Today:
• What is machine learning?
• Decision tree learning
• Course logistics

Readings:
• “The Discipline of ML”
• Mitchell, Chapter 3
• Bishop, Chapter 14.4

Machine Learning:

Study of algorithms that
• improve their performance P
• at some task T
• with experience E

well-defined learning task: <P,T,E>
Learning to Predict Emergency C-Sections

[Sims et al., 2000]

9714 patient records, each with 215 features

One of 18 learned rules:

If No previous vaginal delivery, and Abnormal 2nd Trimester Ultrasound, and Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: 26/41 = .63,
Over test data: 12/20 = .60

Learning to classify text documents

OPEC GIVE AWAY  Spam  vs  not spam

admin@rec.com
3:24 PM (3 hours ago)

OPEC Foreign Processing Department
> OPEC Fund for International Development (OFID)
> Martin Street, Birstall, Batley
> West Yorkshire, WF7 9PJ - UK
> 
> We wish to inform you of the OFID first quarter ballot final result. Your
email ID emerge in our 2nd category as a winner for a cash prize of
$100,000.00 (one hundred thousand USD). This is from 21 winners from email
list of 10,000,000 individuals, corporate and private organisations, NGO's and
public sectors selected globally in this category.
>
> The OPEC Fund for International Development (OFID) is a foundation owned
by the Organization of Petroleum Exporting Countries (OPEC). This foundation
is funded by member nations which include: Algeria, Indonesia, Iran, Iraq,
Kuwait, Libya, Nigeria, Qatar, United Arab Emirates and Venezuela.
>
> OFID is a development organization aimed at improving lives across the
world. This program tagged "Grass root Program" is part of efforts to improve
international housing problems, support the research for the eradication of
Ebola Virus and improve standard of living through direct participation in
community development across several communities all over the world by
empowering selected individuals as an engine for economic growth and social
development.
Learning to detect objects in images

(Prof. H. Schneiderman)

Example training images for each orientation

Learn to classify the word a person is thinking about, based on fMRI brain activity
Learning prosthetic control from neural implant

Machine Learning - Practice

Object recognition
- Support Vector Machines
- Bayesian networks
- Hidden Markov models
- Deep neural networks
- Reinforcement learning
- ....
Machine Learning - Theory

PAC Learning Theory (supervised concept learning)

- # examples ($m$)
- Error rate ($\varepsilon$)
- Representational complexity ($H$)
- Failure probability ($\delta$)

$m \geq \frac{1}{\varepsilon} (\ln |H| + \ln(1/\delta))$

Other theories for
- Reinforcement skill learning
- Semi-supervised learning
- Active student querying
- ...

... also relating:
- # of mistakes during learning
- Learner’s query strategy
- Convergence rate
- Asymptotic performance
- Bias, variance

Machine Learning in Computer Science

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - ...

- This ML niche is growing (why?)
Machine Learning in Computer Science

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - …

- This ML niche is growing
  - Improved machine learning algorithms
  - Increased volume of online data
  - Increased demand for self-customizing software

Tom’s prediction: ML will be fastest-growing part of CS this century
What You’ll Learn in This Course

• The primary Machine Learning algorithms
  – Logistic regression, Bayesian methods, HMM’s, SVM’s, reinforcement learning, decision tree learning, boosting, unsupervised clustering, …

• How to use them on real data
  – text, image, structured data
  – your own project

• Underlying statistical and computational theory

• Enough to read and understand ML research papers

Course logistics
Machine Learning 10-601

website: www.cs.cmu.edu/~ninamf/courses/601sp15

Faculty
• Maria Balcan
• Tom Mitchell

TA’s
• Travis Dick
• Kirsten Early
• Ahmed Hefny
• Micol Marchetti-Bowick
• Willie Neiswanger
• Abu Saparov

Course assistant
• Sharon Cavlovich

See webpage for
• Office hours
• Syllabus details
• Recitation sessions
• Grading policy
• Honesty policy
• Late homework policy
• Piazza pointers
• ...

Highlights of Course Logistics

On the wait list?
• Hang in there for first few weeks

Homework 1
• Available now, due friday

Grading:
• 30% homeworks (~5-6)
• 20% course project
• 25% first midterm (March 2)
• 25% final midterm (April 29)

Academic integrity:
• Cheating ➔ Fail class, be expelled from CMU

Late homework:
• full credit when due
• half credit next 48 hrs
• zero credit after that
• we’ll delete your lowest HW score
• must turn in at least n-1 of the n homeworks, even if late

Being present at exams:
• You must be there – plan now.
• Two in-class exams, no other final
Maria-Florina Balcan: Nina

- Foundations for Modern Machine Learning
  - E.g., interactive, distributed, life-long learning
- Theoretical Computer Science, especially connections between learning theory & other fields

Travis Dick

- When can we learn many concepts from mostly unlabeled data by exploiting relationships between concepts.
- Currently: Geometric relationships
Kirstin Early

- Analyzing and predicting energy consumption
- Reduce costs/usage and help people make informed decisions

**Predicting energy costs** from features of home and occupant behavior

**Energy disaggregation:** decomposing total electric signal into individual appliances

---

Ahmed Hefny

- How can we learn to track and predict the state of a dynamical system only from noisy observations?
- Can we exploit supervised learning methods to devise a flexible, local minima-free approach?

**Extracted 2D state trajectory**

**Observations** (oscillating pendulum)
Micol Marchetti-Bowick

How can we use machine learning for biological and medical research?

- Using genotype data to build personalized models that can predict clinical outcomes
- Integrating data from multiple sources to perform cancer subtype analysis
- Structured sparse regression models for genome-wide association studies

Willie Neiswanger

- If we want to apply machine learning algorithms to BIG datasets…
- How can we develop parallel, low-communication machine learning algorithms?
- Such as embarrassingly parallel algorithms, where machines work independently, without communication.

Example
Abu Saparov

- How can knowledge about the world help computers understand natural language?
- What kinds of machine learning tools are needed to understand sentences?

“Carolyn ate the cake with a fork.”

```
<table>
<thead>
<tr>
<th>person_eats_food</th>
</tr>
</thead>
<tbody>
<tr>
<td>consumer</td>
</tr>
<tr>
<td>Carolyn</td>
</tr>
<tr>
<td>food</td>
</tr>
<tr>
<td>cake</td>
</tr>
<tr>
<td>instrument</td>
</tr>
<tr>
<td>fork</td>
</tr>
</tbody>
</table>
```

“Carolyn ate the cake with vanilla.”

```
<table>
<thead>
<tr>
<th>person_eats_food</th>
</tr>
</thead>
<tbody>
<tr>
<td>consumer</td>
</tr>
<tr>
<td>Carolyn</td>
</tr>
<tr>
<td>food</td>
</tr>
<tr>
<td>cake</td>
</tr>
<tr>
<td>topping</td>
</tr>
<tr>
<td>vanilla</td>
</tr>
</tbody>
</table>
```

Tom Mitchell

How can we build never-ending learners?
Case study: never-ending language learner (NELL) runs 24x7 to learn to read the web

Recently-Learned Facts

<table>
<thead>
<tr>
<th>instance</th>
<th>mean avg. precision top 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>zillion_stars is a geometric shape</td>
<td></td>
</tr>
<tr>
<td>many_other_books is a kind of media</td>
<td></td>
</tr>
<tr>
<td>street_fighter_2_champion_edition is software</td>
<td></td>
</tr>
<tr>
<td>spicy_coconut_yogurt_chicken_breasts is a type of meat</td>
<td></td>
</tr>
<tr>
<td>infill_walls is something found in or on buildings</td>
<td></td>
</tr>
<tr>
<td>state_university is a sports team also known as notre_dame</td>
<td></td>
</tr>
<tr>
<td>harrods is a tourist attraction in the city london</td>
<td></td>
</tr>
<tr>
<td>weiskopf plays the sport golf</td>
<td></td>
</tr>
<tr>
<td>hat is a clothing item to go with coveralls</td>
<td></td>
</tr>
<tr>
<td>james_cameron directed the movie titanic</td>
<td></td>
</tr>
</tbody>
</table>

see  http://rtw.ml.cmu.edu
Function Approximation and Decision tree learning

Function approximation

**Problem Setting:**
- Set of possible instances $X$
- Unknown target function $f : X \rightarrow Y$
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$

**Input:**
- Training examples $\{ <x^{(i)}, y^{(i)}> \}$ of unknown target function $f$

**Output:**
- Hypothesis $h \in H$ that best approximates target function $f$
Simple Training Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

A Decision tree for 
\( f: \langle \text{Outlook, Temperature, Humidity, Wind} \rangle \rightarrow \text{PlayTennis?} \)

Each internal node: test one discrete-valued attribute \( X_i \)
Each branch from a node: selects one value for \( X_i \)
Each leaf node: predict \( Y \) (or \( P(Y|X \in \text{leaf}) \))
### Decision Tree Learning

**Problem Setting:**
- Set of possible instances $X$
  - each instance $x$ in $X$ is a feature vector
  - e.g., $<\text{Humidity}=$low, $\text{Wind}=$weak, $\text{Outlook}=$rain, $\text{Temp}=$hot$>$
- Unknown target function $f : X \rightarrow Y$
  - $Y=1$ if we play tennis on this day, else 0
- Set of function hypotheses $H=\{ h \mid h : X \rightarrow Y \}$
  - each hypothesis $h$ is a decision tree
  - trees sorts $x$ to leaf, which assigns $y$

![Decision Tree Diagram](image)

**Input:**
- Training examples $\{<x^{(i)}, y^{(i)}>\}$ of unknown target function $f$

**Output:**
- Hypothesis $h \in H$ that best approximates target function $f$
**Decision Trees**

Suppose \( X = <X_1, ..., X_n> \) where \( X_i \) are boolean-valued variables

How would you represent \( Y = X_2 \cdot X_5 \)? \( Y = X_2 \lor X_5 \)

How would you represent \( X_2 \cdot X_5 \lor X_3 \cdot X_4 (\neg X_i) \)

---

**A Tree to Predict C-Section Risk**

Learned from medical records of 1000 women

Negative examples are C-sections

\[
[833+,167-] .83+ .17-
\]

Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| Primiparous = 0: [399+,13-] .97+ .03-
| Primiparous = 1: [368+,68-] .84+ .16-
| Fetal_Distress = 0: [334+,47-] .88+ .12-
| Birth_Weight < 3349: [201+,10.6-] .95+ .
| Birth_Weight >= 3349: [133+,36.4-] .78+
| Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
Top-Down Induction of Decision Trees

$node = \text{Root}$

Main loop:

1. $A \leftarrow$ the “best” decision attribute for next $node$
2. Assign $A$ as decision attribute for $node$
3. For each value of $A$, create new descendant of $node$
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?

Sample Entropy

- $S$ is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in $S$
- $p_\ominus$ is the proportion of negative examples in $S$
- Entropy measures the impurity of $S$

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$
Entropy

Entropy $H(X)$ of a random variable $X$

$$H(X) = - \sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

$H(X)$ is the expected number of bits needed to encode a randomly drawn value of $X$ (under most efficient code)

Why? Information theory:
- Most efficient possible code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random $X$ is:

$$\sum_{i=1}^{n} P(X = i) (- \log_2 P(X = i))$$

Entropy

Entropy $H(X)$ of a random variable $X$

$$H(X) = - \sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

Specific conditional entropy $H(X|Y=v)$ of $X$ given $Y=v$:

$$H(X|Y = v) = - \sum_{i=1}^{n} P(X = i|Y = v) \log_2 P(X = i|Y = v)$$

Conditional entropy $H(X|Y)$ of $X$ given $Y$:

$$H(X|Y) = \sum_{v \in values(Y)} P(Y = v) H(X|Y = v)$$

Mutual information (aka Information Gain) of $X$ and $Y$:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$
Information Gain is the mutual information between input attribute $A$ and target variable $Y$.

Information Gain is the expected reduction in entropy of target variable $Y$ for data sample $S$, due to sorting on variable $A$:

$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y | A)$$

Simple Training Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Selecting the Next Attribute

Which attribute is the best classifier?

\[ S: [9+, 5-] \]
\[ E = 0.940 \]

**Humidity**

- High
- Normal

\[ [3+, 4-] \]
\[ E = 0.985 \]

\[ [6+, 1-] \]
\[ E = 0.992 \]

**Wind**

- Weak
- Strong

\[ [6+, 2-] \]
\[ E = 0.811 \]

\[ [3+, 3-] \]
\[ E = 1.00 \]

\[ \text{Gain} (S, \text{Humidity}) = .940 \cdot (7/14) \cdot 0.985 - (7/14) \cdot 0.992 = .151 \]

\[ \text{Gain} (S, \text{Wind}) = .940 \cdot (8/14) \cdot 0.811 - (6/14) \cdot 1.0 = .048 \]

\[ \{D1, D2, ..., D14\} \]
\[ [9+, 5-] \]

**Outlook**

- Sunny
- Overcast
- Rain

\[ \{D1, D2, D8, D9, D11\} \]
\[ [2+, 3-] \]

\[ \{D3, D7, D12, D13\} \]
\[ [4+, 0-] \]

\[ \{D4, D5, D6, D10, D14\} \]
\[ [3+, 2-] \]

Which attribute should be tested here?

\[ S_{\text{Sunny}} = \{D1, D2, D8, D9, D11\} \]

\[ \text{Gain} (S_{\text{Sunny}}, \text{Humidity}) = .970 \cdot (3/5) \cdot 0.0 - (2/5) \cdot 0.0 = .970 \]

\[ \text{Gain} (S_{\text{Sunny}}, \text{Temperature}) = .970 \cdot (2/5) \cdot 0.0 - (2/5) \cdot 1.0 - (1/5) \cdot 0.0 = .570 \]

\[ \text{Gain} (S_{\text{Sunny}}, \text{Wind}) = .970 \cdot (2/5) \cdot 1.0 - (3/5) \cdot 0.918 = .019 \]
Each internal node: test one discrete-valued attribute $X_i$
Each branch from a node: selects one value for $X_i$
Each leaf node: predict $Y$

Which Tree Should We Output?

- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

Occam's razor: prefer the simplest hypothesis that fits the data
Why Prefer Short Hypotheses? (Occam’s Razor)

Arguments in favor:

Arguments opposed:

Why Prefer Short Hypotheses? (Occam’s Razor)

Argument in favor:
• Fewer short hypotheses than long ones
  → a short hypothesis that fits the data is less likely to be a statistical coincidence
  → highly probable that a sufficiently complex hypothesis will fit the data

Argument opposed:
• Also fewer hypotheses with prime number of nodes and attributes beginning with “Z”
• What’s so special about “short” hypotheses?
Overfitting in Decision Trees

Consider adding noisy training example #15:

*Sunny, Hot, Normal, Strong, PlayTennis = No*

What effect on earlier tree?

![Decision Tree Diagram]

Overfitting

Consider a hypothesis $h$ and its

- Error rate over training data: $\text{error}_{\text{train}}(h)$
- True error rate over all data: $\text{error}_{\text{true}}(h)$

We say $h$ overfits the training data if

$$\text{error}_{\text{true}}(h) > \text{error}_{\text{train}}(h)$$

Amount of overfitting =

$$\text{error}_{\text{true}}(h) - \text{error}_{\text{train}}(h)$$
Overfitting in Decision Tree Learning

Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune
Reduced-Error Pruning

Split data into training and validation set

Create tree that classifies training set correctly

Do until further pruning is harmful:

1. Evaluate impact on validation set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves validation set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning
Continuous Valued Attributes

Create a discrete attribute to test continuous

- \(Temperature = 82.5\)
- \((Temperature > 72.3) = t, f\)

<table>
<thead>
<tr>
<th>Temperature:</th>
<th>40</th>
<th>48</th>
<th>60</th>
<th>72</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayTennis:</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Attributes with Many Values

Problem:

- If attribute has many values, \(Gain\) will select it
- Imagine using \(Date = Jun.3.1996\) as attribute

One approach: use \(GainRatio\) instead

\[
GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}
\]

\[
SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}
\]

where \(S_i\) is subset of \(S\) for which \(A\) has value \(v_i\)
You should know:

- Well posed function approximation problems:
  - Instance space, $X$
  - Sample of labeled training data $\{ <x^{(i)}, y^{(i)}> \}$
  - Hypothesis space, $H = \{ f: X \rightarrow Y \}$

- Learning is a search/optimization problem over $H$
  - Various objective functions
    - minimize training error (0-1 loss)
    - among hypotheses that minimize training error, select smallest (?)

- Decision tree learning
  - Greedy top-down learning of decision trees (ID3, C4.5, ...)
  - Overfitting and tree/rule post-pruning
  - Extensions…

Questions to think about (1)

- ID3 and C4.5 are heuristic algorithms that search through the space of decision trees. Why not just do an exhaustive search?
Questions to think about (2)

• Consider target function $f: \langle x_1, x_2 \rangle \rightarrow y$, where $x_1$ and $x_2$ are real-valued, $y$ is boolean. What is the set of decision surfaces describable with decision trees that use each attribute at most once?

Questions to think about (3)

• Why use Information Gain to select attributes in decision trees? What other criteria seem reasonable, and what are the tradeoffs in making this choice?
Questions to think about (4)

• What is the relationship between learning decision trees, and learning IF-THEN rules

Learned from medical records of 1000 women
Negative examples are C-sections

[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
  | Previous_Ceception = 0: [767+,81-] .90+ .10-
  | Primiparous = 0: [399+,13-] .97+ .03-
  | Primiparous = 1: [368+,68-] .84+ .16-
  | | Fetal_Distress = 0: [334+,47-] .88+ .12-
  | | | Birth_Weight < 3349: [204+,10.6-] .96+ .
  | | | Birth_Weight >= 3349: [133+,36.4+] .78+
  | | | Fetal_Distress = 1: [34+,21-] .62+ .38-
  | Previous_Ceception = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-

One of 18 learned rules:

If No previous vaginal delivery, and Abnormal 2nd Trimester Ultrasound, and Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: 26/41 = .63,
Over test data: 12/20 = .60