Case Study 2: Document Retrieval

Spectral Clustering

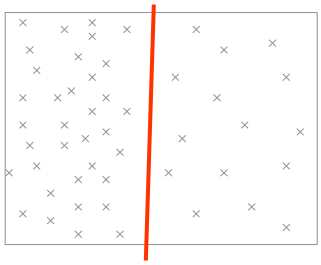
Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington

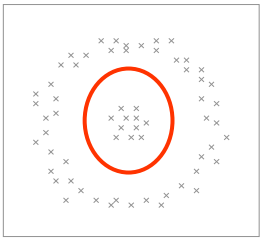
Emily Fox

February 14th, 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)

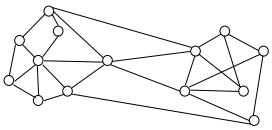




Setup

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

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

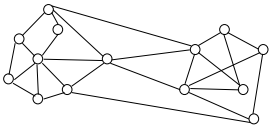


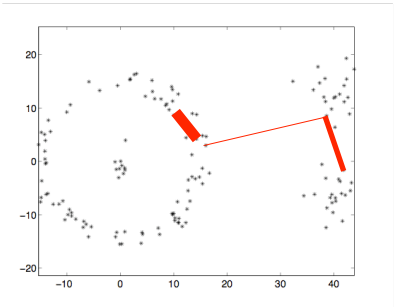
G = {V, E}

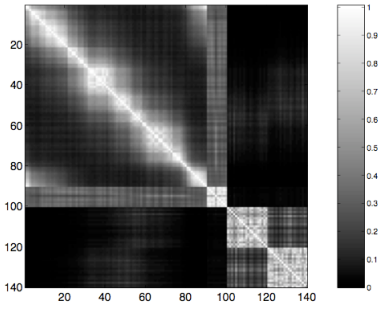
- Problem: Want to partition graph such that edges between groups have low weights

Graph Terminology I

- Weighted adjacency matrix

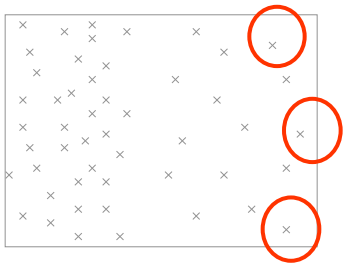






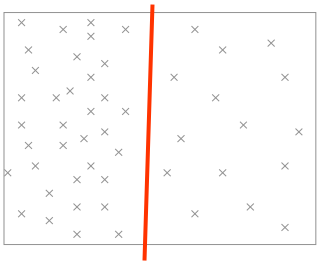
Issues with MinCut

- MinCut favors isolated clusters

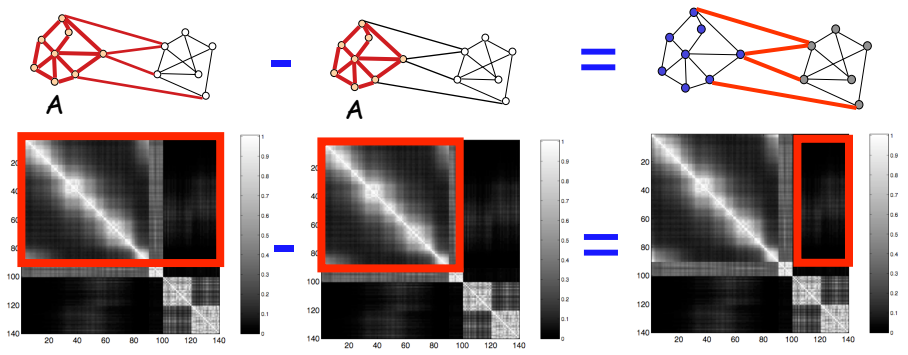


Cuts Accounting for Size

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



Restating Cut Metric



$x^T D x$ $\quad$ $x^T W x$ $\quad$ $x^T (D - W) x$

Ratio Cuts for General k

- Define cluster indicator variables:

$$F_{ij} = \begin{cases} 1/\sqrt{|A_j|} & v^i \in A_j \\ 0 & otherwise \end{cases}$$

$$F'_{\mathcal{A}} F_{\mathcal{A}} = I$$

$$F_{\mathcal{A}} \in \mathbb{R}^{N \times k}$$

- RatioCut

$$\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} f'_{\mathcal{A}i} L f_{\mathcal{A}i} = \text{Tr}(F'_{\mathcal{A}} L F_{\mathcal{A}})$$

- Reformulating RatioCut problem

$$\min_{A_1,\ldots,A_k} \text{Tr}(F'_{\mathcal{A}} L F_{\mathcal{A}}) \quad \text{s.t. } F'_{\mathcal{A}} F_{\mathcal{A}} = I$$

- Relaxation

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

Normalized Cuts for General k

- Define cluster indicator variables:

$$F_{ij} = \begin{cases} 1/\sqrt{\text{vol}(A_j)} & v_i \in A_j \\ 0 & ow \end{cases} \qquad \begin{aligned} F_{\mathcal{A}}' F_{\mathcal{A}} &= I \\ F_{\mathcal{A}}' D F_{\mathcal{A}} &= I \end{aligned}$$

- Reformulating RatioCut problem

$$\min_{A_1,\ldots,A_k} \text{Tr}(F_{\mathcal{A}}' L F_{\mathcal{A}}) \text{ s.t. } F_{\mathcal{A}}' D F_{\mathcal{A}} = I$$

- Relaxation

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

- 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}

Random Walks on Graphs

- Stochastic process with random jumps from v_i to v_j wp:
- Transition matrix:
- Connection to graph Laplacian:
- 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,
- Proposition: $\mathrm{Ncut}(A, \bar{A}) = P(A \mid \bar{A}) + P(\bar{A} \mid A)$
- Proof:
- Minimizing normalized cuts is equivalent to minimizing the probability of transitioning between clusters

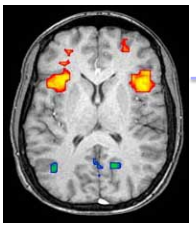
Case Study 3: fMRI Prediction

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



Classifier
(logistic regression, kNN, …)

~~HAMMER~~
or
HOUSE

fMRI

fMRI

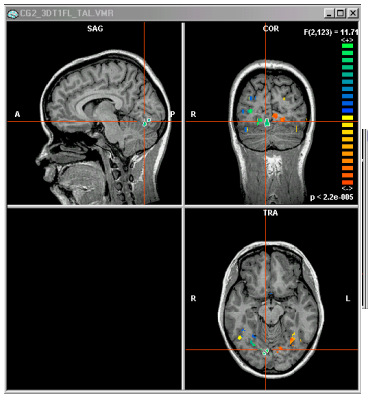
~1 mm resolution

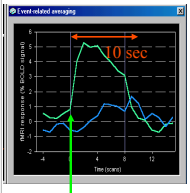
~1 image per sec.

20,000 voxels/image

safe, non-invasive

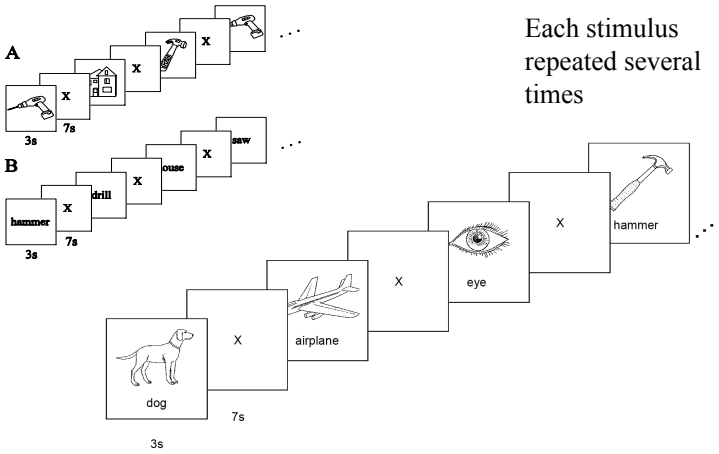
measures Blood Oxygen Level Dependent (BOLD) response





Typical fMRI response to impulse of neural activity

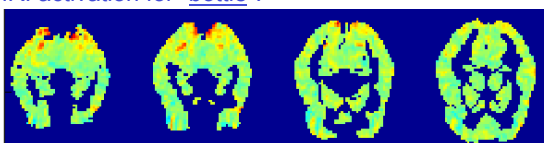
Typical Stimuli



Each stimulus repeated several times

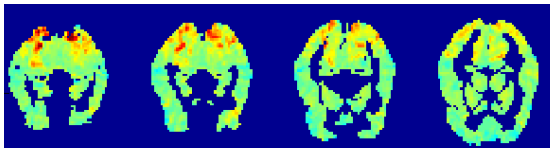
fMRI Activation

fMRI activation for "bottle":



bottle

Mean activation averaged over 60 different stimuli:



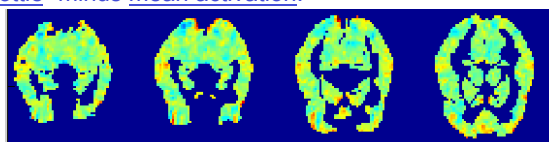
fMRI activation

high

average

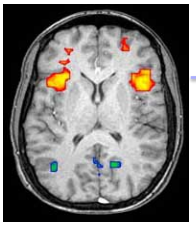
below average

"bottle" minus mean activation:



fMRI Prediction Task

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

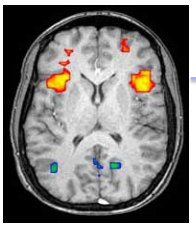


Classifier
(logistic regression, kNN, …)

~~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

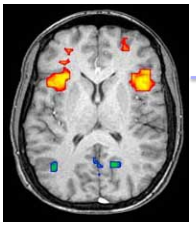


Classifier
(logistic regression, kNN, …)

~~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?



Classifier
(logistic regression, kNN, …)

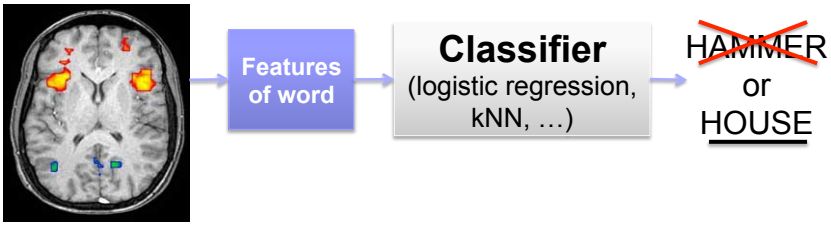
~~HAMMER~~
or
HOUSE

Semantic Features

Semantic feature values: "**celery**"	Semantic feature values: "**airplane**"
0.8368, eat	0.8673, ride
0.3461, taste	0.2891, see
0.3153, fill	0.2851, say
0.2430, see	0.1689, near
0.1145, clean	0.1228, open
0.0600, open	0.0883, hear
0.0586, smell	0.0771, run
0.0286, touch	0.0749, lift
…	…
…	…
0.0000, drive	0.0049, smell
0.0000, wear	0.0010, wear
0.0000, lift	0.0000, taste
0.0000, break	0.0000, rub
0.0000, ride	0.0000, manipulate

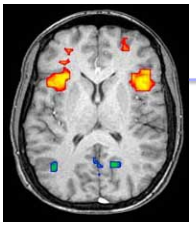
Zero-Shot Classification

- From training data, learn two mappings:
  - S: input image → semantic features
  - L: semantic features → word
- Can use "cheap" co-occurrence data to help learn L



fMRI Prediction Subtask

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



Features of word

Linear Regression – *review*

- Model:

- MLE: $\hat{\theta} = \arg\max_{\theta} \log p(D \mid \theta)$

- Minimizing RSS= least squares regression

Linear Regression – *review*

- Taking the gradient
  - Reformulate objective
  - Set gradient = 0

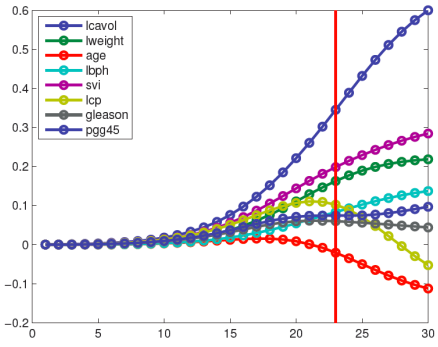
Ridge Regression

- Ameliorating issues with overfitting:
- New objective:
  - Reformulate:
  - Set gradient = 0

Ridge Regression

- Solution is indexed by the regularization parameter λ
- Larger λ
- Smaller λ
- As λ → 0
- As λ →∞

Ridge Coefficient Path



From
Kevin Murphy
textbook

- Typical approach: select λ using cross validation

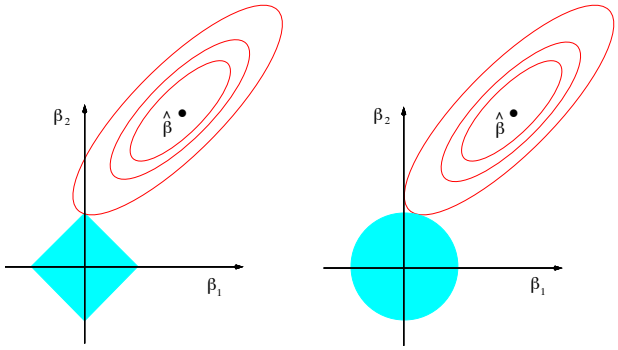
Variable Selection

- 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
  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit
- Try new penalty: Penalize non-zero weights
  - Penalty:
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param λ

LASSO Regression

- **LASSO:** least absolute shrinkage and selection operator
- New objective:

Geometric Intuition for Sparsity



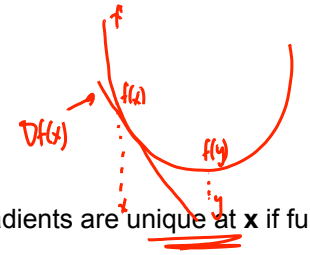
Lasso Ridge Regression

Soft Threshholding

- To see why LASSO results in sparse solutions, look at conditions that must hold at optimum
- L1 penalty $||\beta||_1$ is not differentiable whenever $\beta_j = 0$
- Look at subgradient…

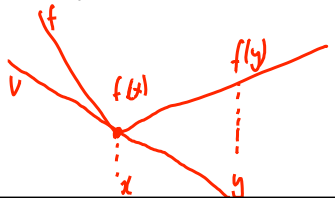
Subgradients of Convex Functions

- Gradients lower bound convex functions:



$f(y) \geq f(x) + \nabla f(x)\,(y-x)$

- Gradients are unique at **x** if function differentiable at **x**
- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function:



$v \in \partial f(x)$ subgradient if $f(y) \geq f(x) + v \cdot (y-x)$

Soft Threshholding

- Gradient of RSS term:
- Subgradient of full objective:

Soft Threshholding

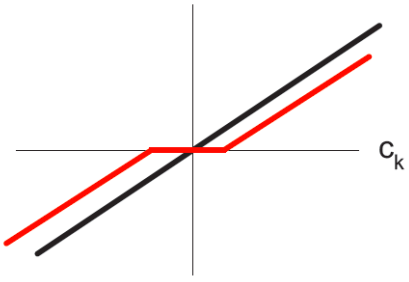
- Set subgradient = 0:

$$\partial_{\beta_j} F(\beta) = \begin{cases} a_j\beta_j - c_j - \lambda & \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\ a_j\beta_j - c_j + \lambda & \beta_j > 0 \end{cases}$$

- The value of $c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta'_{-j} x^i_{-j})$ constrains β_j

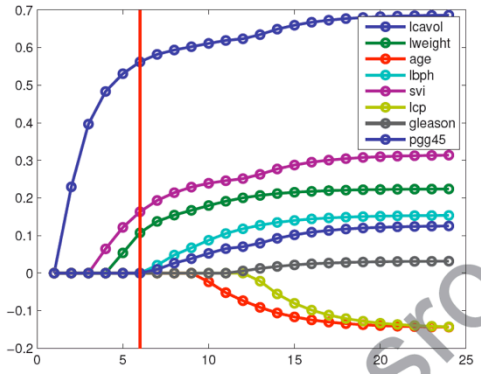
Soft Threshholding

$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$



From Kevin Murphy textbook

LASSO Coefficient Path

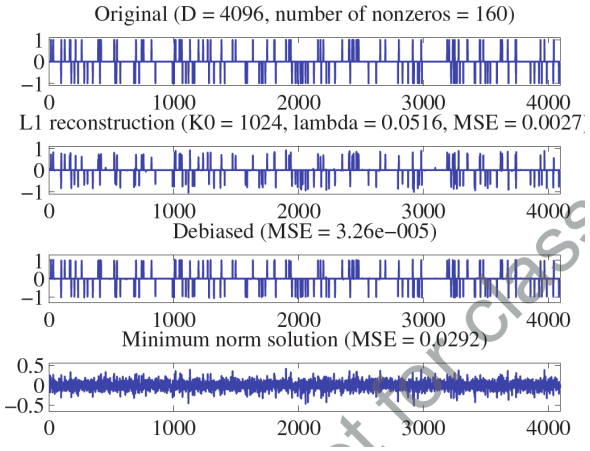


From
Kevin Murphy
textbook

LASSO Example

Term	Least Squares	Ridge	Lasso
Intercept	2.465	2.452	2.468
lcavol	0.680	0.420	0.533
lweight	0.263	0.238	0.169
age	−0.141	−0.046	
lbph	0.210	0.162	0.002
svi	0.305	0.227	0.094
lcp	−0.288	0.000	
gleason	−0.021	0.040	
pgg45	0.267	0.133	

Debiasing



From Kevin Murphy textbook

LASSO Algorithms

- Standard convex optimizer
- Least angle regression (LAR)
  - Efron et al 2004
  - Computes entire path of solutions
  - State-of-the-art until 2008
- Pathwise coordinate descent – new
- More on these algorithms next time…

Acknowledgements

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