Sampling algorithms and core-sets for $L_p$ regression and applications

Michael W. Mahoney

Yahoo Research

(For more info, see: http://www.cs.yale.edu/homes/mmahoney)
In MANY applications (in statistical data analysis and scientific computation), one has n observations (values of a dependent variable y measured at values of an independent variable t):

\[ y_i = y(t_i), \ i = 1, \ldots, n \]

Model \( y(t) \) by a linear combination of d basis functions:

\[ y(t) \approx x_1 \phi_1(t) + \cdots + x_d \phi_d(t) \]

A is an \( n \times d \) “design matrix” with elements:

\[ A_{ij} = \phi_j(t_i) \]

In matrix-vector notation:

\[ y \approx Ax \]
Many applications of this!

- **Astronomy**: Predicting the orbit of the asteroid Ceres (in 1801!).
  
  Gauss (1809) -- see also Legendre (1805) and Adrain (1808).

  First application of “least squares optimization” and runs in $O(nd^2)$ time!

- **Bioinformatics**: Dimension reduction for classification of gene expression microarray data.

- **Medicine**: Inverse treatment planning and fast intensity-modulated radiation therapy.

- **Engineering**: Finite elements methods for solving Poisson, etc. equation.

- **Control theory**: Optimal design and control theory problems.

- **Economics**: Restricted maximum-likelihood estimation in econometrics.

- **Image Analysis**: Array signal and image processing.

- **Computer Science**: Computer vision, document and information retrieval.

- **Internet Analysis**: Filtering and de-noising of noisy internet data.

- **Data analysis**: Fit parameters of a biological, chemical, economic, social, internet, etc. model to experimental data.
Large Graphs and Data at Yahoo

Explicit: graphs and networks

- Web Graph
- Internet
- Yahoo! Photo Sharing (Flickr)
- Yahoo! 360 (Social network)

Implicit: transactions, email, messenger

- Yahoo! Search marketing
- Yahoo! mail
- Yahoo! messenger

Constructed: affinity between data points

- Yahoo! Music
- Yahoo! Movies
- Yahoo! Etc.
<table>
<thead>
<tr>
<th>search</th>
<th>$ bid</th>
<th>advertiser</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessory desk</td>
<td>.83</td>
<td>Office Max</td>
</tr>
<tr>
<td>alfred hitchcock</td>
<td>.01</td>
<td>VideoFlicks.com</td>
</tr>
<tr>
<td>educational software</td>
<td>.13</td>
<td>Buy.com</td>
</tr>
<tr>
<td>educational software</td>
<td>.4</td>
<td>eBay International AG</td>
</tr>
<tr>
<td>educational software</td>
<td>.17</td>
<td>OfficeMax</td>
</tr>
<tr>
<td>epson</td>
<td>.28</td>
<td>Buy.com</td>
</tr>
<tr>
<td>epson</td>
<td>.4</td>
<td>eBay International AG</td>
</tr>
<tr>
<td>fax</td>
<td>.13</td>
<td>Buy.com</td>
</tr>
<tr>
<td>fax</td>
<td>.4</td>
<td>eBay International AG</td>
</tr>
<tr>
<td>fax</td>
<td>.38</td>
<td>OfficeMax</td>
</tr>
<tr>
<td>game</td>
<td>.02</td>
<td>Net Business</td>
</tr>
<tr>
<td>game</td>
<td>.25</td>
<td>Buy.com</td>
</tr>
<tr>
<td>george harrison</td>
<td>.15</td>
<td>eBay International AG</td>
</tr>
<tr>
<td>george harrison</td>
<td>.05</td>
<td>MusicStack</td>
</tr>
<tr>
<td>george harrison</td>
<td>.01</td>
<td>VideoFlicks.com</td>
</tr>
<tr>
<td>jackson michael</td>
<td>.05</td>
<td>MusicStack</td>
</tr>
<tr>
<td>jackson michael</td>
<td>.01</td>
<td>VideoFlicks.com</td>
</tr>
</tbody>
</table>
Least-norm approximation problems

Recall a linear measurement model:

\[ y = Ax + \epsilon \]

\( y \) are the measurements
\( x \) is the unknown
\( \epsilon \) is an error process

A common optimization problem:

\[ \min ||Ax - b|| \]

\( A \in R^{n \times d}, \ n > d \)
\( b \in R^n \)
\( || \cdot || \) is a norm on \( R^n \)

Let \( y = b \),

- then \( x^* = \arg \min_x ||Ax - b|| \) is the “best” estimate of \( x \)
- then \( Ax^* \) is the point in \( R(A) \) “closest” to \( b \).
Norms of common interest

Let: \( r = Ax - b \in \mathbb{R}^n \) denote the vector of residuals.

**Least-squares approximation:**

\[
\text{minimize: } \|Ax - b\|_2^2 = r_1^2 + r_2^2 + \cdots + r_n^2
\]

**Chebyshev or mini-max approximation:**

\[
\text{minimize: } \|Ax - b\|_\infty = \max\{|r_1|, \ldots, |r_n|\}
\]

**Sum of absolute residuals approximation:**

\[
\text{minimize: } \|Ax - b\|_1 = |r_1| + |r_2| + \cdots + |r_n|
\]
Recall the **Lp norm** for $z \in \mathbb{R}^n$:

$$
\|z\|_p = \left( \sum_{i=1}^{n} |z_i|^p \right)^{1/p}, \quad p \in [1, \infty)
$$

$$
\|z\|_{\infty} = \max_i |z_i|
$$

$$
\|z\|_2 = \sum_i z_i^2 = z^T z
$$

**Some inequality relationships include:**

$$
\frac{1}{\sqrt{n}} \|z\|_2 \leq \|z\|_{\infty} \leq \|z\|_2 \leq \|z\|_1 \leq \sqrt{n} \|z\|_2
$$
We are interested in over-constrained Lp regression problems, \( n \gg d \).

Typically, there is no \( x \) such that \( Ax = b \).

Want to find the "best" \( x \) such that \( Ax \approx b \).

Lp regression problems are convex programs (or better!).

There exist poly-time algorithms.

We want to solve them faster!
Solution to Lp regression

Lp regression can be cast as a **convex program** for all \( p \in [1, \infty] \).

For \( p=1 \), **Sum of absolute residuals** approximation (minimize \( \|Ax-b\|_1 \)):

- Cast as an LP: \( \text{minimize } 1^T t \)
  - such that \( -t \leq Ax - b \leq t \)

For \( p=\infty \), **Chebyshev or mini-max** approximation (minimize \( \|Ax-b\|_\infty \)):

- Cast as an LP: \( \text{minimize } t \)
  - such that \( -t1 \leq Ax - b \leq t1 \)

For \( p=2 \), **Least-squares** approximation (minimize \( \|Ax-b\|_2 \)):

- solution satisfies normal equations: \( A^T Ax = A^T b \)
- \( x^* = (A^T A)^{-1} A^T b \), if \( \text{rank}(A) = n \)
Solution to L2 regression

**Cholesky Decomposition:**

If $A$ is full rank and well-conditioned,
decompose $A^TA = R^TR$, where $R$ is upper triangular, and
solve the normal equations: $R^Tx = A^Tb$.

**QR Decomposition:**

Slower but numerically stable, esp. if $A$ is rank-deficient.
Write $A=QR$, and solve $Rx = Q^Tb$.

**Singular Value Decomposition:**

Most expensive, but best if $A$ is very ill-conditioned.
Write $A=U\Sigma V^T$, in which case: $x_{\text{opt}} = A^*b = V\Sigma^{-1}U^Tb$.

Complexity is $O(nd^2)$ for all of these, but constant factors differ.

$$
\mathcal{Z}_2 = \min_{x \in \mathbb{R}^d} ||b - Ax||_2 \\
= ||b - A\hat{x}||_2
$$

Projection of $b$ on the subspace spanned by the columns of $A$

$$
\mathcal{Z}_2^2 = ||b||_2^2 - ||AA^+b||_2^2 \\
\hat{x} = A^+b
$$
Pseudoinverse of $A$
Questions ...

\[ Z_p = \min_{x \in \mathbb{R}^d} |b - Ax|_p = |b - A\hat{x}|_p \]

**Approximation algorithms:**

*Can we approximately solve general \( L_p \) regression qualitatively faster than existing “exact” methods?*

**Core-sets (or induced sub-problems):**

*Can we find a small set of constraints s.t. solving the \( L_p \) regression on those constraints gives an approximation?*

**Generalization (for machine learning):**

*Does the core-set or approximate answer have similar generalization properties to the full problem or exact answer? (Still open!)*
# Overview of Five Lp Regression Algorithms

| Alg. 1 | Sampling (core-set) | p=2 | (1+ε)-approx | O(nd²) | Drineas, Mahoney, Muthukrishnan (SODA06) |
| Alg. 2 | Projection | p=2 | (1+ε)-approx | O(nd²) | “obvious“ |
| Alg. 3 | Projection | p=2 | (1+ε)-approx | o(nd²) | Sarlos (FOCS06) |
| Alg. 4 | Sampling | p=2 | (1+ε)-approx | o(nd²) | DMMS07 |
| Alg. 5 | Sampling (core-set) | p ∈ [1,∞) | (1+ε)-approx | O(nd⁵) +o(“exact“) | Dasgupta, Drineas, Harb, Kumar, Mahoney (submitted) |

**Note:** Ken Clarkson (SODA05) gets a (1+ε)-approximation for L1 regression in $O^*(d^{3.5}/ε^4)$ time.

He preprocessed $[A,b]$ to make it “well-rounded” or “well-conditioned” and then sampled.
Algorithm 1: Sampling for L2 regression

\[ Z_2 = \min_{x \in \mathbb{R}^d} \| b - Ax \|_2 = \| b - A\hat{x} \|_2 \]

\[
\begin{pmatrix}
A \\
\end{pmatrix}
\begin{pmatrix}
\hat{x} \\
\end{pmatrix} \approx
\begin{pmatrix}
b \\
\end{pmatrix}
\]

\( n \times d, \quad n \gg d \)

\textbf{Algorithm}

1. Fix a set of probabilities \( p_i, i=1\ldots n, \) summing up to 1.

2. Pick \( r \) indices from \( \{1,\ldots,n\} \) in \( r \) i.i.d. trials, with respect to the \( p_i \)'s.

3. For each sampled index \( j \), keep the \( j \)-th row of \( A \) and the \( j \)-th element of \( b \); rescale both by \( (1/rp_j)^{1/2} \).

4. Solve the induced problem.
Random sampling algorithm for L2 regression

\[
\mathcal{Z}_{2,s} = \min_{x \in \mathbb{R}^d} \|b_s - A_s x\|_2 = \|b_s - A_s \hat{x}_s\|_2
\]

\[
\begin{pmatrix}
A_s \\
\end{pmatrix}
\begin{pmatrix}
\hat{x}_s \\
\end{pmatrix}
\approx
\begin{pmatrix}
b_s \\
\end{pmatrix}
\]

scaling to account for undersampling

\[
|\mathcal{Z}_2 - \mathcal{Z}_{2,s}| \leq? \\
\|\hat{x} - \hat{x}_s\|_2 \leq? \\
\|A\hat{x}_s - b\|_2 \leq?
\]
Our results for $p=2$

If the $p_i$ satisfy a condition, then with probability at least $1-\delta$,

$$Z_{2,s} \leq (1 + \epsilon) Z_2$$

$$Z_2 \leq \|A\hat{x}_s - b\|_2 \leq (1 + \epsilon) Z_2$$

$$\|\hat{x} - \hat{x}_s\|_2 \leq \frac{\epsilon}{\sigma_{\min}(A)} Z_2$$

The sampling complexity is

$$r = O(d \log(d) \log(1/\delta) / \epsilon^2)$$
Our results for $p=2$, cont’d

If the $p_i$ satisfy a condition, then with probability at least $1-\delta$,

\[
Z_{2,s} \leq (1 + \epsilon) Z_2
\]

\[
Z_2 \leq \|A\hat{x}_s - b\|_2 \leq (1 + \epsilon) Z_2
\]

\[
\|\hat{x} - \hat{x}_s\|_2 \leq \epsilon \left(\frac{\kappa(A)}{\gamma}\right) \|\hat{x}\|_2
\]

The sampling complexity is

\[
r = O(d \log(d) \log(1/\delta)/\epsilon^2)
\]
Condition on the probabilities (1 of 2)

• **Important:** Sampling process must NOT lose any rank of $A$.
  
  (Since pseudoinverse will amplify that error!)

  $$Ax \approx b \quad \rightarrow \quad x_{OPT} = A^+b = V_A \Sigma_A^{-1} U_A^T b$$
  $$SAx \approx Sb \quad \rightarrow \quad x_{OPT} = (SA)^+ Sb = V_{SA} \Sigma_{SA}^{-1} U_{SA}^T Sb$$

• Sampling with respect to row lengths will fail.
  
  (They get coarse statistics to additive-error, not relative-error.)

• Need to disentangle "subspace info" and "size-of-$A$ info."
The condition that the $p_i$ must satisfy, are, for some $\beta_1 \in (0,1]$:

$$p_i \geq \beta_1 \frac{\|U(i)\|_2^2}{\sum_{j=1}^{n} \|U(j)\|_2^2}$$

Notes:

- Using the norms of the rows of any orthonormal basis suffices, e.g., Q from QR.
- $O(nd^2)$ time suffices (to compute probabilities and to construct a core-set).
- Open question: Is $O(nd^2)$ necessary?
- Open question: Can we compute good probabilities, or construct a coreset, faster?
- Original conditions (DMM06a) were stronger and more complicated.
Interpretation of the probabilities (1 of 2)

• What do the lengths of the rows of the $n \times d$ matrix $U = U_A$ “mean”?

• Consider possible $n \times d$ matrices $U$ of $d$ left singular vectors:

  $I_n|_k = k$ columns from the identity
  
  row lengths = 0 or 1
  
  $I_n|_k \times \rightarrow \times$

  $H_n|_k = k$ columns from the $n \times n$ Hadamard (real Fourier) matrix
  
  row lengths all equal
  
  $H_n|_k \times \rightarrow$ maximally dispersed

  $U_k = k$ columns from any orthogonal matrix
  
  row lengths between 0 and 1

• The lengths of the rows of $U = U_A$ correspond to a notion of information dispersal (i.e., where information is sent.)
The lengths of the rows of $U = U_A$ also correspond to a notion of statistical leverage or statistical influence.

- $p_i \approx \|U_{(i)}\|_2^2 = (AA^*)_{ii}$, i.e. they equal the diagonal elements of the “prediction” or “hat” matrix.
**Critical observation**

\[ Z_2 = \min_{x \in \mathbb{R}^d} \| b - Ax \|_2 = \| b - A\hat{x} \|_2 \]

\[ \begin{pmatrix}
A \\
\hat{x} \\
b
\end{pmatrix} \approx \begin{pmatrix}
n \times d, & n >> d
\end{pmatrix} \]

Sample & rescale
Critical observation, cont’d

\[ Z_2 = \min_{x \in \mathbb{R}^d} \|b - Ax\|_2 = \|b - A\hat{x}\|_2 \]

\[
\begin{pmatrix}
U
\end{pmatrix} \cdot \begin{pmatrix}
\Sigma
\end{pmatrix} \cdot \begin{pmatrix}
V \end{pmatrix}^T \begin{pmatrix}
\hat{x}
\end{pmatrix} \approx \begin{pmatrix}
b
\end{pmatrix}
\]

sample & rescale only \( U \)

sample & rescale
Critical observation, cont’d

\[ Z_{2,s} = \min_{x \in \mathbb{R}^d} \| b_s - A_s x \|_2 = \| b_s - A_s \hat{x}_s \|_2 \]

\[
\begin{pmatrix}
U_s \\
\Sigma \\
V \\
\end{pmatrix} \cdot \begin{pmatrix}
\Sigma \\
V \\
\end{pmatrix}^T \begin{pmatrix}
\hat{x}_s \\
b_s \\
\end{pmatrix} \approx \begin{pmatrix}
b_s \\
\end{pmatrix}
\]

Important observation: \( U_s \) is “almost orthogonal,” i.e., we can bound the spectral and the Frobenius norm of

\[ U_s^T U_s - I. \]

(FKV98, DK01, DKM04, RV04)
(Slow Random Projection) Algorithm:

Input: An $n \times d$ matrix $A$, a vector $b \in \mathbb{R}^n$.
Output: $x'$ that is approximation to $x_{OPT} = A^*b$.

• Construct a random projection matrix $P$, e.g., entries from $N(0,1)$.
• Solve $Z' = \min_x ||P(Ax-b)||_2$.
• Return the solution $x'$.

Theorem:
• $Z' \leq (1+\varepsilon) Z_{OPT}$.
• $||b-Ax'||_2 \leq (1+\varepsilon) Z_{OPT}$.
• $||x_{OPT}-x'||_2 \leq (\varepsilon/\sigma_{min}(A)) ||x_{OPT}||_2$.
• Running time is $O(nd^2)$ - due to $PA$ multiplication.
Random Projections and the Johnson-Lindenstrauss lemma

**J-L Lemma:** For every set $S$ of $n$ points in $\mathbb{R}^d$ and every $\epsilon > 0$, there exists a mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$, where $k = O(\epsilon^{-2} \log n)$, such that for all pairs $u, v \in S$:

$$(1 - \epsilon)|u - v|^2 \leq |f(u) - f(v)|^2 \leq (1 + \epsilon)|u - v|^2.$$ 

**Algorithmic results for J-L:**

- JL84: project to a random subspace
- FM88: random orthogonal matrix
- DG99: random orthogonal matrix
- IM98: matrix with entries from $\mathbb{N}(0,1)$
- Achlioptas03: matrix with entries from $\{-1,0,+1\}$
- Alon03: dependence on $n$ and $\epsilon$ (almost) optimal
Dense Random Projections and JL

\( P \) (the projection matrix) must be dense, i.e., \( \Omega(n) \) nonzeros per row.

- \( P \) may hit "concentrated" vectors, i.e. \( \|x\|_\infty / \|x\|_2 \approx 1 \)
  - e.g. \( x=(1,0,0,...,0)^T \) or \( U_A \) with non-uniform row lengths.
- Each projected coordinate is linear combination of \( \Omega(n) \) input coordinates.
- Performing the projection takes \( O(nd^2) \) time.

**Note:** Expensive sampling probabilities are needed for exactly the same reason!

**Ques:** What if \( P/S \) hits "well rounded" vectors, i.e., \( \|x\|_\infty / \|x\|_2 \approx 1/\sqrt{n} \)?
Let \( \Phi = PHD \) be a “preprocessed” projection:

\[
P \in \mathbb{R}^{k \times d} \quad \text{s.t.} \quad \begin{cases} 
P_{ij} \sim N(0, 1/q), \text{ with prob. } q, \text{ where } q = O\left(\frac{\log^2 n}{d}\right) \\ 
P_{ij} = 0, \text{ with prob. } 1 - q 
\end{cases}
\]

\( H \in \mathbb{R}^{d \times d} \) is a normalized Hadamard matrix:

\[
H_{ij} = d^{-1/2} (-1)^{<i-1,j-1>}
\]

\( D \in \mathbb{R}^{d \times d} \) is a diagonal matrix:

\( D_{ii} \) drawn from \( +1, -1 \) w.p. \( 1/2 \)
Fast Johnson-Lindenstrauss lemma (2 of 2)

Ailon and Chazelle (STOC06)

**Fast J-L Lemma:** Let $\Phi = PHD \in \mathbb{R}^{k \times d}$ be the sparse random projection as above. Given a set $S$ of $n$ points in $\mathbb{R}^d$ and an $\epsilon > 0$, for all pairs $u, v \in S$:

$$(1 - \epsilon) |u - v|^2_2 \leq |\Phi u - \Phi v|^2_2 \leq (1 + \epsilon) |u - v|^2_2.$$

**Notes:**
- $P$ - does the projection;
- $H$ - “uniformizes” or “densifies” sparse vectors;
- $D$ - ensures that wph dense vectors are not sparsified.

**Multiplication is “fast”**
- by $D$ - since $D$ is diagonal;
- by $H$ - use Fast Fourier Transform algorithms;
- by $P$ - since it has $O(\log^2 n)$ nonzeros per row.
Algorithm 3: Faster Projection for L2

(Fast Random Projection) Algorithm:

Input: An $n \times d$ matrix $A$, a vector $b \in \mathbb{R}^n$.
Output: $x'$ that is approximation to $x_{OPT}=A^*b$.

- Preprocess $[A \ b]$ with randomized Hadamard rotation $H_nD$.
- Construct a sparse projection matrix $P$ (with $O(\log^2 n)$ nonzero/row).
- Solve $Z' = \min_x \|\Phi(Ax-b)\|_2$ (with $\Phi=PH_nD$).
- Return the solution $x'$.

Theorem:

- $Z' \leq (1+\varepsilon) Z_{OPT}$.
- $\|b-Ax'\|_2 \leq (1+\varepsilon) Z_{OPT}$.
- $\|x_{OPT}-x'\|_2 \leq (\varepsilon/\sigma_{\min}(A))\|x_{OPT}\|_2$.
- Running time is $O(nd \log n) = o(nd^2)$ since projection is sparse!
Algorithm 4: Faster Sampling for L2

(Fast Random Sampling) Algorithm:

Input: An $n \times d$ matrix $A$, a vector $b \in \mathbb{R}^n$.
Output: $x'$ that is approximation to $x_{OPT} = A^*b$.

- Preprocess $[A \ b]$ with randomized Hadamard rotation $H_n D$.
- Construct a uniform sampling matrix $S$ (with $O(d \log d \log^2 n/\varepsilon^2)$ samples).
- Solve $Z' = \min_x ||\Phi(Ax-b)||_2$ (with $\Phi=SH_n D$).
- Return the solution $x'$.

Theorem:
- $Z' \leq (1+\varepsilon) Z_{OPT}$.
- $||b-Ax'||_2 \leq (1+\varepsilon) Z_{OPT}$.
- $||x_{OPT}-x'||_2 \leq (\varepsilon/\sigma_{\min}(A))||x_{OPT}||_2$.
- Running time is $O(nd \log n) = o(nd^2)$ since sampling is uniform!!
Proof idea for $o(nd^2)$ L2 regression

Sarlos (FOCS06) and Drineas, Mahoney, Muthukrishnan, and Sarlos 07

$$Z_{\text{exact}} = \min_x \|Ax-b\|_2$$

- Sample w.r.t. $p_i = \|U_{A,(i)}\|_2^2/d$ -- the “right” probabilities.
- Projection must be dense since $p_i$ may be very non-uniform.

$$Z_{\text{rotated}} = \min_x \|HD(Ax-b)\|_2$$

- $HDA = HDU_A \Sigma_A V_A^T$
- $p_i = \|U_{HDA,(i)}\|_2^2$ are approximately uniform (up to $\log^2 n$ factor)

$$Z_{\text{sampled/projected}} = \min_x \|(S/P)HD(Ax-b)\|_2$$

- Sample a “small” number of constraints and solve sub-problem;
  - “small” is $O(\log^2 n)$ here versus constant w.r.t $n$ before.
- Do “sparse” projection and solve sub-problem;
  - “sparse” means $O(\log^2 n)$ non-zeros per row.
What made the L2 result work?

The L2 sampling algorithm worked because:

- For $p=2$, an orthogonal basis (from SVD, QR, etc.) is a “good” or “well-conditioned” basis.
  (This came for free, since orthogonal bases are the obvious choice.)

- Sampling w.r.t. the “good” basis allowed us to perform “subspace-preserving sampling.”
  (This allowed us to preserve the rank of the matrix.)

Can we generalize these two ideas to $p \neq 2$?
Let $A$ be an $n \times m$ matrix of rank $d << n$, let $p \in [1, \infty)$, and $q$ its dual.

**Definition:** An $n \times d$ matrix $U$ is an $(\alpha, \beta, p)$-well-conditioned basis for $\text{span}(A)$ if:

1. $|||U|||^p \leq \alpha$, (where $|||U|||^p = (\Sigma_{ij} |U_{ij}|^p)^{1/p}$)
2. For all $z \in \mathbb{R}^d$, $||z||_q \leq \beta ||Uz||_p$.

$U$ is a *p-well-conditioned basis* if $\alpha, \beta = d^{O(1)}$, independent of $m,n$. 

---

**p-well-conditioned basis (definition)**
Let $A$ be an $n \times m$ matrix of rank $d \ll n$, let $p \in [1, \infty)$, and $q$ its dual.

**Theorem:** There exists an $(\alpha, \beta, p)$-well-conditioned basis $U$ for $\text{span}(A)$ s.t.:

if $p < 2$, then $\alpha = \frac{d}{p+1/2}$ and $\beta = 1$,

if $p = 2$, then $\alpha = \frac{d}{2}$ and $\beta = 1$,

if $p > 2$, then $\alpha = \frac{d}{p+1/2}$ and $\beta = \frac{d}{q-1/2}$.

$U$ can be computed in $O(nmd+nd^5 \log n)$ time (or just $O(nmd)$ if $p = 2$).
Algorithm:

- Let \( A = QR \) be any QR decomposition of \( A \).
  (Stop if \( p = 2 \).)
- Define the norm on \( \mathbb{R}^d \) by \( \| z \|_{Q,p} = \| Qz \|_p \).
- Let \( C \) be the unit ball of the norm \( \| \cdot \|_{Q,p} \).
- Let the \( d \times d \) matrix \( F \) define the Lowner-John ellipsoid of \( C \).
- Decompose \( F = G^T G \),
  where \( G \) is full rank and upper triangular.
- Return \( U = QG^{-1} \)
  as the \( p \)-well-conditioned basis.
Subspace-preserving sampling

Let $A$ be an $n \times m$ matrix of rank $d \ll n$, let $p \in [1, \infty)$.
Let $U$ be an $(\alpha, \beta, p)$-well-conditioned basis for $\text{span}(A)$.

**Theorem**: Randomly sample rows of $A$ according to the probability distribution:

$$p_i \geq \min \left\{ 1, \frac{\|U(i)\|^p_p}{\|U\|^p_p} r \right\}$$

where:

$$r \geq 32^p (\alpha \beta)^p (d \ln(\frac{12}{\epsilon}) + \ln(\frac{2}{\delta}))/ (p^2 \epsilon^2)$$

Then, with probability $1 - \delta$, the following holds for all $x$ in $\mathbb{R}^m$:

$$\|\|S Ax\|_p - \|Ax\|_p\| \leq \epsilon \|Ax\|_p$$
Algorithm 5: Approximate Lp regression

Input: An n x m matrix A of rank d<<n, a vector b \( \in \mathbb{R}^n \), and p \( \in [1, \infty) \).
Output: x" (or x' if do only Stage 1).

- Find a \( p \)-well-conditioned basis U for span(A).

- **Stage 1 (constant-factor):**
  - Set \( p_i \approx ||U(i)||r_1 \), where \( r_1 = O(36^p d^{k+1}) \) and \( k = \max\{p/2+1, p\} \).
  - Generate (implicitly) a sampling matrix S from \( \{p_i\} \).
  - Let x' be the solution to: \( \min_x ||S(Ax-b)||_p \).

- **Stage 2 (relative-error):**
  - Set \( q_i \approx \min\{1, \max\{p_i, Ax'-b\}\} \), where \( r_2 = O(r_1/\varepsilon^2) \).
  - Generate (implicitly, a new) sampling matrix T from \( \{q_i\} \).
  - Let x" be the solution to: \( \min_x ||T(Ax-b)||_p \).
Theorem for approximate Lp regression

**Constant-factor approximation:**
- Run Stage 1, and return $x'$. Then w.p. $\geq 0.6$:
  $$\|Ax' - b\|_p \leq 8 \|Ax_{opt} - b\|_p.$$

**Relative-error approximation:**
- Run Stage 1 and Stage 2, and return $x''$. Then w.p. $\geq 0.5$:
  $$\|Ax'' - b\|_p \leq (1+\varepsilon) \|Ax_{opt} - b\|_p.$$

**Running time:**
- The $i^{th}$ ($i=1,2$) stage of the algorithm runs in time:
  $$O(nmd + nd^5 \log n + \phi(20r_i,m)),$$
where $\phi(s,t)$ is the time to solve an $s$-by-$t$ Lp regression problem.
Extensions and Applications

(Theory:) Relative-error CX and CUR low-rank matrix approximation.

- $\|A - CC^+ A\|_F \leq (1+\varepsilon) \|A - A_k\|_F$
- $\|A - CUR\|_F \leq (1+\varepsilon) \|A - A_k\|_F$

(Theory:) Core-sets for Lp regression problems, $p \in [1,\infty)$.

(Application:) DNA SNP and microarray analysis.

- SNPs are “high leverage” data points.

(Application:) Feature Selection and Learning in Term-Document matrices.

- Regularized Least Squares Classification.
- Sometimes performs better than state of the art supervised methods.
Conclusion

Fast Sampling Algorithm for L2 regression:

- **Core-set** and $(1+\varepsilon)$-approximation in $O(nd^2)$ time.
- Expensive but Informative sampling probabilities.
- Runs in $o(nd^2)$ time after randomized Hadamard preprocessing.

Fast Projection Algorithm for L2 regression:

- Gets a $(1+\varepsilon)$-approximation in $o(nd^2)$ time.
- Uses the recent “Fast” Johnson-Lindenstrauss Lemma.

Sampling algorithm for Lp regression, for $p \in [1,\infty)$:

- **Core-set** and $(1+\varepsilon)$-approximation in $o(\text{exact})$ time ($\Theta(\text{exact})$ time for $p=2$).
- Uses $p$-well-conditioned basis and subspace-preserving sampling.