CHAPTER 2:  
Supervised Learning

Class web page:
https://courses.soe.ucsc.edu/courses/cmpe242/Fall11/01
Concept Learning (not in Bishop) from labeled Examples

- “Things” represented by a feature vector $x$ and a label or target $t$ (also called $y$ or $r$), often $t$ in $\{1,0\}$ or $\{+,-\}$
- Domain (or instance space) is set of all possible feature vectors
- A Hypothesis (sometimes called a Concept) partitions the domain into + and - regions
  - Or is a function labelling examples + or -
  - Or is just the + region
Assumption: iid Examples

- Distribution of things and measurements defines some unknown (but fixed) $p(x)$ over domain
- Target concept $C$ gives the “correct” labels, $C(x)$, as a function of the features, $x$
- Find a hypothesis $h$ that is “close” to $C$
  - A loss function $l(t, t')$ measures error of predictions, usually $l(t, t')=0$ if $t=t'$ and $l(t,t')=1$ otherwise
  - Want to minimize $\int P(x) l(C(x), h(x))$ -- probability of error for 0-1 loss
Tasty Coffee example

- Objects are cups of coffee
- Measure strength and sugar
- Each measurement is a feature or attribute
- Other features? (cream, temperature, roast)
- Features numeric (precision? Accuracy?)
- Label (or class) is “+” (tasty) or “-” (not)
- Example is \((x, t)\) pair, \(x\) in \(\mathbb{R}^2\), \(t\) in \{+, -\}
Terminology Review

- **Domain**: set of all possible $\mathbf{x}$ vectors
- **Concept**: a boolean function on domain, a mapping from $\mathbf{x}$'s to “1” and “0”; or “T” and “F”; or “+” and “-”; or a subset of domain
- **Target**: the concept to be learned
- **Hypothesis class/space**: is the set of hypotheses (concepts) that can be output by a given learning algorithm
Strength and sugar measured 0 to 10

Domain has 121 different instances

How to predict from these examples?

<table>
<thead>
<tr>
<th>strength</th>
<th>sugar</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>?</td>
</tr>
</tbody>
</table>
- **Version space**: all concepts in hypotheses space **consistent** with training set.
- If hypoth. Space is all $2^{121}$ concepts, then version space evenly split on *every* unseen instance (hw problem)
- Need *inductive bias* (smaller hypothesis space), otherwise generalization is hopeless
- Assume “tasty coffee” is a rectangle in $R^2$
- Rectangles in $R^2$ are concepts C for which there exists $c_1, c_2, c_3, c_4$ so that $C(x) = 1$ iff $c_1 \leq x_1 \leq c_2$ and $c_3 \leq x_2 \leq c_4$
Hypothesis class $\mathcal{H}$ of rectangles

$h(x) = \begin{cases} 
1 & \text{if } h \text{ classifies } x \text{ as positive} \\
0 & \text{if } h \text{ classifies } x \text{ as negative}
\end{cases}$

Error of $h$ on data $\mathcal{X}$

$$E(h) = \sum_{n=1}^{N} 1(h(x_n) \neq t_n)$$
A table and a diagram showing data points for strength and sugar features, with labels indicating whether they are positive (+) or negative (-) or unknown (?). The table includes:

<table>
<thead>
<tr>
<th>strength</th>
<th>sugar</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>strength</td>
<td>sugar</td>
<td>label</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>?</td>
</tr>
</tbody>
</table>
Noise

Data not always perfect

- Unmeasured Features
- Attribute noise
- Label noise
- Noise can model hypothesis space approximations

Domain

target
Noise

- Data not always perfect
- Attribute noise
- Label noise
- Noise can model hypothesis space approximations
Noise

- Data not always perfect
- Attribute noise
- Label noise
- Noise can model hypothesis space approximations

![Diagram showing a domain with a target circle and hypothesis rectangles]

Target-circle

hypothesis-rectangles

Domain
Key interplay

- Underlying pattern being learned
- Features available
- Hypothesis space
- Number of examples available

The trick is finding the right mix, but...
Key interplay

- Underlying pattern being learned
- Features available
- Hypothesis space
- Number of examples available

The trick is finding the right mix, but…

The smaller the hypothesis space, the luckier we have to be to catch the pattern
Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  - Complexity of \( \mathcal{H} \), \( c(\mathcal{H}) \),
  - Training set size, \( N \),
  - Generalization error, \( E \), on new data

  - As \( N \) increases, \( E \) decreases
  - As \( c(\mathcal{H}) \)↑, first \( E \)↓ and then \( E \)↑

- also PAC theory & VC-style generalization bounds
Model Selection & Generalization

- Learning is an **ill-posed problem**; data is not sufficient to find a unique solution
- The need for **inductive bias**, assumptions about $\mathcal{H}$
- **Generalization**: How well a model performs on new data - *What we really want!*
- **Overfitting**: find $h$ that is too complex
- **Underfitting**: find $h$ that is too simple
- **Regularization** can help (penalizes complexity)
Estimating Errors

- To estimate generalization error, we need data unseen during training. Often data split into
  - Training set (50%)
  - Validation set (25%) (did training work? Use for Parameter selection)
  - Final Test (publication) set (25%)
- Resampling when there are few examples (cross validation)
Summary: Supervised Learning as parameter estimation

Model (hypothesis class): \( g_\theta(x) \) (gives prediction)

Error/Loss function: \( E_\theta = \sum_{n=1}^{N} L(t_n, g_\theta(x_n)) \)

Optimization procedure: \( \hat{\theta} \approx \arg\min_{\theta} (E_\theta) \)
A (very) little on feature selection,
And then …
on to Bayesian learning (ch. 3-5)

- Feature selection reduces the dimensionality (number of features) used to describe the instances
Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Less parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions
Feature Selection vs Extraction

- **Feature selection**: Choosing \( k < d \) important features, ignoring the remaining \( d - k \) dimensions.
- **Feature extraction**: Project the original \( x_i, \ i = 1, ..., d \) dimensions to new \( k < d \) dimensions, \( z_j, \ j = 1, ..., k \)

Principal components analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)
(also clustering-based approaches)
see Guyon-Elisseeff
Example

- Not a rectangle in $x_1, x_2$
- How to make it a rectangle?

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Example

- Not a rectangle in $x_1, x_2$
- How to make it a rectangle?
- It is a rectangle in three dimensions: $x_1, x_2$, and $x_1 \times x_2$