular Frog</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;regular&quot;</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>&quot;dart&quot;</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

• Risk of deciding “regular” *given new observation x is*

\[
\sum_j L[j \rightarrow \text{reg}] P_{Y|X}(j \mid x) =
\]

\[
= L[\text{reg} \rightarrow \text{reg}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \rightarrow \text{reg}] P_{Y|X}(\text{dart} \mid x)
\]

\[
= 0 \times 0.9 + 10 \times 0.1 = 1
\]
(Conditional) Risk

• Info dart-snake has *given new x*

\[ P_{Y|X}(j \mid x) = \begin{cases} 0.1 & j = \text{dart} \\ 0.9 & j = \text{regular} \end{cases} \]

• Risk of deciding “dart” *given x* is

\[
\sum_j L[j \rightarrow \text{dart}] P_{Y|X}(j \mid x) = \\
= L[\text{reg} \rightarrow \text{dart}] P_{Y|X}(\text{reg} \mid x) + L[\text{dart} \rightarrow \text{dart}] P_{Y|X}(\text{dart} \mid x) \\
= 10 \times 0.9 + 0 \times 0.1 = 9
\]

• The dart-snake optimally decides “regular” *given x*

• Once again, this is consistent with the probabilities

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>dart frog</th>
<th>regular frog</th>
</tr>
</thead>
<tbody>
<tr>
<td>regular</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>dart</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Dart-Snake Losses
(Conditional) Risk

• In summary, if both snakes have

\[
P_{Y|X}(j \mid x) = \begin{cases} 
0 & j = \text{dart} \\
1 & j = \text{regular} 
\end{cases}
\]

then both say “regular”

• However, if

\[
P_{Y|X}(j \mid x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular} 
\end{cases}
\]

  – the vulnerable snake decides “dart”
  – the predator snake decides “regular”

• The infinite loss for saying regular when frog is dart, makes the vulnerable snake much more cautious!
(Conditional) Risk, Loss, & Probability

• Note that the only factors involved in the Risk

\[ R(x, i) = \sum_j L[j \rightarrow i] P_{Y|X}(j \mid x) \]

are
– the Loss Function \[ L[i \rightarrow j] \]
– and the Measurement-Conditional Probabilities \[ P_{Y|X}(j \mid x) \]

• The risk is the expected loss of the decision ("on average, you will loose this much!")
• The risk is not necessarily zero!
(Conditional) Risk, Loss, & Probability

• The best that the “vulnerable” ordinary snake can do when

\[
P_{Y|x}(j | x) = \begin{cases} 
0.1 & j = \text{dart} \\
0.9 & j = \text{regular}
\end{cases}
\]

is to *always decide “dart”* and accept the loss of 9

• Clearly, because starvation will lead to death, a more realistic loss function for an ordinary snake would have to:
  – Account for how hungry the snake is. (If the snake is starving, it will have to be more risk preferring.)
  – Assign a finite cost to the choice of “regular” when the frog is a dart. (Maybe dart frogs will only make the snake super sick sometimes.)

• In general, the loss function is not “learned”
  – You know how much mistakes will cost you, or assess that in some way
  – What if I can’t do that? -- one reasonable default is the 0/1 loss function
0/1 Loss Function

- This is the case where we assign
  - i) zero loss for no error \textit{and} ii) equal loss for the two error types

\[
L[i \rightarrow j] = \begin{cases} 
0 & i = j \\
1 & i \neq j
\end{cases}
\]

<table>
<thead>
<tr>
<th>snake prediction</th>
<th>dart frog</th>
<th>regular frog</th>
</tr>
</thead>
<tbody>
<tr>
<td>“regular”</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>“dart”</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Under the 0/1 loss:

\[
i^*(x) = \arg \min_i \sum_j L[j \rightarrow i] P_{Y|X}(j \mid x)
\]

\[
= \arg \min_i \sum_{j \neq i} P_{Y|X}(j \mid x)
\]
0/1 Loss Function

• Equivalently:

\[
i^*(x) = \arg \min_i \sum_{j \neq i} P_{Y|X}(j \mid x) = \arg \min_i [1 - P_{Y|X}(i \mid x)] = \arg \max_i P_{Y|X}(i \mid x)
\]

• Thus the Optimal Decision Rule is
  – Pick the class that has largest posterior probability given the observation \( x \). (I.e., pick the most probable class)

• This is the Bayes Decision Rule (BDR) for the 0/1 loss
  – We will simplify our discussion by assuming this loss, but you should always be aware that other losses may be used
0/1 Loss Function

- The risk of this optimal decision is

\[
R(x, i^*(x)) = \sum_j L \left( j \rightarrow i^*(x) \right) P_{Y|X}(j \mid x)
\]

\[
= \sum_{j \neq i^*(x)} P_{Y|X}(j \mid x)
\]

\[
= 1 - P_{Y|X}(i^*(x) \mid x)
\]

- This is the probability that \( Y \) is different from \( i^*(x) \) given \( x \), which is the \( x \)-conditional probability that the optimal decision is wrong.

- The expected Optimal Risk \( R = E_X[ R(x, i^*(x)) ] \) is the probability of error of the optimal decision.
Any questions?