Machine Learning

Russ Greiner
Alberta Ingenuity Centre for Machine Learning
Department of Computing Science
University of Alberta
Diagnosing Butterfly-itis

Hmmm... perhaps Butterfly-it is??
Classifying a Patient

<table>
<thead>
<tr>
<th>#wing</th>
<th>#ant</th>
<th>nectar-orient</th>
<th>...</th>
<th>color</th>
<th>butterfly</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>Y</td>
<td>...</td>
<td>Pale</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>N</td>
<td>...</td>
<td>Clear</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>N</td>
<td>...</td>
<td>Pale</td>
<td>No</td>
</tr>
</tbody>
</table>

#antennae

Learner

Classifier

butterfly

No
Visualizing Patient Data

+ = yes  - = No

- What about this new patient?
This is learning...

- Given data:

  ![Data points graph]

- Predicting “label” of new patient
  - Here: Negative “−” (not butterfly—it is)

- This is an **EDUCATED GUESS**:
  - ... not based on post-mortem, definitive test, ...
  - use to decide on treatment, etc.
Challenges to Learning

- Easy:

- Harder:

- High Dimension:
Machine Learning studies ...

Computers that use “experiences” to improve *performance* of some system

Computers that use “*annotated data*” to *autonomously* produce effective “rules”

- to diagnose diseases
- to identify relevant articles
- to assess credit risk
- ...

8
Outline

- Successes
  - Mining Data Sets
  - Sequential Analysis
  - Control
- Basic ideas
  - Foundations
  - Algorithms
  - Statistical Issues
- Current Research
Successes: Mining Data Sets
Computer learns...

- to find ideal customers
  Credit Card approval (AMEX)
  - Humans ≈50%; ML is >70%!

- to find best person for job
  Telephone Technician Dispatch [Danyluk/Provost/Carr 02]
  - BellAtlantic used ML to learn rules to decide which technician to dispatch
  - Saved $10+ million/year

- to predict purchasing patterns
  - Victoria Secret (stocking)

- to help win games
  - NBA (scouting)

- to catalogue celestial objects [Fayyad et al. 93]
  - Discovered 22 new quasars
  - >92% accurate, over tetrabytes
2: Sequential Analysis

- **BioInformatics 1**: identifying genes
  - Glimmer [Delcher et al, 95]
  - identifies 97+% of genes, automatically!

- **BioInformatics 2**: Predicting protein function, ...

- **Recognizing Handwriting**

- **Recognizing Spoken Words**
  - “How to wreck a nice beach”
3: Control

- **TD-Gammon** *(Tesauro 1993; 1995)*
  - World-champion level play by *learning* ...
  - by playing millions of games against itself!

- **Drive autonomous vehicles** *(Thrun 2005)*
  - DARPA Grand Challenge

- **Printing Press Control** *(Evans/Fisher 1992)*
  - Control rotogravure printer, prevent groves, ...
    - specific to each plant
  - More complete than human experts
  - Used for 10+ years, reduced problems from 538/year to 26/year!

- **Oil refinery**
  - Separate oil from gas
  - ... in 10 minutes (human experts require 1+ days)

- **Manufacture nuclear fuel pellets** *(Leech, 86)*
  - Saves Westinghouse >$10M / year

- **Adaptive** agents / user-interfaces
### US/Canada Data Miners - your current annual income or status: [144 votes total]

<table>
<thead>
<tr>
<th>Category</th>
<th>Votes</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>student (46)</td>
<td></td>
<td>32%</td>
</tr>
<tr>
<td>unemployed (10)</td>
<td></td>
<td>7%</td>
</tr>
<tr>
<td>$60,000 or less (20)</td>
<td></td>
<td>14%</td>
</tr>
<tr>
<td>61 to 80 (18)</td>
<td></td>
<td>13%</td>
</tr>
<tr>
<td>81 to 100 (17)</td>
<td></td>
<td>12%</td>
</tr>
<tr>
<td>101 to 120 (21)</td>
<td></td>
<td>15%</td>
</tr>
<tr>
<td>over 120 (12)</td>
<td></td>
<td>8%</td>
</tr>
</tbody>
</table>

- Alberta Ingenuity Centre for Machine Learning
Outline

- Successes
- Basic ideas
  - Foundations
  - Algorithms
  - Statistical Issues
- Current Research
Learning is ... Training a Classifier

<table>
<thead>
<tr>
<th>Temp.</th>
<th>Press.</th>
<th>Sore Throat</th>
<th>...</th>
<th>Colour</th>
<th>diseaseX</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>95</td>
<td>Y</td>
<td>...</td>
<td>Pale</td>
<td>No</td>
</tr>
<tr>
<td>22</td>
<td>110</td>
<td>N</td>
<td>...</td>
<td>Clear</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>10</td>
<td>87</td>
<td>N</td>
<td>...</td>
<td>Pale</td>
<td>No</td>
</tr>
</tbody>
</table>

**Learner**

**Classifier**

<table>
<thead>
<tr>
<th>Temp</th>
<th>Press.</th>
<th>Sore- Throat</th>
<th>...</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>90</td>
<td>N</td>
<td></td>
<td>Pale</td>
</tr>
</tbody>
</table>
Why Learn?
Why not just “program it in”?

Appropriate Classifier ...

- ... is not known
  Medical diagnosis... Credit risk... Control plant...
- ... is too hard to “engineer”
  Drive a car... Recognize speech...
- ... changes over time
  Plant evolves...
- ... user specific
  Adaptive user interface...
Why Machine Learning is especially relevant **now**!

- **Growing flood of online data**
  - customer records, telemetry from equipment, scientific journals, ...

- **Recent progress in algorithms and theory**
  - SVM, Reinforcement Learning, Boosting, ...
  - PAC-analysis, SRM, ...

- **Computational power** is available
  - networks of fast machines

- **Budding industry** in many application areas
  - market analysis, adaptive process control, decision support, ...
Outline

- Successes
- Basic ideas
  - Foundations
  - Algorithms
    - Linear Separators
      - Support Vector Machines
    - Artificial Neural Nets
    - Decision Trees
    - Naïve Bayes
    - Nearest Neighbor, ...
  - Statistical Issues
    - Current Research
    - UofA + AICML
General Process

- Given “labeled data”

- Learn CLASSIFIER, that can predict label of NEW instance
Alg 1: Linear Separators

- Draw “separating line”

- If \( \#\text{antennae} \leq 2 \), then butterfly-itis
- So \( ? \) is Not butterfly-itis.
Can be “angled”...

2.3 × #w + -7.5 × #a + 1.2 = 0

- If \( 2.3 \times \#Wings + -7.5 \times \#antennae + 1.2 > 0 \)
  
  then butterfly-itis
Linear Separators, in General

- Given data (many features)

<table>
<thead>
<tr>
<th>F₁</th>
<th>F₂</th>
<th>...</th>
<th>Fₙ</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>95</td>
<td>...</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>22</td>
<td>80</td>
<td>...</td>
<td>-2</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>...</td>
<td>1.9</td>
<td>No</td>
</tr>
</tbody>
</table>

- find “weights” \( \{w₁, w₂, \ldots, wₙ, w₀\} \)
  such that

\[
\sum_{i=1}^{n} w_i F_i + w₀ > 0
\]

means \( \text{Class} = \text{Yes} \)
Linear Separator

\[ \sum_i w_i \times F_i \]

Just view \( F_0 = 0 \), so \( w_0 \) ...
Linear Separator

- Challenge:
  - Given labeled data, find “correct” \( \{ w_i \} \)
  - “Perceptron”
Support Vector Machine (SMO)

- Many linear separators ...
- Which is best?
Support Vector Machine (SMO)

- Decision boundary should be as far away as possible from the data
  \[ \Rightarrow \text{maximize margin, } m \]
Linear Separators – Facts

- **GOOD NEWS:**
  - If data is linearly separated,
  - Then **FAST ALGORITHM** finds correct \( \{w_i\} \)

- But...
Linear Separators – Facts

- **GOOD NEWS:**
  - If data is linearly separated,
  - Then **FAST ALGORITHM** finds correct $\{w_i\}$!

- But...

- Some “data sets” are **NOT** linearly separable!
Alg 2: Artificial Neural Nets

- Why not use *SET* of connected Linear Separators?

F_1 \rightarrow \Sigma_i w_i \times F_i \rightarrow \text{output}

F_2 \rightarrow \Sigma_i w_i \times F_i \rightarrow \text{output}

\vdots

F_n \rightarrow \Sigma_i w_i \times F_i \rightarrow \text{output}

\rightarrow \Sigma_i w_i \times F_i \rightarrow \text{output}

\begin{cases}
\text{Yes} \\
\text{No}
\end{cases}

Skip
Artificial Neural Nets

- Can Represent *ANY* classifier!
  - w/ just 1 “hidden” layer...
  - in fact...

\[ \sum_i w_i \times F_i \]
ANNs: Architecture

- Different # of layers
- Different structures
  - what’s connected to what..
- Different “squashing function”
Uses of Artificial Neural Nets

Learning to ...

- drive a car
- assess credit risk
- pronouncing words (NETtalk)
- recognize handwritten characters
- control plant
- ...

32
Algorithm 3: Decision Trees

- Given data, decide on best *first* split

- Then consider each subset of data:
  - decide on its best split

- Recur... until “purity”
Alg 3: Decision Trees

- Consider data:

  - “Hierarchical Split”
    - Divide and conquer

![Decision Tree Diagram]
Alg 3: Decision Trees

- Partitioned data:
- "Hierarchical Split"
  - Divide and conquer
Issues ⇒ Demo

- Issues:
  - How to split?
  - When to stop?
  - Avoid overfitting
  - Real vs Discrete

- AIXploratorium!
  http://www.cs.ualberta.ca/~aixplore
Alg 4: Naïve Bayes

- **First-order stats:**
  - $P(\text{SNP}_1 = 1 \mid \text{“Dx=+” })$
  - $P(\text{SNP}_1 = 2 \mid \text{“Dx=+” })$
  - $P(\text{SNP}_1 = 3 \mid \text{“Dx=+” })$
  - $P(\text{SNP}_1 = 1 \mid \text{“Dx=-” })$
  - $P(\text{SNP}_1 = 2 \mid \text{“Dx=-” })$
  - $P(\text{SNP}_1 = 3 \mid \text{“Dx=-” })$

- Similarly for $\text{SNP}_2, \text{SNP}_3, \ldots \text{SNP}_{53}$
Naïve Bayes, con’t

| b  | P(SNP₁=1 | Dx=b) | P(SNP₁=2 | Dx=b) | P(SNP₁=3 | Dx=b) |
|----|---------|---------|---------|---------|
| +  | 0.05    | 0.92    | 0.03    |
| -- | 0.80    | 0.19    | 0.01    |
Naïve Bayes, con’t

![Diagram showing Naïve Bayes model with SNP2 and Dx nodes]

| b  | P(SNP2=1 | Dx=b) | P(SNP2=2 | Dx=b) | P(SNP2=3 | Dx=b) |
|----|---------|---------|---------|---------|
| +  | 0.15    | 0.05    | 0.80    |
| -- | 0.73    | 0.10    | 0.17    |
Naïve Bayes, con’t

Dx

SNP_{53}

| b  | P(SNP_{53}=1 | Dx=b) | P(SNP_{53}=2 | Dx=b) | P(SNP_{53}=3 | Dx=b) |
|----|-------------|-------------|-------------|-------------|
| +  | 0.90        | 0.05        | 0.05        |
| -- | 0.70        | 0.20        | 0.10        |
Naïve Bayes, con’t

\[
P(\pm b \mid s_1, s_2, \ldots, s_{53}) = \frac{1}{z} P(\pm b) \prod_i P(SNP_i = s_i \mid \pm b)
\]

\[
P(\pm b \mid s_1, s_2, \ldots, s_{53}) = \frac{1}{z} P(\pm b) \prod_i P(SNP_i = s_i \mid \mp b)
\]

Answer: Take larger of

\[
\begin{align*}
P(\pm b) \prod_i P(SNP_i = s_i \mid \pm b) \\
P(\mp b) \prod_i P(SNP_i = s_i \mid \mp b)
\end{align*}
\]
Classification

- Which is more likely: +h vs –h ?

- Given independencies:
  + values:

  \[ \text{argmax}_h P( h \mid +b, \neg j ) \]
  \[ = \text{argmax}_h P( h ) \times P( +b \mid h ) \times P( \neg j \mid h ) \]
  \[ = \text{argmax}_h \{ 0.05 \times 0.95 \times 0.2, \ 0.95 \times 0.03 \times 0.7 \} \]
  \[ -h \text{ as } 0.0095 < 0.01995 \]
"Naïve Bayes"

Classification Task:

Given \( \{O_1 = v_1, ..., O_n = v_n\} \)

Find \( h_i \) that maximizes \( P(H = h_i \mid O_1 = v_1, ..., O_n = v_n) \)

\[
\begin{align*}
P(H = h_i) \\
P(O_j = v_k \mid H = h_j)
\end{align*}
\]

\text{Independent: } P(O_j \mid H, O_k, \ldots) = P(O_j \mid H)

\[
P(H = h_i \mid O_1 = v_1, \ldots, O_n = v_n) = \frac{1}{\alpha} P(H = h_i) \prod_j P(O_j = v_j \mid H = h_i)
\]

Find \( \text{argmax} \ \{h_i\} \)
Other Algorithms

- Nearest Neighbor
- Learning “Ensembles”
  - Ways to combine “ok” classifiers, to be better
  - Boosting, Bagging, Stacking, ...
- More than just + vs – ...
  - {Ok, MildSick, AverageSick, VerySick}
  - Real values $\mathbb{R}$
Outline

- Successes
- Basic ideas
  - Foundations
  - Algorithms
  - Statistical Issues
    1. Goal of learning
    2. Why should Learning work?
    3. How much data is needed?
    4. How to evaluate a classifier?
    5. Overfitting
    6. Computational Efficiency
    7. Imbalanced data (fraud detection)
    8. Non-IID tuples (stock market, temporal)
- Current Research
1. Goal of Learning?

- If goal of learning is just score well on *training data* ...
- *Trivial:* just memorize data!
  - \{ a is No, b is Yes, d is No \}
- Instead: want to do well on
  - *NEW UNSEEN data*
- On e =

- How can learning possibly succeed?
2. Why should Learning work?

- **Rare is rare**
  - If patient type is *common*, then it is in sample
    - If in sample, classifier “gets” it
  - If patient type NOT *common*, then ... so what?
    - Classifier will be wrong, but penalty is small
- Overfitting can be prevented
- More data is better
Why should Learning work?

- Overall Population

- Draw sample \( S = \) \( a a a b a a \)
- Learn classifier \( C \) that does well on \( S \):
  - As \( S \) includes \( a \ b \), \[ C(a) = \text{No} \]
  - \( C(b) = \text{Yes} \)
- Notice \( d \) not is \( S \)
  \[ C(d) = ?? \]
How good is Classifier $C$?

- To evaluate $C$
  - Draw new patient, $x \in \{a, b\}$
  - Compute $C(x)$
  - Correct?

Given true distribution,

- $\text{expect } x = a$ ... or $x = b$
  - Here: $C(x)$ is correct!
- Otherwise, $C(x)$ may be wrong.
  - But this is rare!
Why should Learning work?

Consider a new patient, \( x \) ...

1. If \( x \) occurs a LOT \( \quad P(x) >> 0 \)
   - \( x \) probably appears in \( S \)
   - As \( C \) does well on \( S \),
     \( C \) gives correct answer on \( x \)

2. If \( x \) occurs rarely \( \quad P(x) \approx 0 \)
   - doesn’t matter if \( C \) is wrong!

- Even good classifiers are wrong occasionally...
Populations

- Train a “Feline classifier” $FC$ using
  - Pets in my neighborhood,
- $FC$ should do well on
  - household cats +
  - household dogs –
- $FC$ will probably be WRONG wrt
  - Tigers

- Not surprising: $FC$ was NOT trained on them!
Similar Patients...

- So far: assume many IDENTICAL patients
  - Same values for each feature

- More realistic:  *Similar* patients...

- Same idea:
  - if need to classify \( x \), and \( x \sim u \) where \( u \in S \),
  - then \( C(u) \approx C(x) \) and probably correct ...

<table>
<thead>
<tr>
<th></th>
<th>( F_1 )</th>
<th>( F_2 )</th>
<th>...</th>
<th>( F_n )</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>35</td>
<td>95</td>
<td>...</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>a'</td>
<td>36</td>
<td>95</td>
<td>...</td>
<td>2.1</td>
<td>No</td>
</tr>
<tr>
<td>b</td>
<td>22</td>
<td>80</td>
<td>...</td>
<td>-2</td>
<td>Yes</td>
</tr>
<tr>
<td>b'</td>
<td>22</td>
<td>78</td>
<td>...</td>
<td>-2.3</td>
<td>Yes</td>
</tr>
<tr>
<td>d</td>
<td>10</td>
<td>50</td>
<td>...</td>
<td>1.9</td>
<td>No</td>
</tr>
</tbody>
</table>
3. How much training data?

- What is best linear separator for...

- Makes a difference: what is “?”?

- Learning gets easier with more training data...
More data helps...

- Suppose next training point is...
- Eliminates 2nd option...
  - Leaving only $? = +$
Learnability Theory

Can QUANTIFY how many training instances are needed, as function of

- Hypothesis space
  - Linear Separators, Decision Trees, ...
- Accuracy required
- Chance of being completely wrong

(Think of Hypothesis Testing...)
4. How to Evaluate a Classifier?

**Labeled Training Data**

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>...</th>
<th>SNP53</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/A</td>
<td>C/C</td>
<td>T/T</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
<tr>
<td>A/A</td>
<td>C/C</td>
<td>A/T</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>A/A</td>
<td>C/T</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>G/A</td>
<td>C/T</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>No</td>
</tr>
</tbody>
</table>

**TRAIN**

**TEST**

Training Set Error... too optimistic
4. How to Evaluate a Classifier?

Labeled Training Data

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>...</th>
<th>SNP53</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/A</td>
<td>C/C</td>
<td>T/T</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
<tr>
<td>A/A</td>
<td>C/C</td>
<td>A/T</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>A/A</td>
<td>C/T</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

TRAIN

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>...</th>
<th>SNP53</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/G</td>
<td>A/G</td>
<td>T/T</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
<tr>
<td>T/C</td>
<td>C/C</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>G/A</td>
<td>T/C</td>
<td>G/G</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
</tbody>
</table>

TEST

Learner

Classifier

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>...</th>
<th>SNP53</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/A</td>
<td>C/C</td>
<td>T/T</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
<tr>
<td>A/A</td>
<td>C/C</td>
<td>A/T</td>
<td>...</td>
<td>T/T</td>
<td>No</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>G/A</td>
<td>C/T</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Training Set Error
... too optimistic
How to Evaluate a Classifier?

**Labeled Training Data**

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/A</td>
<td>C/C</td>
<td>T/T</td>
<td>No</td>
</tr>
<tr>
<td>A/A</td>
<td>C/C</td>
<td>A/T</td>
<td>Yes</td>
</tr>
<tr>
<td>A/A</td>
<td>C/T</td>
<td>A/A</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/A</td>
<td>C/T</td>
<td>A/A</td>
<td>No</td>
</tr>
</tbody>
</table>

**TRAIN**

<table>
<thead>
<tr>
<th>SNP1</th>
<th>SNP2</th>
<th>SNP3</th>
<th>...</th>
<th>SNP53</th>
<th>Dx?</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/G</td>
<td>A/G</td>
<td>T/T</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
<tr>
<td>T/C</td>
<td>C/C</td>
<td>A/A</td>
<td>...</td>
<td>T/T</td>
<td>Yes</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td></td>
<td>:</td>
<td></td>
</tr>
<tr>
<td>G/A</td>
<td>T/C</td>
<td>G/G</td>
<td>...</td>
<td>T/C</td>
<td>No</td>
</tr>
</tbody>
</table>

**TEST**

**Simple Hold-out Set Error**

... slightly pessimistic
How to Evaluate a Classifier?

- K-fold Cross Validation
  - Eg, K=3
- Not as pessimistic
  - every point is test example, once
Estimating Error: Cross Validation

“Cross-Validation”

\[
\text{CV}(\text{data } S, \text{alg } L, \text{int } k) \\
\text{Divide } S \text{ into } k \text{ disjoint sets } \{ S_1, S_2, \ldots, S_k \} \\
\text{For } i = 1..k \text{ do} \\
\quad \text{Run } L \text{ on } S_{-i} = S - S_i \\
\quad \text{obtain } h_i := L(S_{-i}) \\
\quad \text{Evaluate } h_i \text{ on } S_i \\
\quad \text{err}_{S_i}(h_i) = \frac{1}{|S_i|} \sum_{(x, y) \in S_i} [h_i(x) - t]^2 \\
\text{Return Average } \frac{1}{k} \sum_i \text{err}_{S_i}(h_i)
\]

⇒ Less Pessimistic

as train on \((k - 1)/k \ |S|\) of the data
Comments on Cross-Validation

- Every point used as Test 1 time, Training k – 1 times

- Computational cost for k-fold Cross-validation ... linear in k

- Should use “balanced CV”
  If class $c_i$ appears in $m_i$ instances,
  
  $$\text{insist each } S_k \text{ include } \approx \frac{1}{k} \frac{m_i}{|S_i|} \text{ such instances}$$

- Use $\text{CV}(S, L, k)$ as ESTIMATE of true error of $L(S)$
  Return $L(S)$ and $\text{CV}(S, L, k)$

- Leave-One-Out-Cross-Validation $k = m !$
  - eg, for Nearest-Neighbor

- Notice different folds are correlated as training sets overlap: $(k-2)/k$ unless $k=2$

- $5 \times 2$-CV
  - Run 2-fold CV, 5 times. . .

Can use CV to estimate parameters in general!
To Form k \textit{Balanced} Folds

1. Partition the data \( S \) based on the class:
   - subset \( S_+ \) has all the positive instances,
   - subset \( S_- \) has all the negative instances.

2. Randomly partition each subset into \( k \) folds --
   - \( S_+ = \bigcup \{ S_{+1}, \ldots, S_{+k} \} \)
   - \( S_- = \bigcup \{ S_{-1}, \ldots, S_{-k} \} \)

3. \( S_j = S_{+j} \cup S_{-j} \) for \( j=1..k \)
Summary of Classifier Results

<table>
<thead>
<tr>
<th>Classifier Accuracy</th>
<th>% Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Logistic</td>
<td>65.85</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>70.73</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>68.29</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>60.98</td>
</tr>
<tr>
<td>&quot;Just say No&quot;</td>
<td>65.85</td>
</tr>
</tbody>
</table>

... using 51 SNPs with 10-fold cross validation
Why so bad? Overfitting!

- What is best separator for...

- Compare:

- Sometimes appropriate to IGNORE details of training data
  - perhaps some training data points are mislabeled!
  - ... or **some features are irrelevant, misleading** ...

- Solution:
  - **Feature selection**
5. “Overfitting”

- Spse we used the WRONG features:
  - whether birthday was odd/even,
  - whether SSN was odd/even
  - whether car license odd/even
  - ...

- Here: NO correlation between
  - butterfly-itis and
  - any (combination) of feature

- Best classifier:
  - Ignore features; just use majority class
Example – continued

- 25% have butterfly-itis
- ½ of patients have $F_1 = 1$
  - Eg: “odd birthday”
- ½ of patients have $F_2 = 1$
  - Eg: “even SSN”
- ... for 10 features
- Decision Tree results
  - over 1000 patients (using these silly features) ...
Decision Tree Results

- Standard decision tree learner:

- Error Rate:
  - Train data: 0%
  - New data: 37%

- Optimal decision tree:

- Error Rate:
  - Train data: 25%
  - New data: 25%
Overfitting

- Some features are not helpful
- Data often noisy
  - typos in recording, error in equipment, human error...
- What is best separator for...

- Sometimes:
  Appropriate to IGNORE details of training data
  - Here: one training data point is mislabeled!
- Simpler hypothesis often better classifier!
  - eg, LINEAR Separator
Overfitting

- Compare...

- $C_1$ appears better (on training data) than $C_2$, but $C_2$ is actually better
- *Overfitting*!
- To address this... reduce dimensionality...
Reduce dimensions...

- Principle Component Analysis
- Feature Selection?
- Sort features based on *Information Gain*

\[
I(A, C) = \sum_{c,a} P(c, a) \log \frac{P(c, a)}{P(c) P(a)}
\]

- Notice:
  - 0 if attribute A is NOT correlated with class C
  - Positive if correlated
- Considers each attribute independent of others!
  Take top k features...

- ...

70
How many features?

- Perhaps try each value $k=1, 2, 3, \ldots$ and see how well each classifier does, on test set

- **No!!**
  - Must NOT use the test set to help learner
    - by selecting the number of features...
  - $\Rightarrow$ no longer unbiased!!!

- Test set only unbiased if you *never never never never* do any *any any any any* learning on the test data
Significance

- Found 8 most “informative” SNPs:
  - \{gn\_3001\_xrcc3, gn\_3040\_cyp2d6\_4, gn\_2442\_mlh1\_1, gn\_2469\_brca2\_12\_13, gn\_3012\_rad51, gn\_3010\_nbs1, gn\_961\_brca1\_5\_a201g, gn\_3002\_xrcc3\}

- Reasonable... associated with cell damage

- Classifier (using 8 snps) was 78% accuracy

- Is this significant?  
  ... especially given our complicated process?

- Suppose NO signal in data.  
  Trivial to get \(\approx 65.9\%\)
Permutation Test

- Randomly rearrange LABELs of data…
  … so no signal left …
- Run thru same algorithm
- Get results (CV)

Shuffle the labels!!
Significance

- One run might still do well. How about 4000 trials ...
- Results of permutation tests:

Conclusion:
- This Works!
Prostate Study #1b

- Cancer vs. No Cancer
  ... using same SNPs
- No correlation found!
  permutation tests result:
Accuracy... Permutation Tests

Naive Bayes Predictive SNP Model

- 2 Standard Deviations
- Naive Bayes Prediction
- 2 Standard Deviations
- Permuted Label Prediction

Percent Accuracy vs Number of SNPs Used With Information Gain Feature Selection
Outline

- Successful
- Basic ideas
  - Foundations
  - Algorithms
  - Statistical Issues
    1. Goal of Learning
    2. Why should Learning work?
    3. How much data is needed?
    4. How to Evaluate Classifier?
    5. Overfitting
    6. Computational Efficiency
    7. Imbalanced data (fraud detection)
    8. Non-IID tuples (stock market, temporal)
    9. Other types of learning
- Current research
Other Types of Learning

- Density Estimation
- Learning Generative Model
- Clustering
Other Types of Learning

- Density Estimation
- Learning Generative Model
- Clustering

- Learning Sequence of Actions
- Reinforcement Learning

Non-IID Data
Other Types of Learning

- Learning non-IID Data
  - Sequences
  - Images
  - ...
Other Types of Learning

- Density Estimation
  - Learning Generative Model
  - Clustering
- Learning Sequence of Actions
  - Reinforcement Learning
- Learning non-IID Data
  - Images
  - Sequences
  - ...
My Research I: Application Pull

- Brain Tumor Growth Prediction
- Proteome Analyst
- Human Metabolome Project
  - Inventory of ALL relevant small molecules ...
- PolyomX
  - Patient-specific cancer treatment
- Understanding Microarray Data
- Complete-Web Recommendation System
  - Find/use “browsing patterns” to identify important *words*, then important *pages*...
My Research II: Technology Push

- Support-Vector Random Fields
- Budgeted Learning
- Computing Variance of Belief Net Response
  - Mixture Using Variance
- Learning Belief Nets
  - Learning Generative Structure
  - Learning Discriminative Structure
  - Learning Discriminative Parameters
- ...

83
Summary

- Machine Learning is a **mature field**
  - solid theoretical foundation
  - many effective algorithms

- ML is **crucial** to large number of important **applications**
  - BioInformatics, WebReDesign, MarketAnalysis, Fraud Detection, ...

- Fun: Lots of intriguing open questions!
- **Exciting time for Machine Learning**
Thank you!