Clustering Markov States into Equivalent Classes using SVD and Heuristic Search Algorithms

Xianping Ge, Sridevi Parise, Padhraic Smyth

Information and Computer Science
University of California, Irvine

http://www.datalab.uci.edu/
Abstract

**Goal:** To approximate a *large* Markov model (e.g., $10^5$ states) with a *smaller* one (e.g., 50 states).

**SVD-based Algorithm:** Data matrix transformed to a (permuted) *block-constant* matrix. (Block $\equiv$ cluster)

**Search-based Algorithms:** Move a state from one cluster to another, so as to maximize data likelihood. (Search for “best” state $\rightarrow$ cluster assignment)
Too Many States?

- $M$-state Markov model has $M^2$ transition probabilities.
  - $M = 50000$ web pages on www.ics.uci.edu
  - $M \approx 400$ UNIX commands in Purdue UNIX user dataset
  - $M = 140072$ English words in Wiretap/Classic corpus

- Difficult to estimate the $M^2$ transition probabilities from limited data.

- Solution: cluster the states!
Problem Statement

Data: A set of sequences generated by a $M$-state Markov model.
  - Or, equivalently, a $M \times M$ matrix of transition counts.
    ($n_{y,y'}$: number of times that $y$ is followed by $y'$.)

Task: To cluster the $M$ states into $K$ clusters ("super-states"),
  where $K \ll M$.
  - $\Rightarrow K$-state Markov model

Goal: To maximize $P(\text{data}|K$-state Model).
SVD-Based Algorithm

- Permutated block-constant matrix: \((s(y): \text{cluster of } y)\)

\[
H_{y,y'} \equiv P \left( s(y') | s(y) \right) / P \left( s(y') \right) \approx \frac{n_{y,y'n}}{n_y n_{y'}}.
\]

\(H_{y,y'}\) depends only on \(s(y')\), \(s(y)\) (i.e., not on \(y', y\)).

<table>
<thead>
<tr>
<th>s</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>c</td>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>c</td>
<td>d</td>
<td>d</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>d</td>
<td>d</td>
<td>c</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>d</td>
<td>d</td>
<td>c</td>
</tr>
</tbody>
</table>

- SVD of matrix with \(K \times K\) constant blocks:
  - \(K\) nonzero singular values
  - Each singular vector is piecewise-constant w.r.t. the clusters.
SVD-Based Algorithm

1. Run SVD on $B$ where $B_{y, y'} = \frac{n_{y, y'} n_{y'}}{n_y n_{y'}}$. Let the nonzero singular values be $\sigma_1, \ldots, \sigma_K$, corresponding to left and right singular vectors $u_1, \ldots, u_K, v_1, \ldots, v_K$.

2. Run a clustering algorithm (e.g., K-means) on the rows of matrix

$$[\sigma_1 u_1, \ldots, \sigma_K u_K, \sigma_1 v_1, \ldots, \sigma_K v_K]$$

- SVD minimizes sum of squared errors (SSE) of matrix $B$ which implies Gaussian noise: misfit with insufficient amount of data.
  - The above algorithm (“SVD-B”) runs SVD on $B$ where $B_{y, y'} = \frac{n_{y, y'} n_{y'}}{n_y n_{y'}}$.
  - “SVD-C”: run SVD on $C$ where $C_{y, y'} = n_{y, y'}$.

- Related work: spectral clustering (e.g., in image segmentation)
Accuracy of the SVD methods

Accuracy

Data Set Size (Number of Sequences)

SVD-C
MNCut
SVD-B
Random
Clustering As a Search Problem

- Each clustering solution is an assignment of $M$ states to $K$ clusters.
- Search the solution space (of all possible assignments) for the best solution!
  - Start from a (random) initial assignment
  - Repeatedly try moving a state from one cluster to another, so as to increase data likelihood.
Search Algorithms

Score function: data likelihood

Heuristics:

GSAT: Of the $M \times (K - 1)$ possible moves, choose the one that leads to the largest increase in the log-likelihood.

GSAT with 10% sampling

ICM: Go through the states sequentially, moving each state to the cluster that maximizes the log likelihood.

ICM with randomized order

Simulated Annealing
## Accuracy & Computation Times on Simulated Data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Time/ICM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSAT</td>
<td>0.889</td>
<td>16.8</td>
</tr>
<tr>
<td>GSAT with 10% sampling</td>
<td>0.881</td>
<td>3.3</td>
</tr>
<tr>
<td>ICM</td>
<td>0.885</td>
<td>1.0</td>
</tr>
<tr>
<td>ICM with randomized order</td>
<td>0.895</td>
<td>1.1</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>0.959</td>
<td>316.3</td>
</tr>
<tr>
<td>SVD-C</td>
<td>0.681</td>
<td>18.1</td>
</tr>
<tr>
<td>Unconstrained HMM</td>
<td>0.43</td>
<td>1464.6</td>
</tr>
<tr>
<td>Random Assignment</td>
<td>0.234</td>
<td>-</td>
</tr>
</tbody>
</table>
Application to User Modeling: Purdue UNIX user data

- Data: Sequences of UNIX commands; each sequence is a session (from login to logout).
- $M \approx 400$ distinct UNIX commands.
- We ran ICM-style clustering algorithm on each user’s sequences, with $K = 10$.
- Note the edit-compile-run cycle (states 9, 3, and 4).
## Clustering of English Words

<table>
<thead>
<tr>
<th>god</th>
<th>would</th>
<th>said</th>
<th>come</th>
</tr>
</thead>
<tbody>
<tr>
<td>life</td>
<td>will</td>
<td>asked</td>
<td>go</td>
</tr>
<tr>
<td>night</td>
<td>could</td>
<td>cried</td>
<td>look</td>
</tr>
<tr>
<td>death</td>
<td>can</td>
<td>replied</td>
<td>love</td>
</tr>
<tr>
<td>course</td>
<td>did</td>
<td>says</td>
<td>use</td>
</tr>
<tr>
<td>water</td>
<td>should</td>
<td>answered</td>
<td>help</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>him</th>
<th>good</th>
<th>men</th>
<th>place</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>long</td>
<td>people</td>
<td>work</td>
</tr>
<tr>
<td>them</td>
<td>better</td>
<td>things</td>
<td>side</td>
</tr>
<tr>
<td>us</td>
<td>high</td>
<td>years</td>
<td>light</td>
</tr>
<tr>
<td>himself</td>
<td>true</td>
<td>words</td>
<td>part</td>
</tr>
<tr>
<td>home</td>
<td>dead</td>
<td>days</td>
<td>power</td>
</tr>
</tbody>
</table>

- Corpus: *Wiretap Online Library / Classic.* 92M bytes.
- Sequences: sentences.
- $M = 140072$ distinct words.
- $K = 48$ word clusters.
- See left for top words in 8 of the clusters.
itoldjohnitwasyou
it old john it was you
i told john it was you

iamanamerican
i a man american
i am an american

whichendedinadeadheat
which ended in a dead heat

\( P = e^{-6.1} \)
\( P = e^{-2.1} \)
\( P = e^{-4.4} \)

(a) (b)