Goals for the lecture

• define the supervised and unsupervised learning tasks
• consider how to represent instances as fixed-length feature vectors
• understand the concepts
  • instance (example)
  • feature (attribute)
  • feature space
  • feature types
  • model (hypothesis)
  • training set
  • supervised learning
  • classification (concept learning)
  • Regression
  • batch vs. online learning
  • i.i.d. assumption
  • generalization
Goals for the lecture (continued)

• understand the concepts
  • unsupervised learning
  • clustering
  • anomaly detection
  • dimensionality reduction

Can I eat this mushroom?

I don’t know what type it is – I’ve never seen it before. Is it edible or poisonous?
Can I eat this mushroom?

suppose we’re given examples of edible and poisonous mushrooms (we’ll refer to these as training examples or training instances)

edible

poisonous

can we learn a model that can be used to classify other mushrooms?

Representing instances using feature vectors

- we need some way to represent each instance
- one common way to do this: use a fixed-length vector to represent features (a.k.a. attributes) of each instance
- also represent class label of each instance

\[
\begin{align*}
\mathbf{x}^{(1)} &= \langle \text{bell, fibrous, gray, false, foul, ...} \rangle \\
\mathbf{x}^{(2)} &= \langle \text{convex, scaly, purple, false, musty, ...} \rangle \\
\mathbf{x}^{(3)} &= \langle \text{bell, smooth, red, true, musty, ...} \rangle \\
\mathbf{y}^{(1)} &= \text{edible} \\
\mathbf{y}^{(2)} &= \text{poisonous} \\
\mathbf{y}^{(3)} &= \text{edible}
\end{align*}
\]
Standard feature types

• **nominal** (including Boolean)
  – no ordering among possible values
  e.g. color \( \in \{ \text{red, blue, green} \} \) (vs. color = 1000 Hertz)

• **linear** (or **ordinal**)
  – possible values of the feature are totally ordered
  e.g. size \( \in \{ \text{small, medium, large} \} \) \( \xrightarrow{\text{discrete}} \)
  weight \( \in [0...500] \) \( \xrightarrow{\text{continuous}} \)

• **hierarchical**
  – possible values are partially **ordered** in an ISA hierarchy
  e.g. shape \( \rightarrow \)

\[
\begin{align*}
\text{closed} & \quad \text{polygon} & \quad \text{continuous} \\
\text{square} & \quad \text{triangle} & \quad \text{circle} & \quad \text{ellipse}
\end{align*}
\]

Feature hierarchy example
Lawrence et al., *Data Mining and Knowledge Discovery* 5(1-2), 2001

Structure of one feature!
Feature space

we can think of each instance as representing a point in a $d$-dimensional feature space where $d$ is the number of features

example: optical properties of oceans in three spectral bands
[Traykovski and Sosik, Ocean Optics XIV Conference Proceedings, 1998]

Another view of the feature-vector representation: a single database table

<table>
<thead>
<tr>
<th></th>
<th>feature 1</th>
<th>feature 2</th>
<th>. . .</th>
<th>feature $d$</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance 1</td>
<td>0.0</td>
<td>small</td>
<td>. . .</td>
<td>red</td>
<td>true</td>
</tr>
<tr>
<td>instance 2</td>
<td>9.3</td>
<td>medium</td>
<td>. . .</td>
<td>red</td>
<td>false</td>
</tr>
<tr>
<td>instance 3</td>
<td>8.2</td>
<td>small</td>
<td>. . .</td>
<td>blue</td>
<td>false</td>
</tr>
<tr>
<td>. .</td>
<td>. .</td>
<td>. .</td>
<td>. .</td>
<td>. .</td>
<td>. .</td>
</tr>
<tr>
<td>instance $n$</td>
<td>5.7</td>
<td>medium</td>
<td>. . .</td>
<td>green</td>
<td>true</td>
</tr>
</tbody>
</table>
The supervised learning task

problem setting
• set of possible instances: \( X \)
• unknown target function: \( f : X \rightarrow Y \)
• set of models (a.k.a. hypotheses): \( H = \{ h \mid h : X \rightarrow Y \} \)

given
• training set of instances of unknown target function \( f \): 
\[
\left( x^{(1)}, y^{(1)} \right), \left( x^{(2)}, y^{(2)} \right), \ldots, \left( x^{(m)}, y^{(m)} \right)
\]

output
• model \( h \in H \) that best approximates target function

The supervised learning task

• when \( y \) is discrete, we term this a classification task (or concept learning)

• when \( y \) is continuous, it is a regression task

• later in the semester, we will consider tasks in which each \( y \) is more structured object (e.g. a sequence of discrete labels)
Batch vs. online learning

In batch learning, the learner is given the training set as a batch (i.e. all at once)

\[
(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \ldots (x^{(m)}, y^{(m)})
\]

In online learning, the learner receives instances sequentially, and updates the model after each (for some tasks it might have to classify/make a prediction for each \(x^{(i)}\) before seeing \(y^{(i)}\))

\[
(x^{(1)}, y^{(1)}) \quad (x^{(2)}, y^{(2)}) \quad (x^{(i)}, y^{(i)})
\]

i.i.d. instances

- we often assume that training instances are independent and identically distributed (i.i.d.) – sampled independently from the same unknown distribution
- later in the course we’ll consider cases where this assumption does not hold
  - cases where sets of instances have dependencies
    - instances sampled from the same medical image
    - instances from time series
    - etc.
  - cases where the learner can select which instances are labeled for training
    - active learning
  - the target function changes over time (concept drift)
Generalization

• The primary objective in supervised learning is to find a model that generalizes – one that accurately predicts $y$ for previously unseen $x$.

Can I eat this mushroom that was not in my training set?

Model representations

throughout the semester, we will consider a broad range of representations for learned models, including

• decision trees
• neural networks
• support vector machines
• Bayesian networks
• logic clauses
• ensembles of the above
• etc.
Mushroom features (from the UCI Machine Learning Repository)

- cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
- cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
- cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
- bruises?: bruises=t, no=f
- odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
- gill-attachment: attached=a, descending=d, free=f, notched=n
- gill-spacing: close=c, crowded=d, distant=n
- gill-size: broad=b, narrow=n
- gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
- stalk-shape: enlarging=e, tapering=t
- stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=m
- stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
- stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
- veil-type: partial=p, universal=u
- veil-color: brown=n, orange=o, white=w, yellow=y
- ring-number: none=n, one=o, two=t
- ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
- spore-print-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, purple=u, white=w, yellow=y
- population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
- habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

A learned decision tree

- if odor=almond, predict edible
- if odor=none ∧ spore-print-color=white ∧ gill-size=narrow ∧ gill-spacing=crowded, predict poisonous
Classification with a learned decision tree

Once we have a learned model, we can use it to classify previously unseen instances.

\[ x = \{ \text{bell, fibrous, brown, false, foul, ...} \} \]

Unsupervised learning

In unsupervised learning, we're given a set of instances, without \( y \)'s:

\[ x^{(1)}, x^{(2)}, \ldots, x^{(m)} \]

Goal: discover interesting regularities that characterize the instances.

Common unsupervised learning tasks:

- *clustering*
- *anomaly detection*
- *dimensionality reduction*
Clustering

given
- training set of instances $x^{(1)}, x^{(2)}, \ldots x^{(m)}$

output
- model $h \in H$ that divides the training set into clusters such that there is intra-cluster similarity and inter-cluster dissimilarity

Clustering example

Clustering irises using three different features (the colors represent clusters identified by the algorithm, not $y$’s provided as input)
Anomaly detection

**learning task**

- given
  - training set of instances $x^{(1)}, x^{(2)} \ldots x^{(m)}$

**output**

- model $h \in H$ that represents “normal” $x$

**performance task**

- given
  - a previously unseen $x$
  
  **determine**
  - if $x$ looks normal or anomalous

---

**Anomaly detection example**

Let's say our model is represented by: 1979-2000 average, ±2 stddev

Does the data for 2012 look anomalous?
Dimensionality reduction

given
• training set of instances $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

output
• model $h \in H$ that represents each $x$ with a lower-dimension feature vector while still preserving key properties of the data

Dimensionality reduction example

We can represent a face using all of the pixels in a given image

More effective method (for many tasks): represent each face as a linear combination of eigenfaces
Dimensionality reduction example

represent each face as a linear combination of \textit{eigenfaces}

\[
x^{(1)} = \alpha^{(1)}_1 \times \begin{pmatrix} 1 \end{pmatrix} + \alpha^{(1)}_2 \times \begin{pmatrix} 1 \end{pmatrix} + \ldots + \alpha^{(1)}_{20} \times \begin{pmatrix} 1 \end{pmatrix}
\]

\[
x^{(1)} = \left< \alpha^{(1)}_1, \alpha^{(1)}_2, \ldots, \alpha^{(1)}_{20} \right>
\]

\[
x^{(2)} = \alpha^{(2)}_1 \times \begin{pmatrix} 1 \end{pmatrix} + \alpha^{(2)}_2 \times \begin{pmatrix} 1 \end{pmatrix} + \ldots + \alpha^{(2)}_{20} \times \begin{pmatrix} 1 \end{pmatrix}
\]

\[
x^{(2)} = \left< \alpha^{(2)}_1, \alpha^{(2)}_2, \ldots, \alpha^{(2)}_{20} \right>
\]

# of features is now 20 instead of # of pixels in images

Other learning tasks

later in the semester we’ll cover other learning tasks that are not strictly supervised or unsupervised

• \textit{reinforcement learning}
• \textit{semi-supervised learning}
• \textit{etc.}