36-350 Data Mining
Tom Minka

Day 3
Using probabilistic models

Categorical case

- Given batch of categorical observations
- Independent and identically distributed
- Sample proportions \( \hat{p}(x) \) = probabilities
  \[ \hat{p}(x) = \frac{\text{number of } x's \text{ in batch}}{\text{size of batch}} = p(x) \]
- Thus infer probabilities (with error)
- Non-parametric: can model arbitrary distribution

Density estimation

- Inference problem: going beyond the sample
- Given sample, want to know about wider population or process
- Result is probability histogram or density curve

Corrected estimate

- Zero counts lead to zero probabilities
  - Not safe
- All counts should be started at 1 (or some other small value):
  \[ p(x) = \frac{(\text{number of } x's \text{ in batch}) + 1}{(\text{size of batch}) + (\text{number of bins})} \]

Applications

- Classification
  - Text: news articles, web pages
  - Imagery: natural scenes, face recognition
- Anomaly detection
  - Cellular phone fraud
  - UNIX intrusions

Classification problem

- Given samples of predefined populations:
  \[ A \quad B \quad C \]
- Determine most likely population of an unlabeled sample
  \[ ? \]
Maximum-likelihood classification

- Estimate distribution of each population:
  \[ \hat{p}(x | \text{pop}) \]  
  \text{pop} = A, B, or C
- Given new sample, compute probability it could have arisen from A, B, or C
- **Likelihood** of each population:
  \[ L(\text{pop}) = p(\text{sample} | \text{pop}) \]
- Assign sample to population with largest L

Text classification procedure

1. Collect all articles labeled “politics” into single batch of words
2. Words are categorical observations, with about 100,000 possible values
3. Compute probability histogram (100,000 bins)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>.0619</td>
<td>president</td>
<td>.0023</td>
</tr>
<tr>
<td>to</td>
<td>.0332</td>
<td>government</td>
<td>.0024</td>
</tr>
<tr>
<td></td>
<td></td>
<td>advance</td>
<td>.0004</td>
</tr>
</tbody>
</table>

Computing L

- Let sample = \{ y_1, ..., y_n \}
- Under independence assumption:
  \[ p(\text{sample} | \text{pop}) = \prod p(y_i | \text{pop}) \]
  \[ = \prod \hat{p}(y_i | \text{pop}) \]
  \[ \log L(\text{pop}) = \sum \log \hat{p}(y_i | \text{pop}) \]
- If value x occurs \( n_x \) times,
  \[ \log L(\text{pop}) = \sum n_x \log \hat{p}(x | \text{pop}) \]

Text classification procedure

- To classify a document, sum over all 100,000 words:
  \[ \log L(\text{politics}) = \sum n_w \log \hat{p}(w | \text{politics}) \]
- Independence assumption not truly satisfied
  - Doesn’t cause serious problems
  - More advanced models are possible, e.g. time series of word observations

Text classification

- News articles: business, politics, religion, etc.
- An article is a sample of words from a word population: business words, politics words, etc.
- Classify an article by most likely population of words it was drawn from
- Popular, successful technique

News monitoring

- Find news articles which are predictive of a change in company stock
- Population A: accompanied by no change in stock
- Population B: accompanied by large change in stock
- Fawcett and Provost, 1999
Class priors

• Some classes are more probable than others, even before we see the sample: $p(\text{class})$
• Use Bayes’ theorem: $p(\text{class} | \text{sample}) = \frac{p(\text{sample} | \text{class})p(\text{class})}{\sum_{\text{class}} p(\text{sample} | \text{class})p(\text{class})}$
• Choose most probable class
• Same as most likely class if priors are equal

Costs

• Different classification errors may have different costs
• E.g. classifying nuclear reactor as “stable” when it isn’t
• Cost of saying $A$ when truth is $B$: $C(A | B)$
• Choose class which minimizes $C(A | \text{sample}) = \sum C(A | B)p(B | \text{sample})$
More complex image model

- Schneiderman & Kanade (2000)
- An image is a set of sub-images sampled from a population of sub-images
- One histogram bin for every possible sub-image, after quantization (about 6,561 bins)
- Requires huge amounts of labeled data

Anomaly detection

- Given a sample from a population:
  \[ \hat{p}(x | A) \]
- Determine if an unlabeled sample is likely to be from the same population

Face, car detection

Solution

- Estimate distribution of the population: \( \hat{p}(x | A) \)
- Given new sample, compute probability it could have arisen from A:
  \[ p(\text{sample} | \text{pop}) \]
- If probability is too small, the sample is anomalous: \[ p(\text{sample} | \text{pop}) < \tau \]

Machine learning methods

- Often based on simple statistical models (or equivalent)
- Tend to ignore inference issues, proper estimation, model checking
- Main issues are computation, object representation

Choosing the threshold

- Low threshold = missed anomalies
- High threshold = false positives
- Generally \( \tau \) is set as high as tolerable
- Resampling training set gives expected number of false positives
Applications of anomaly detection

- Similarity
  - Retrieving similar documents, images
  - Query by example
- Dissimilarity
  - Activity monitoring, surveillance
  - Fraud detection
  - Computer intrusions

Potentially frauded account

<table>
<thead>
<tr>
<th>Time</th>
<th>Day</th>
<th>Length</th>
<th>From</th>
<th>To</th>
<th>Fraud?</th>
</tr>
</thead>
<tbody>
<tr>
<td>10am</td>
<td>Mon</td>
<td>13m</td>
<td>NY</td>
<td>CT</td>
<td></td>
</tr>
<tr>
<td>3pm</td>
<td>Fri</td>
<td>5m</td>
<td>NY</td>
<td>NY</td>
<td></td>
</tr>
<tr>
<td>1pm</td>
<td>Tue</td>
<td>9m</td>
<td>NY</td>
<td>CT</td>
<td></td>
</tr>
<tr>
<td>2am</td>
<td>Wed</td>
<td>35s</td>
<td>MA</td>
<td>NY</td>
<td>Y</td>
</tr>
<tr>
<td>9pm</td>
<td>Thu</td>
<td>24s</td>
<td>MA</td>
<td>MA</td>
<td>Y</td>
</tr>
</tbody>
</table>

Cellular cloning fraud

- Cellphones continually broadcast their serial number and customer ID, without encryption
- Inexpensive equipment can catch these numbers and program a second phone to use them
- Free, untraceable calls!
- Even PINs are unencrypted

Caller profiling

- Make categorical variable x ranging over (time, location) combinations
- Compute probability histogram of x for each customer:

<table>
<thead>
<tr>
<th>(Time, Location)</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(9am, NY)</td>
<td>.12</td>
</tr>
<tr>
<td>(5pm, NY)</td>
<td>.09</td>
</tr>
<tr>
<td>(10am, NY)</td>
<td>.04</td>
</tr>
<tr>
<td>(11pm, MA)</td>
<td>.01</td>
</tr>
</tbody>
</table>

Catching fraud

- Classification doesn’t work
  - Bandit population isn’t distinct from legitimate population
  - An unusual call for you is typical for me
- Must spot differences from a customer’s profile
- Individual calls are not enough evidence
  - Must use batches

Fraud detection

- Compute probability of today’s calls:
  \[
  p(\text{calls} \mid \text{profile}) = \prod_i p(\text{call}_i \mid \text{profile})
  \]
- Flag account if \( p(\text{calls} \mid \text{profile}) < t \)
- Choose t based on size of fraud dept.
- Can also incorporate potential cost of fraud
- Calls are not really independent
UNIX intrusion

• Prior to ssh, telnet had same problems as cellphones
• Security holes allow crackers to log in as legitimate users
• Must spot differences from user's profile
  – Which commands are used

Recurring problem

• Most applications require reducing the number of bins (quantization)
  – Words, colors, times, locations, UNIX commands
• For computational as well as estimation reasons
• What is best way to reduce?

Catching intrusion

• Make categorical variable ranging over UNIX commands
• Compute probability histogram for each user:

<table>
<thead>
<tr>
<th>Command</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>ps</td>
<td>.03</td>
</tr>
<tr>
<td>gcc</td>
<td>.05</td>
</tr>
<tr>
<td>kill</td>
<td>.001</td>
</tr>
<tr>
<td>ns</td>
<td>.001</td>
</tr>
</tbody>
</table>

Catching intrusion

• Each login session is sequence of commands
• For each session, compute

\[ p(\text{commands} \mid \text{profile}) = \prod_{j} p(\text{command}_j \mid \text{profile}) \]

• Fawcett and Provost, 1999