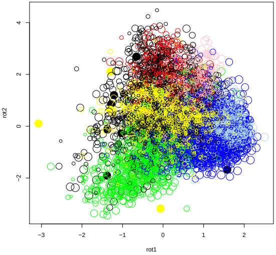


Linear methods for large data

Dean Foster

Amazon

This is a talk about some other people's paper

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

- by Halko, Martinsson, and Tropp.
- It is my current favorite paper.
- Today, I'll be applying it to a linear regression.

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: (a la Mahoney)
  - 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: (a la Mahoney)
  - 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 should be unimpressed.

Using random methods for regression

Toy problem: $p \ll n$:

- Solving least squares: (a la Mahoney)
  - 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 should be unimpressed.
- Alternative fast (but stupid) method:
  - Do least squares on a sub-sample of size n/p
  - Runs in time np.
  - Same accuracy as the fast methods.

A better fast regression

- Create "sub-sample" $\hat{X} \equiv AA^\top X$ and estimate

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

A better fast regression

- Create "sub-sample" $\hat{X} \equiv AA^\top X$ and estimate

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

- (Mahoney also subsampled Y and hence lost accuracy.)
- New method is fast and accurate (NIPS 2013a)
- What about $p \gg n$?
  - Sub-sample columns almost works
  - Fast matrix approximation fixes the "almost" (NIPS 2013b)
  - Aside: yields fast ridge regression also (JMLR 2013)
- What about $p \approx n$?
  - needs stochastic gradient also. (UAI 2014)

Applications of fast matrix methods:

1. Least squares regression (we just finished).
2. Sparse Linear Regression (today's talk).
3. Fast CCAs.
4. Fast HMMs.
5. Fast parsing.
6. Fast clustering.

Problem statement: L0 regression

- Problem:

$$Y = X\beta + \sigma Z$$

  using prediction risk $E|\mathbf{X}\beta - \mathbf{X}\hat{\beta}|_2^2$.
- Target risk is $q\sigma^2$ for the correct set of q variables.

Risk Inflation

Theorem (Foster and George, 1994)

For any orthogonal X matrix, using a penality of $2\log(p)$ yields a risk that is within a $2\log(p)$ factor of the target.

Risk Inflation

Theorem (Foster and George, 1994)

For any orthogonal X matrix, using a penality of $2\log(p)$ yields a risk that is within a $2\log(p)$ factor of the target.

- Also proven by Donoho and Johnstone in the same year.
- The bound is tight.
- The same bound works for Lasso.

Risk Inflation

Theorem (Foster and George, 1994)

For any ~~orthogonal~~ X matrix, using a penality of $2\log(p)$ yields a risk that is within a 4 $\log(p)$ factor of the target.

Risk Inflation

Theorem (Foster and George, 1994)

For any ~~orthogonal~~ X matrix, using a penality of $2\log(p)$ yields a risk that is within a 4 $\log(p)$ factor of the target.

- This bound is also tight: I.e. there are design matrices for which any estimator does this badly.
- Lasso's risk inflation is infinite for bad X's

Machine learning = Statistics + computation

- Naive algorithm takes 2^p time
- Greedy runs fast (takes np^2 time)
- Called stepwise regression
- How well does it perform?

A success for stepwise regression

Theorem (Natarajan 1995)

Stepwise regression will have a prediction accuracy of at most twice optimal using at most $\approx 18|X^+|_2^2 q$ *variables.*

A success for stepwise regression

Theorem (Natarajan 1995)

Stepwise regression will have a prediction accuracy of at most twice optimal using at most $\approx 18|X^+|_2^2 q$ *variables.*

- The $|X^+|_2$ is a measure of co-linearity.
- The risk inflation is a disaster.
- Suggests three goals:
  - sparse answers
  - accuracy
  - speed

L0 regression is hard

Theorem (Zhang, Wainwright, Jordan 2014)

There exists an design matrix X such that no polynomial time algorithm which outputs q variables achieves a risk better than

$$R(\hat{\theta}) \gtrsim \frac{1}{\gamma^2(X)} \sigma^2 q \log(p).$$

Where γ is the RE, a measure of co-linearity.

L0 regression is hard

Theorem (Zhang, Wainwright, Jordan 2014)

There exists an design matrix X such that no polynomial time algorithm which outputs q variables achieves a risk better than

$$R(\hat{\theta}) \gtrsim \frac{1}{\gamma^2(X)} \sigma^2 q \log(p).$$

Where γ is the RE, a measure of co-linearity.

- Actual statement is much more complex and involves a version of the assumption that $P \neq NP$.

L0 regression is hard

Theorem (Zhang, Wainwright, Jordan 2014)

There exists an design matrix X such that no polynomial time algorithm which outputs q variables achieves a risk better than

$$R(\hat{\theta}) \gtrsim \frac{1}{\gamma^2(X)}\sigma^2 q \log(p).$$

Where γ is the RE, a measure of co-linearity.

- Note: No cheating on the dimension.
- What if we let it use $2q$ variables? Could we get good risk?

L0 regression is VERY hard

Theorem (Foster, Karloff, Thaler 2014)

No algorithm exists which achieves all three of the following goals:

- *Runs efficiently (i.e. in polynomial time)*
- *Runs accurately (i.e. risk inflation < p)*
- *Returns sparse answer (i.e.* $|\hat{\beta}|_0 \ll p$*)*

What to do?

- Hard problems exist
- So, assume the world is nice and we can get
  - a small model
  - with accurate prediction
  - that runs fast
- Called alpha investing

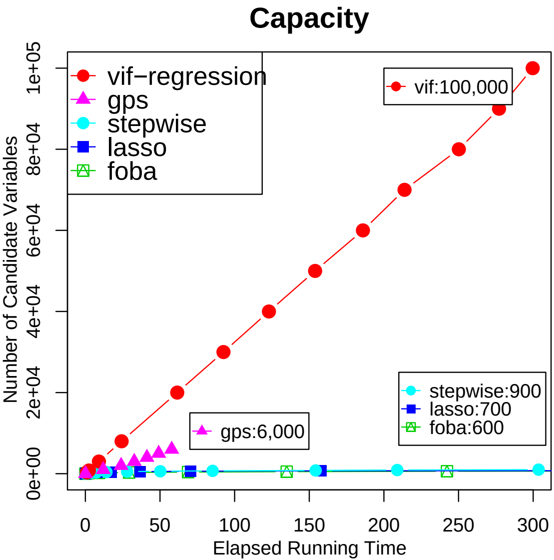
VIF regression

- Basic method: Stream over the features, trying them in order
- Even more gready than stepwise regression (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

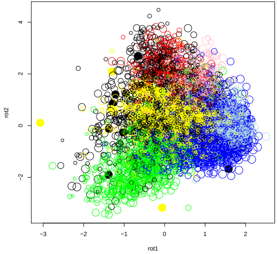
- Basic method: Stream over the features, trying them in order
- Even more gready than stepwise regression (2006)
- Instead of orthogonalizing each new X, only approximately orthogonalize it. (2011)
  - Can be done via sampling
  - Can be done use fast matrix methods
- Nice statistical properties:
  - For sub-modular problems, this will generate almost as good an estimator as best subsets. (2013)
  - provides mFDR protection (2008)

VIF speed comparison



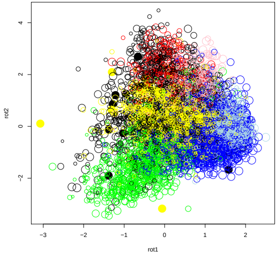
Conclusions

- These new fast matrix methods are easy to prove theorems about.
- They generate statistically useful results.
- So, read Halko, Martinsson, and Tropp!



Conclusions

- These new fast matrix methods are easy to prove theorems about.
- They generate statistically useful results.
- So, read Halko, Martinsson, and Tropp!



Thanks!