Case Study 2: Document Retrieval

Spectral Clustering

Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Emily Fox
February 14th, 2013

©Emily Fox 2013

New Approach: Spectral Clustering

- **Goal:** Cluster observations
- **Method:**
  - Use similarity metric between observations
  - Form a similarity graph
  - Use standard linear algebra and optimization techniques to cut graph into connected components (clusters)
Setup

- Data: $x^1, \ldots, x^N$
- Similarity metric: $s_{ij}$

- Similarity graph
  - Nodes $v^i$
  - Edge weights $w_{ij} = f(s_{ij})$

- Problem: Want to partition graph such that edges between groups have low weights

Graph Terminology I

- Weighted adjacency matrix
  $$W = \{w_{ij}\} : i,j=1,\ldots,N$$
Issues with MinCut

- MinCut favors isolated clusters

Cuts Accounting for Size

- Ratio cuts (RatioCut)
- Normalized cuts (Ncut)
- Lead to “balanced” clusters
Restating Cut Metric

\[ x^T D x \]  
\[ x^T W x \]  
\[ x^T (D - W) x \]

Ratio Cuts for General k

- Define cluster indicator variables:
\[
F_{ij} = \begin{cases} 
\frac{1}{\sqrt{|A_j|}} & v^i \in A_j \\ 
0 & \text{otherwise} 
\end{cases} 
\]

RatioCut
\[
\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} f'_{Ai} L f_{Ai} = \text{Tr}(F'_A L F_A)
\]

Reformulating RatioCut problem
\[
\min_{A_1, \ldots, A_k} \text{Tr}(F'_A L F_A) \quad \text{s.t.} \quad F'_A F_A = I
\]

Relaxation
\[
\min_{F \in \mathbb{R}^{N \times k}} \text{Tr}(F' L F) \quad \text{s.t.} \quad F' F = I
\]

Solution: \( F = 1^{st} k \) eigenvectors of \( L \)
Normalized Cuts for General k

- Define cluster indicator variables:
  \[ F_{ij} = \begin{cases} 1/\sqrt{\text{vol}(A_j)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases} \]
  \[ F_A F_A' = I \]
  \[ F_A' D F_A = I \]

- Reformulating RatioCut problem
  \[ \min_{A_1, \ldots, A_k} \text{Tr}(F_A' L F_A) \quad \text{s.t.} \quad F_A' D F_A = I \]

- Relaxation
  \[ \min_{H \in \mathbb{R}^N \times k} \text{Tr}(H' D^{-1/2} L D^{-1/2} H) \quad \text{s.t.} \quad H' H = I \]

- Solution:
  - \( H \) is matrix of first \( k \) eigenvectors of \( L_{sym} \), which is equivalent to the approximate \( F \) being the first \( k \) eigenvectors of \( L_{rw} = I - D^{-1} W \)

Random Walks on Graphs

- Stochastic process with random jumps from \( v_i \) to \( v_j \) with:
  \[ p_{ij} = \frac{w_{ij}}{d_i} \]

- Transition matrix:
  \[ P = D^{-1} W \]

- Connection to graph Laplacian:
  \[ L_{rw} = I - D^{-1} W = I - P \]

- Intuitively, want to partition graph s.t. random walk stays in cluster for a while and rarely jumps between clusters
Random Walks on Graphs

- Assume that stationary distribution exists and is unique. Then,
  \[ \pi = (\pi_1, \ldots, \pi_n) \quad \pi_i = \frac{d_i}{\text{vol}(V)} \]

- Proposition: \( \text{Ncut}(A, \bar{A}) = P(A \mid A) + P(A \mid \bar{A}) \)

- Proof:
  \[ \text{Ncut}(A, \bar{A}) = \frac{P(X_0 \in A, X_1 \in \bar{A})}{P(X_0 \in A)} = \frac{\sum_{i \in \bar{A}, j \in B} \pi_i \pi_j}{\text{vol}(A)} \]

- Minimizing normalized cuts is equivalent to minimizing the probability of transitioning between clusters

Case Study 3: fMRI Prediction

fMRI Prediction Task, LASSO Regression

Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington

Emily Fox
February 14th, 2013
fMRI Prediction Task

- **Goal:** Predict word stimulus from fMRI image

  \[ \text{Can we read your brain?} \]

  ![fMRI Image]

  **Classifier** (logistic regression, kNN, ...)

  ![Hammer Image]

  ![House Image]

fMRI
fMRI

- High res.
  - ~1 mm resolution
  - Pretty slow
- 1 image per sec.
- Safe, non-invasive
- Measures Blood Oxygen Level Dependent (BOLD) response

Typical fMRI response to impulse of neural activity

Typical Stimuli

- Each stimulus repeated several times
  - Shown multiple times

- A
  - X
  - X
  - X
  - X
- B
  - hammer
  - X
  - X
  - X
  - X
- Dog
  - X
  - X
  - X
  - X
- Airplane
  - X
  - X
  - X
  - X
- Eye
  - X
  - X
  - X
  - X
- Hammer
  - X
  - X
  - X
  - X
fMRI Activation

fMRI activation for "bottle":

Mean activation averaged over 60 different stimuli:

"bottle" minus mean activation:

is this enough?

fMRI Prediction Task

- **Goal**: Predict word stimulus from fMRI image
- **Challenges**:
  - \( p \gg N \) (feature dimension >> sample size)
  - Cost of fMRI recordings is high
  - Only have a few training examples for each word

Classifier (logistic regression, kNN, …)

HAMMER
or
HOUSE
Zero-Shot Classification

- **Goal**: Classify words not in the training set
- **Challenges**:
  - Cost of fMRI recordings is high
  - Can’t get recordings for every word in the vocabulary

**Never showed “giraffe” in scanner**

- **Classifier** (logistic regression, kNN, …)

HAMMER or HOUSE

---

Zero-Shot Classification

- **Goal**: Classify words not in the training set
- **Challenges**:
  - Cost of fMRI recordings is high
  - Can’t get recordings for every word in the vocabulary
  - We don’t have many brain images, but we have a lot of info about the words and how they relate (co-occurrence, etc.)
  - How do we utilize this “cheap” information?

**Never showed “giraffe” in scanner**

- **Classifier** (logistic regression, kNN, …)

HAMMER or HOUSE
Semantic Features

Google Trillion word corpus

Semantic feature values: “celery”
- 0.8368, eat
- 0.3461, taste
- 0.3153, fill
- 0.2430, see
- 0.1145, clean
- 0.0600, open
- 0.0586, smell
- 0.0286, touch
- ...
- 0.0000, drive
- 0.0000, wear
- 0.0000, lift
- 0.0000, break
- 0.0000, ride

Semantic feature values: “airplane”
- 0.8673, ride
- 0.2891, see
- 0.2851, say
- 0.1689, near
- 0.1228, open
- 0.0883, hear
- 0.0771, run
- 0.0749, lift
- ...
- 0.0049, smell
- 0.0010, wear
- 0.0000, taste
- 0.0000, rub
- 0.0000, manipulate

Zero-Shot Classification

- From training data, learn two mappings:
  - $S$: input image $\rightarrow$ semantic features
  - $L$: semantic features $\rightarrow$ word

- Can use “cheap” co-occurrence data to help learn $L$

Classifier (logistic regression, $k$NN, …)

Features of word

Predict

HAMMER or HOUSE

Training: $\text{saw} [\ldots] \rightarrow \text{dog}, \text{many}$

Use both $A + B$

Predict: $\text{saw} [\ldots] \rightarrow \text{dog}$

$A = \{ \ldots \}$

$B = \{ \ldots \}$

$N$ examples $\ldots N$ small
fMRI Prediction Subtask

- **Goal:** Predict semantic features from fMRI image

\[ \text{Learning } S: \text{ images } \rightarrow \text{ semantic features} \]

Features of word

\[ y^i \]

\[ x^i \]

\[ X^{20,000} \]

\[ y^{\text{d}} \]

\[ y^{\text{d}} \]

\[ d = \# \text{ of semantic features} \in \mathbb{R}^d \]

\[ \hat{\beta} = \arg \min_{\beta} \text{RSS}(\beta) = \arg \min_{\beta} \sum_{i=1}^{N} (y^i - \beta^T x^i)^2 \]

\[ \hat{\theta} = \arg \max_{\theta} \log p(D \mid \theta) \]

Linear Regression – *review*

- **Model:**
  \[ y^i = \beta_0 + \beta_1 x_1^i + \cdots + \beta_p x_p^i + \epsilon^i \]
  \[ = \beta^T x^i + \epsilon^i \]
  \[ \epsilon^i \sim N(0, \sigma^2) \Rightarrow y^i \sim N(\beta^T x^i, \sigma^2) \]

- **MLE:**
  \[ \hat{\beta} = \arg \min_{\beta} \text{NLL}(\beta) = \arg \min_{\beta} \frac{1}{2} \text{RSS}(\beta) = \frac{1}{2} \sum_{i=1}^{N} (y^i - \beta^T x^i)^2 \]

- Minimizing RSS = least squares regression
Linear Regression – review

- Taking the gradient
  - Reformulate objective
    \[
    \begin{bmatrix}
    e_1^t \\
    \vdots \\
    e_n^t 
    \end{bmatrix} = \begin{bmatrix}
    y_1 \\
    \vdots \\
    y_n 
    \end{bmatrix} - \begin{bmatrix}
    x_1^t \\
    \vdots \\
    x_n^t 
    \end{bmatrix} \begin{bmatrix}
    \beta_0 \\
    \vdots \\
    \beta_p 
    \end{bmatrix}
    \]
    \[
    \frac{1}{2} \text{RSS}(\beta) = \frac{1}{2} (y - XB)^T (y - XB) = \frac{1}{2} \beta^T (X^T X) \beta - \beta^T (X^T y)
    \]
  - Set gradient = 0
    \[
    \frac{\partial}{\partial \beta} \text{RSS}(\beta) = \frac{1}{\partial \beta} (X^T X \beta - X^T y) = 0 + \text{const.}
    \]
    \[
    \Rightarrow \hat{\beta}_{\text{ML}} = (X^T X)^{-1} X^T y \text{ low rank prep matrix. !!!}
    \]

Ridge Regression

- Ameliorating issues with overfitting:
  - New objective:
    \[
    \min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda \| \beta \|_2^2
    \]
    \[
    \min_{\beta} \text{RSS}(\beta) \text{ s.t. } \| \beta \|_2 \leq S
    \]
  - Reformulate:
    \[
    F(\beta) = \frac{1}{2} \beta^T (X^T X) \beta - \beta^T (X^T y) + \text{const.} + \frac{1}{2} \lambda \beta^T \beta
    \]
    \[
    = \frac{1}{2} \beta^T (X^T X + \lambda I) \beta - \beta^T (X^T y) + \text{const.}
    \]
  - Set gradient = 0
    \[
    \hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} (X^T y)
    \]
Ridge Regression

- Solution is indexed by the regularization parameter $\lambda$
- Larger $\lambda$: high reg.
- Smaller $\lambda$: low reg.
- As $\lambda \to 0$ \( \hat{\beta}_{\text{ridge}} \to \hat{\beta}_{\text{ML}} \)
- As $\lambda \to \infty$ \( \hat{\beta}_{\text{ridge}} \to 0 \)

Ridge Coefficient Path

- Typical approach: select $\lambda$ using cross validation (CV)

From Kevin Murphy textbook
Variable Selection

- Ridge regression: Penalizes large weights

- What if we want to perform “feature selection”?  
  - E.g., Which regions of the brain are important for word prediction?  
  - Can’t simply choose predictors with largest coefficients in ridge solution  
  - Computationally impossible to perform “all subsets” regression

- Try new penalty: Penalize non-zero weights
  - Penalty:  
    \[ ||\beta||_1 = \sum |\beta_j| \]
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param \( \lambda \)

Acknowledgements

- Some material in this lecture was based on slides provided by:
  - Jianbo Shi – spectral clustering
  - Tom Mitchell – fMRI
  - Rob Tibshirani – LASSO