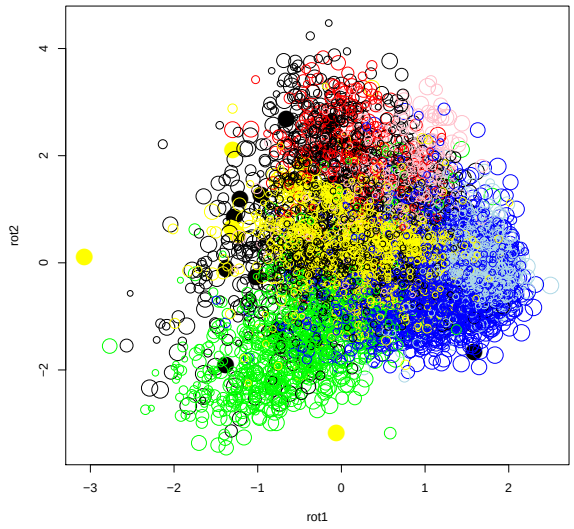


Linear methods for large data

Dean P. Foster

talk title

Better title

> "**Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions**"

- Unfortuantely it not written by me.
- by N. Halko, P. G. Martinsson, and J. A. Tropp.
- It is my current favorite paper.
- Fun probability theory: all about normal distributions.
- Fast matrix methods (SVDs, regressions, etc.)

talk title

Better title

> "**Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions**"

- Unfortuantely it not written by me.
- by N. Halko, P. G. Martinsson, and J. A. Tropp.
- It is my current favorite paper.
- Fun probability theory: all about normal distributions.
- Fast matrix methods (SVDs, regressions, etc.)
- How can we use these methods in statistics?

Basic method

problem Find a low rank approximation to a $n \times m$ matrix M.

solution Find a $n \times k$ matrix A such that $M \approx AA^\top M$

Basic method

problem Find a low rank approximation to a $n \times m$ matrix M.

solution Find a $n \times k$ matrix A such that $M \approx AA^\top M$

Construction A is constructed by:

1. create a random $m \times k$ matrix Ω (iid normals)
2. compute $M\Omega$
3. Compute thin SVD of result: $UDV^\top = M\Omega$
4. $A = U$

FAST MATRIX REGRESSIONS

Using random methods for regression

Toy problem: $p \ll n$:

Using random methods for regression

Toy problem: $p \ll n$:

- Solving least squares:
  - Generates provably accurate results.
  - Instead of np^2 time, it runs in np time.
  - This is fast! (I.e. as fast as reading the data.)

Using random methods for regression

Toy problem: $p \ll n$:

- Solving least squares:
  - Generates provably accurate results.
  - Instead of np^2 time, it runs in np time.
  - This is fast! (I.e. as fast as reading the data.)

- But we statisticians should be unimpressed.

Using random methods for regression

Toy problem: $p \ll n$:

- Solving least squares:
  - Generates provably accurate results.
  - Instead of np^2 time, it runs in np time.
  - This is fast! (I.e. as fast as reading the data.)
- But we statisticians should be unimpressed.
- Alternative fast (but stupid) method:
  - Do least squares on a subsample of size n/p
  - Runs in time np.
  - Same accuracy as the fast methods.

A fast regression algorithm

- Create "subsample" $\hat{X} \equiv AA^\top X$
- Estimate $X^\top X$:

$$\begin{aligned} X^\top X &\approx \hat{X}^\top \hat{X} \\ &= XAA^\top AA^\top X \\ &= (A^\top X)^\top (A^\top X) \end{aligned}$$

- Estimate $\hat{\beta} = (\hat{X}^\top \hat{X})^{-1} X^\top Y$

Fast and accurate

- As fast as only reading the data (np time)
- As accurate as using all the data (2013)
- Fast matrix multiply does the subsampling
- Fast PCAs regression
  - Subsample columns almost works
  - Fast matrix multiply fixes the “almost” (2013)
  - Aside: yields fast ridge regression also (2010)

Fast and accurate

- As fast as only reading the data (np time)
- As accurate as using all the data (2013)
- Fast matrix multiply does the subsampling
- Fast PCAs regression
  - Subsample columns almost works
  - Fast matrix multiply fixes the “almost” (2013)
  - Aside: yields fast ridge regression also (2010)

Ideas I’ll discuss in detail talk:

- streaming variable selection.
- Fast CCAs.
- Fast HMMs.
- Fast clustering.

(1) VIF regression

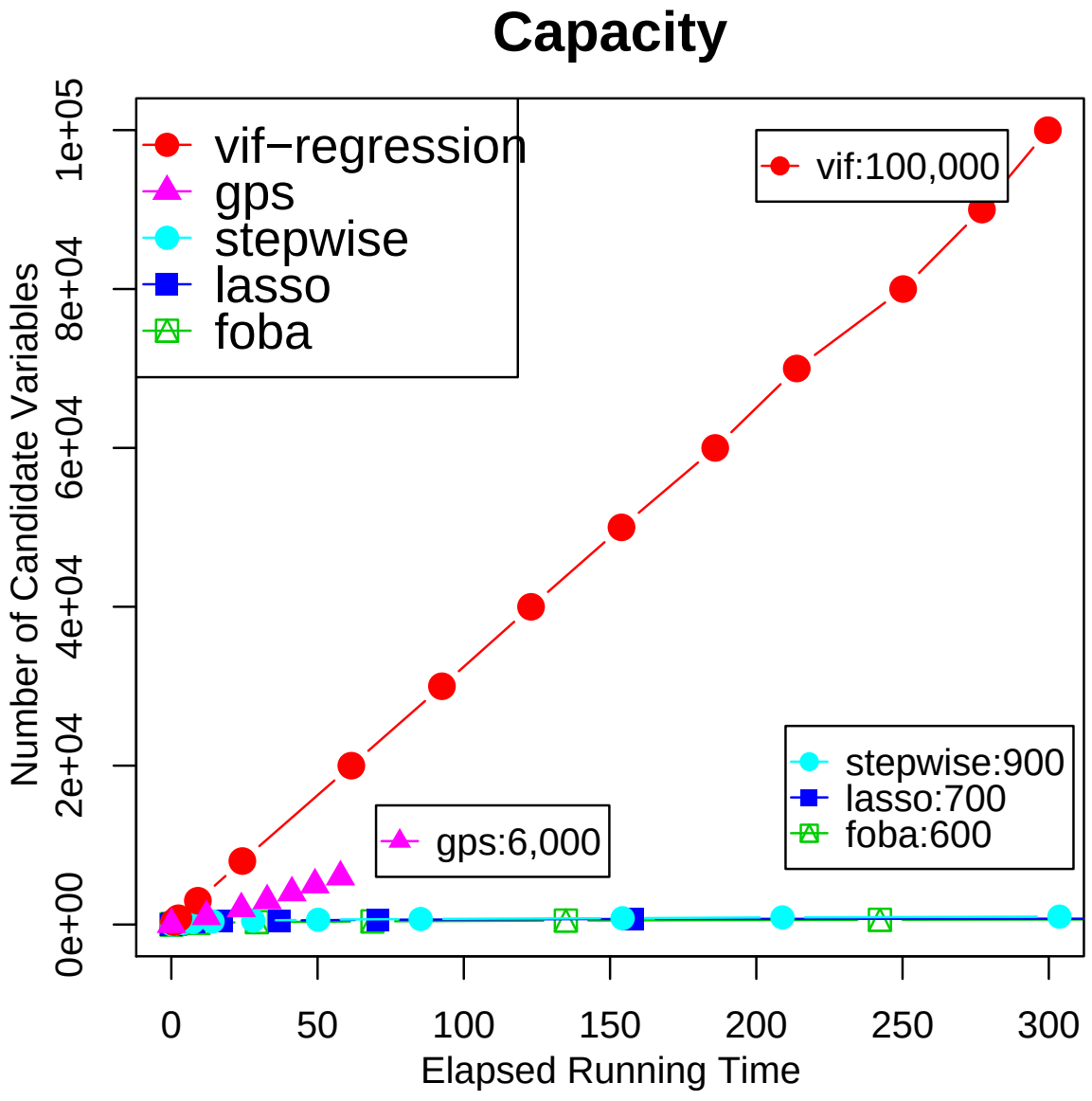
VIF regression

- Basic method: Stream over the features, trying them in order
- provides mFDR protection (2006)
- Instead of orthogonalizing each new X, only approximately orthogonalize it. (2011)
  - Can be done via sampling
  - Can be done use fast matrix methods

VIF regression

- Basic method: Stream over the features, trying them in order
- provides mFDR protection (2006)
- Instead of orthogonalizing each new X, only approximately orthogonalize it. (2011)
  - Can be done via sampling
  - Can be done use fast matrix methods
- For sub-modular problems, this will generate almost as good an estimator as best subsets. (2013)
- For kertotic X's, random Hadamard matrixes work better at a $log(p)^2$ cpu cost.

VIF speed comparison



(2) CCA for Semi-supervised data

CCA: Usual data table for data mining

$$\left[\begin{array}{c} Y \\ (n \times 1) \end{array}\right] \left[\begin{array}{c} X \\ (n \times p) \end{array}\right]$$

with $p \gg n$

With unlabeled data

m rows of unlabeled data:

$$\begin{bmatrix} Y \\ \\ n \times 1 \end{bmatrix} \qquad \begin{bmatrix} X \\ \\ (n+m) \times p \\ \\ \\ \\ \end{bmatrix}$$

With alternative X's

m rows of unlabeled data, and two sets of equally useful X's:

$$\begin{bmatrix} Y \\ \\ n \times 1 \end{bmatrix} \quad \begin{bmatrix} X \\ \\ (n+m) \times p \end{bmatrix} \quad \begin{bmatrix} Z \\ \\ (n+m) \times p \end{bmatrix}$$

With: $m \gg n$

Examples

- Named entity recognition
  - Y = person / place
  - X = name itself
  - Z = words before target
- Topic identification (medline)
  - Y = topic
  - X = abstract
  - Z = text
- Sitcom speaker identification:
  - Y = which character is speaking
  - X = video
  - Z = sound
- We will call these the multi-view setup

What if we run CCA on X and Z?

CCA = canonical correlation analysis

- Find the directions that are most highly correlated
- Close to PCA (principal components analysis)

What if we run CCA on X and Z?

CCA = canonical correlation analysis

- Find the directions that are most highly correlated
- Close to PCA (principal components analysis)

- Numerically an Eigen-value problem
- So, we can use our new fast algorithms here (Hsu, Kakade, Zhang 2012)

The Main Result

Theorem (Foster and Kakade, '06)

Let $\hat{\beta}$ be the Ridge regression estimator with weights induced by the CCA. Then under the multi-view assumption

$$\mathit{Risk}(\hat{\beta}) \leq \left(5\alpha + \frac{\sum \lambda_i^2}{n}\right) \sigma^2$$

The Main Result

Theorem (Foster and Kakade, '06)

Let $\hat{\beta}$ be the Ridge regression estimator with weights induced by the CCA. Then under the multi-view assumption

$$Risk(\hat{\beta}) \leq \left(5\alpha + \frac{\sum \lambda_i^2}{n}\right) \sigma^2$$

Estimator is least squares plus a penalty of:

$$\sum_i \frac{1-\lambda_i}{\lambda_i} \beta_i^2$$

Where λ_i's are the correlations

The Main Result

Theorem (Foster and Kakade, '06)

Let $\hat{\beta}$ be the Ridge regression estimator with weights induced by the CCA. Then under the multi-view assumption

$$Risk(\hat{\beta}) \leq \left(5\alpha + \frac{\sum \lambda_i^2}{n}\right) \sigma^2$$

Multivew property α is the multiview property:

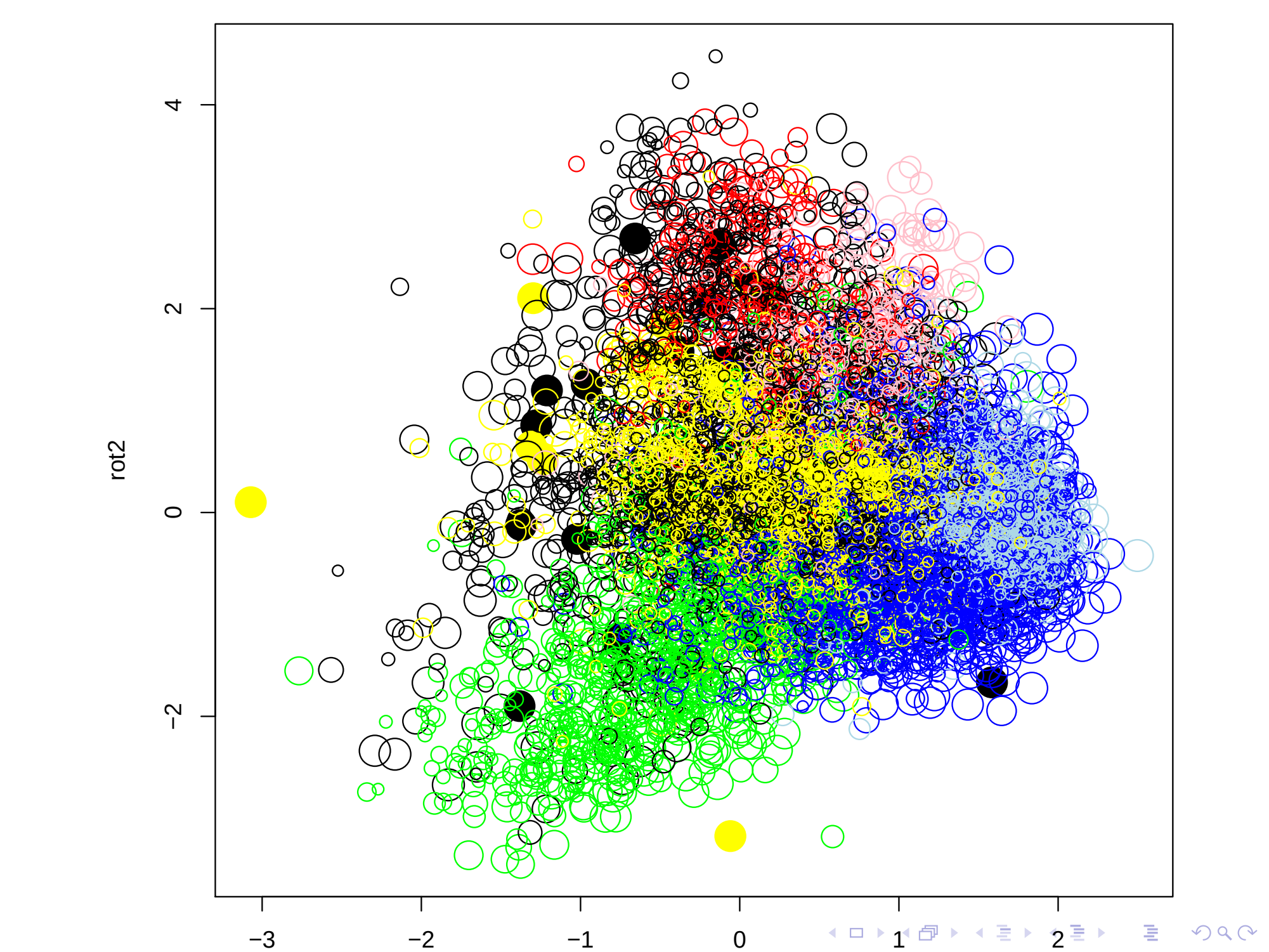
$$\begin{aligned} \sigma_x^2 &\leq \sigma_{x,z}^2(1+\alpha) \\ \sigma_z^2 &\leq \sigma_{x,z}^2(1+\alpha) \end{aligned}$$

- 5α is the bias
- $\frac{\sum \lambda_i^2}{n}$ is variance

HMMs intro: Two views of words

HMMs intro: Two views of words

- Past and future are conditionally independent for a Markov process
- Treat them as two views
- For HMM, we have 3 views, past, future, current observation
- We call the result “eigenwords”
- Predict POS for a word


rot2
4
2
0
−2
−3
−2
−1
0
1
2

Key

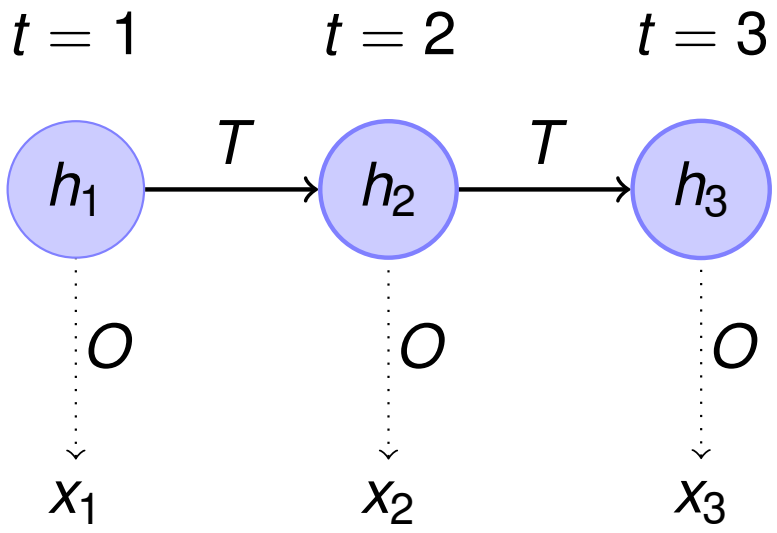
Colors:

- nouns = Blue (dark = NN1, light = NN2)
- verbs = red (dark = VV1, light = VV2)
- adj = green
- unk = yellow
- black = all else

Size = 1/Zipf order, top 50 are solid, rest are open.

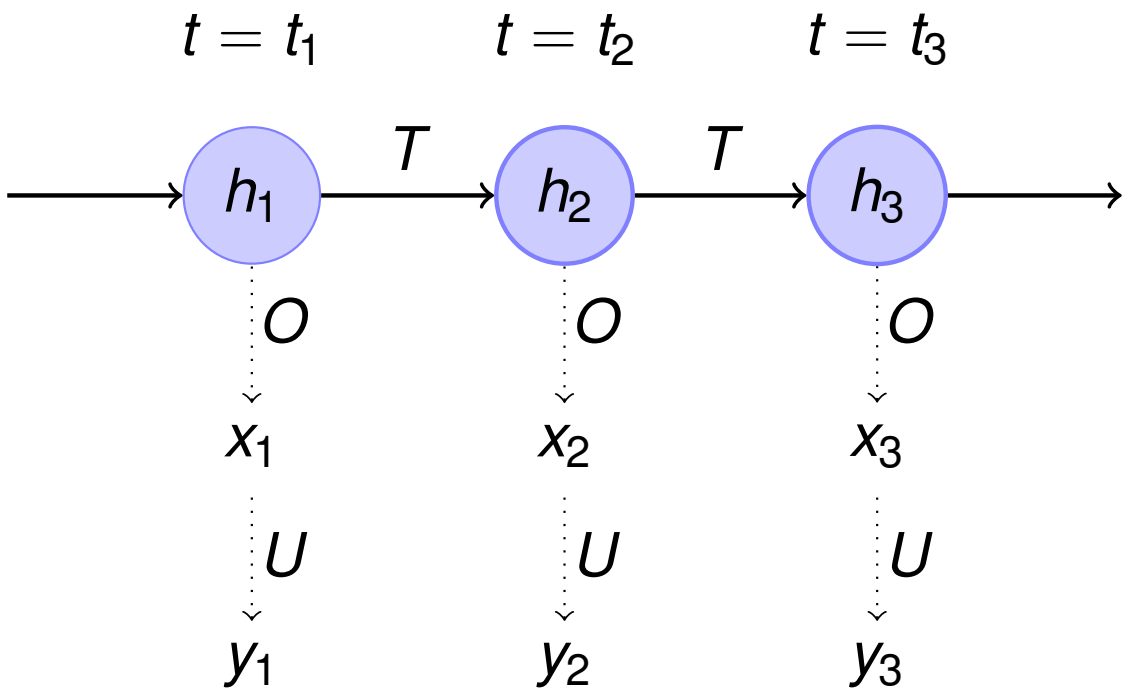
(3) Tri-linear: HMMs

The Hidden Markov Model



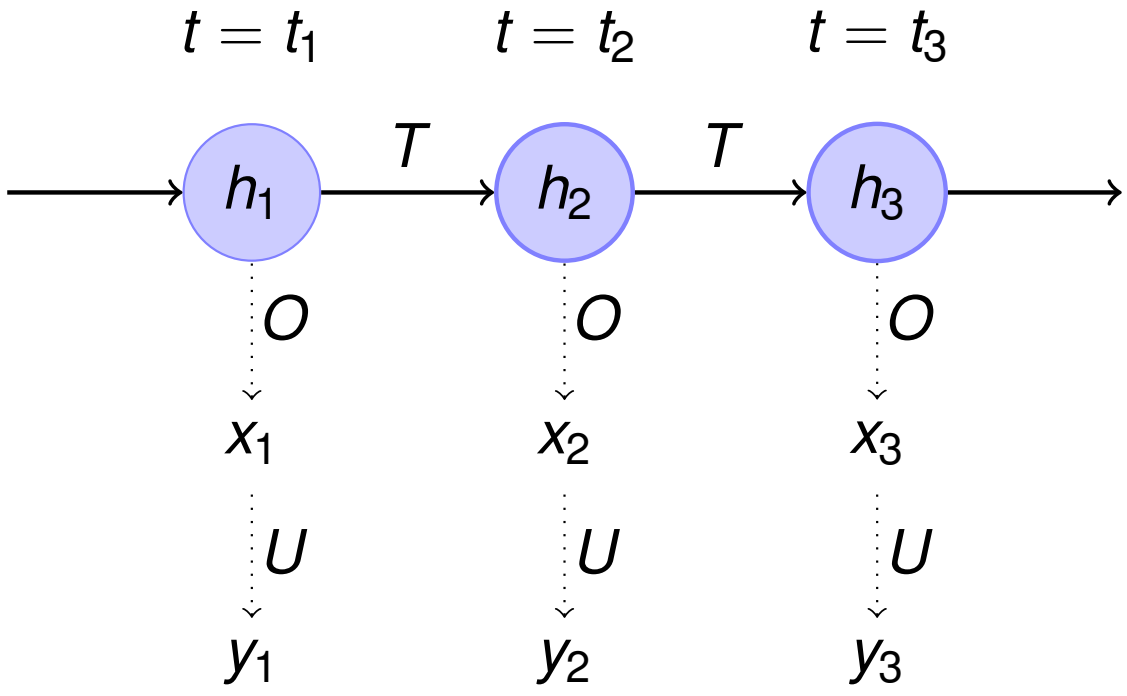
HMM with states h_1, h_2, and h_3 which generate observations x_1, x_2, and x_3.

Reduced Dimension Hidden Markov Model



The Y's are our eigenwords.

Reduced Dimension Hidden Markov Model



The Y's are our eigenwords.

$$\Pr(x_t, \ldots, x_1) = 1^T T \mathrm{diag}(OU^\top y_t) \cdots T \mathrm{diag}(OU^\top y_1) \pi$$

HMM theorem

Theorem (with Rodu, Ungar)

Let X_t be generated by an $m \geq 2$ state HMM. Suppose we are given a U which has the property that range$(O) \subset$ range(U) and $|U_{ij}| \leq 1$. Using N independent triples, we have

$$N \geq \frac{128 m^2 (2t+3)^2}{\epsilon^2 \, \Lambda^2 \sigma_m^4} \log\left(\frac{2m}{\delta}\right) \cdot \overbrace{\frac{\epsilon^2/(2t+3)^2}{(\sqrt[2t+3]{1+\epsilon}-1)^2}}^{\approx 1}$$

implies that

$$1 - \epsilon \leq \left| \frac{\widehat{\Pr}(x_1, \ldots, x_t)}{\Pr(x_1, \ldots, x_t)} \right| \leq 1 + \epsilon$$

holds with probability at least $1 - \delta$.

Results on ConLL task

- Results on 2 NLP sequence labeling problems: NER (CoNLL '03 shared task) and Chunking (CoNLL '00 shared task).
- Trained on $\sim$ 65 million tokens of unlabeled text in a few hours!

Relative reduction in error over state-of-the-art:

Embedding/Model	NER	Chunking
C&W	15.0%	18.8%
HLBL	19.5%	20.2%
Brown	12.1%	18.7%
Ando+Zhang	5.6%	14.6%

Extending to trees

Theorem (with Rodu, Ungar, Dhillon, Collins)

Same as before–but for dependency parse trees.

Ran Dependency parsing on Penn Treebank

- Raw MST Parser is 91.8% accurate
- Adding eigenwords: 2.6% error reduction
- eigenwords plus Re-ranking: 7.3% error reduction

Neural data: raw

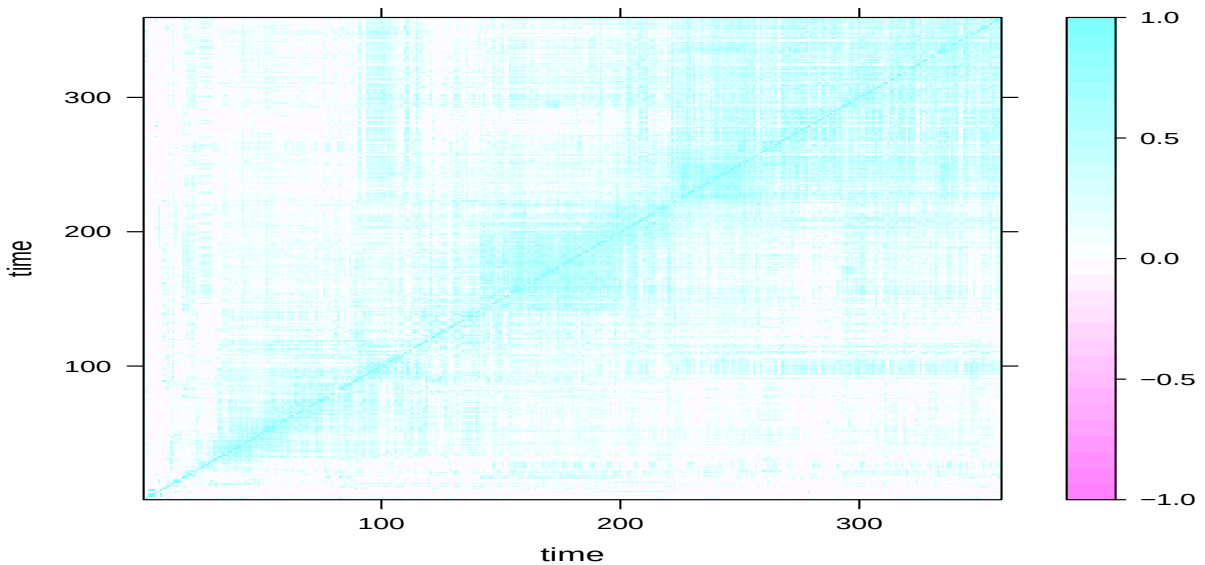


Figure 1: Correlation of raw observations, binned at 10 second bins

Neural data: reduced dimension

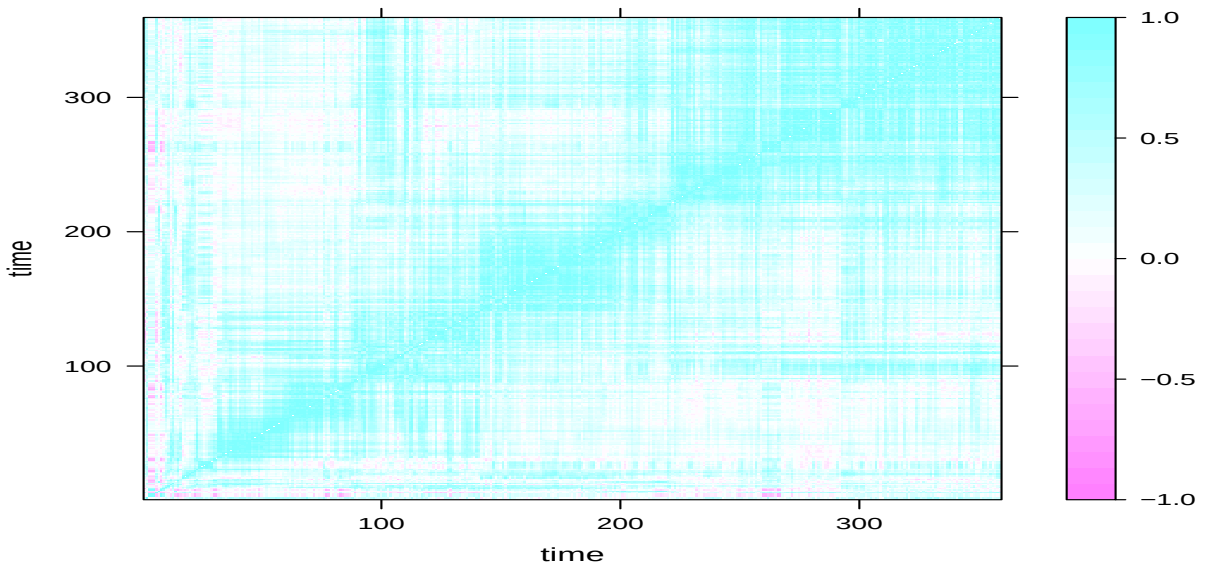


Figure 2: Correlations among reduced dimensional observations k=10

Neural data: state estimate

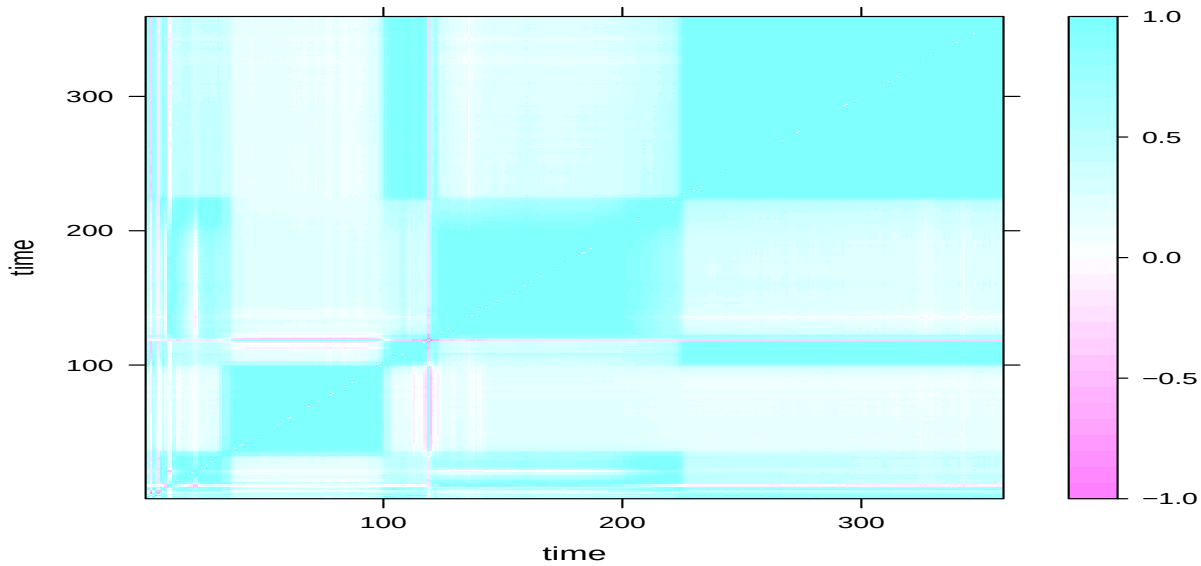
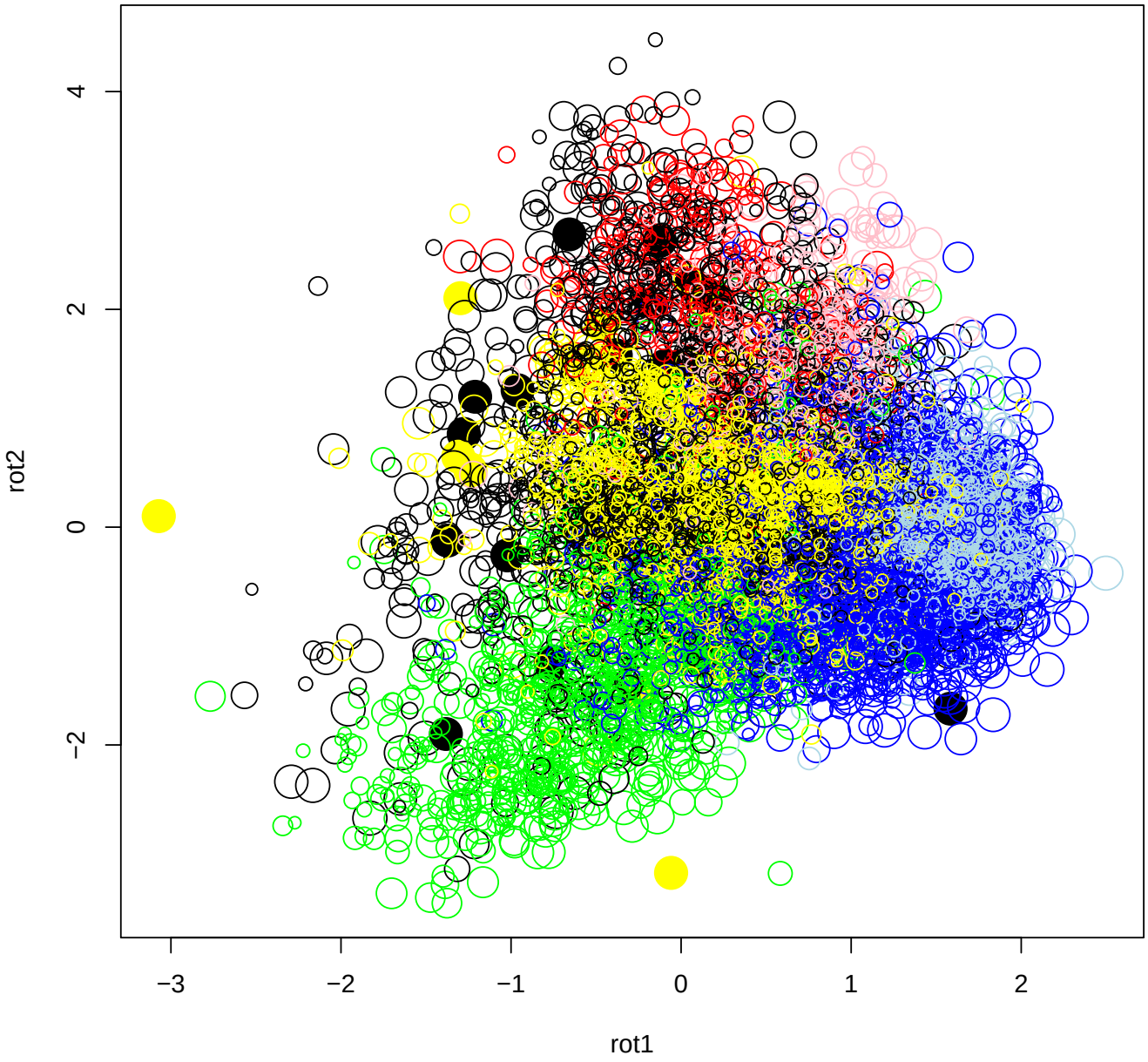


Figure 3: Correlations among the states of the system as time progresses k=10

(4) TETRA-LINEAR: Clustering

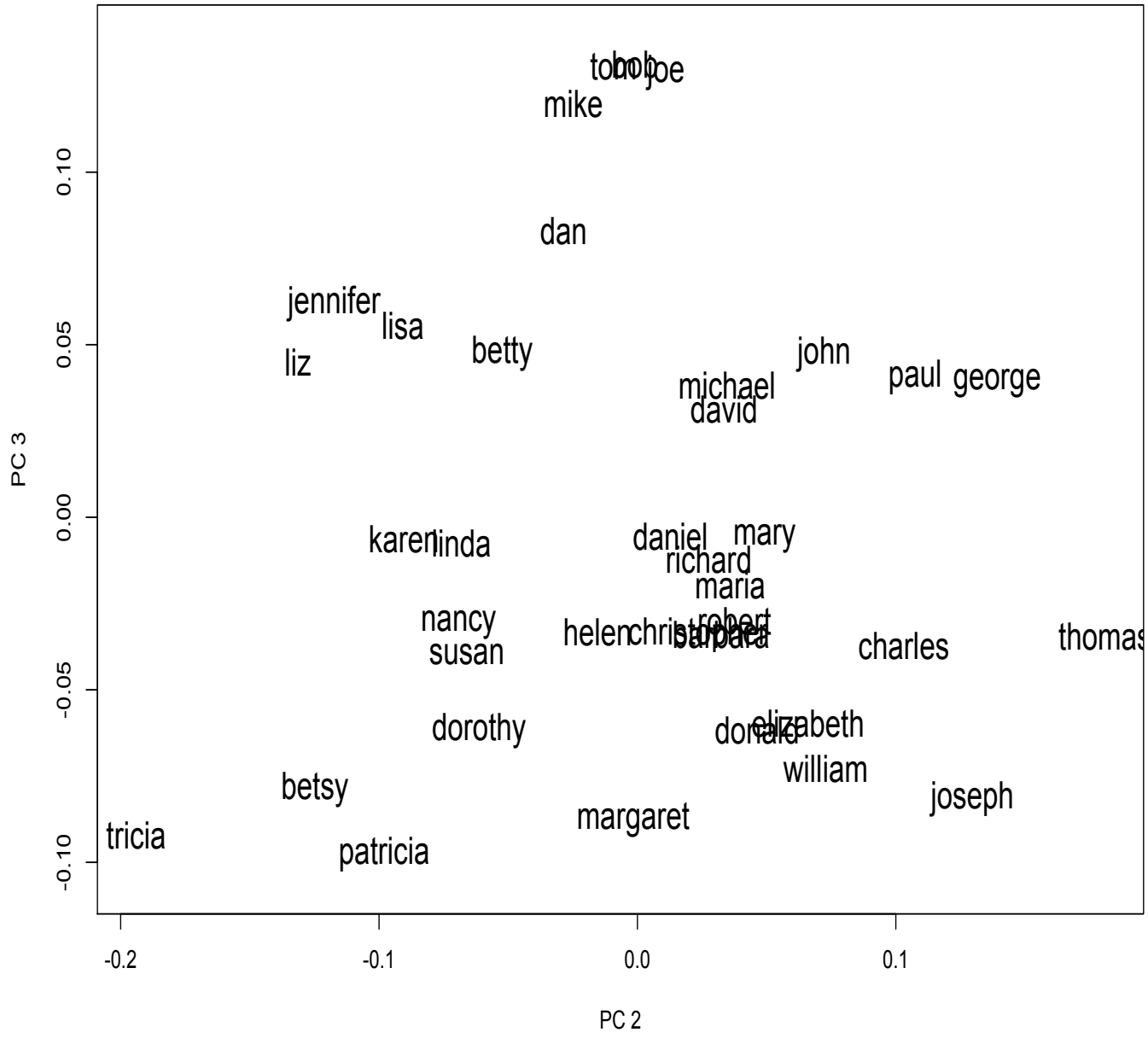


If you rotate this, you will see there are “pointy” directions

Clustering theorem

Theorem (with Hsu, Kakade, Liu, Anima, NIPS 2012)

Maximizing $E(\mu^\top X)^4$ will find the natural coordinate system for LDA.


tom
bob
joe
mike
dan
jennifer
lisa
liz
betty
john
paul
george
michael
david
karen
linda
daniel
mary
richard
maria
nancy
susan
helen
barbara
charles
thomas
dorothy
donald
elizabeth
william
betsy
joseph
margaret
tricia
patricia
PC 3
0.10
0.05
0.00
-0.05
-0.10
-0.2
-0.1
0.0
0.1
PC 2

COAUTHORS

Coauthors

		U Penn	Elsewhere
Stat	faculty	Edward George Robert Stine	Tong Zhang (Rutgers) Daniel Hsu (MSR) Sham Kakade (MSR)
	students	Dongyu Lin (ATT) Jordan Rodu Kory Johnson Yichao Lu	
CS	faculty	Lyle Ungar	Michael Collins (Columbia)
	students	Parmaveer Dhillon Jing Zhou (real world)	Shay Cohen (Columbia) Karl Stratos (Columbia)

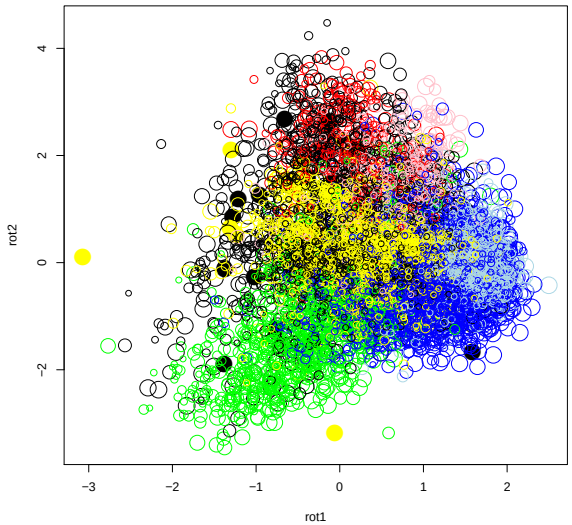
Coauthors

		U Penn	Elsewhere
Stat	faculty	Edward George Robert Stine	Tong Zhang (Rutgers) Daniel Hsu (MSR) Sham Kakade (MSR)
	students	Dongyu Lin (ATT) Jordan Rodu Kory Johnson Yichao Lu	
CS	faculty	Lyle Ungar	Michael Collins (Columbia)
	students	Parmaveer Dhillon Jing Zhou (real world)	Shay Cohen (Columbia) Karl Stratos (Columbia)

’94 ’00 ’04 ’06 ’08 ’10 ’11 ’12 ’13 ’13 ’12 ’12 ’12 ’12 ’12

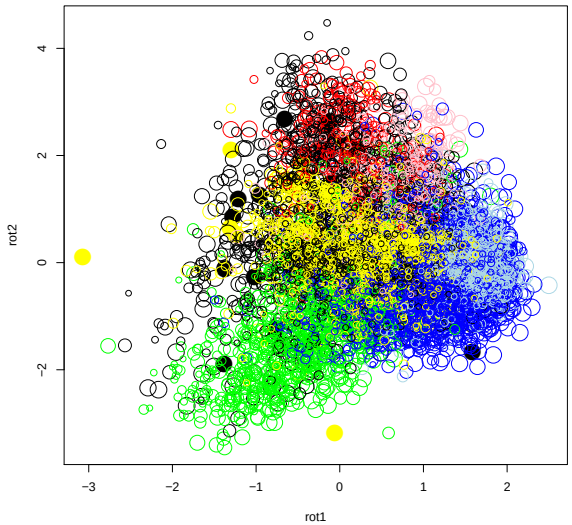
Conclusions

- These new fast matrix methods are easy to program.
- Generate statistically useful results.
- Room for interesting new probability theory.



Conclusions

- These new fast matrix methods are easy to program.
- Generate statistically useful results.
- Room for interesting new probability theory.



Thanks!

Fast Regressions

Theorem (with Yichao Lu, Parmaveer Dhillion, Lyle Ungar)

If $n > p^3$, then the algorithm defined by:

- *Let $m = \sqrt{n}$*
- *Pull out a subsample of size m from X's and call it Z.*
- *Let $\hat{\beta} \equiv (Z^\top Z)^{-1} X^\top Y$*

then the CPU time is $O(np)$ and accuracy is as good as the usually estimator.

Fast Principle components regressions

Theorem (with Yichao Lu, Parmaveer Dhillion, Lyle Ungar)

If $p > n$, then using a SRHT on the columns followed by regression will take $O(np\log(p))$ time and lose a constant factor on the statistical accuracy.

PCR is close to ridge regression

Theorem (with Sham Kakade, Parmaveer Dhillion, Lyle Ungar)

A ridge regression can be quickly approximated by regressing on the top principle components. In particular, for a ridge parameter λ using components with singular values larger than λ will be within a factor of 4 of the ridge estimator on statistical accuracy.

mFDR for streaming feature selection

Let $W(j)$ be the "alpha wealth" at time j. Then for a series of p-values p_j, we can define:

$$W(j) - W(j-1) = \begin{cases} \omega & \text{if } p_j \leq \alpha_j \,, \\ -\alpha_j/(1-\alpha_j) & \text{if } p_j > \alpha_j \,. \end{cases} \tag{1}$$

Theorem

(Foster and Stine, 2006) An alpha-investing rule governed by (1) with initial alpha-wealth $W(0) \leq \alpha\,\eta$ and pay-out $\omega \leq \alpha$ controls $mFDR_\eta$ at level α.

VIF regression

Theorem

(Foster, Dongyu Lin, 2011) VIF regression approximates a streaming feature selection method with speed $O(np)$.

Submodular

Theorem

(Foster, Johnson, Stine, 2013) If the R-squared in a regression is submodular (aka subadditive) then a streaming feature selection algorithm will find an estimator whose out risk is within a factor of $e/(e-1)$ of the optimal risk.