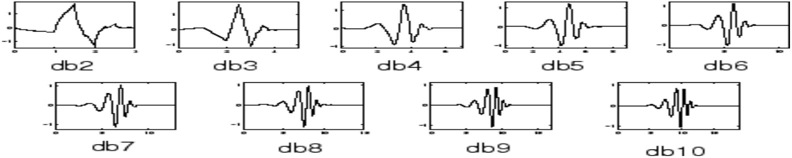
Outline for today

- Dictionary learning as a model for the first layer of a deep net
- Algorithms used for recovery of sparse activations:
  Selection of a subset of a dictionary for optimal signal representation
  Proofs of recovery of sparse activations using one step thresholding, matching pursuit algorithms, and convex regularisers
- The K-SVD algorithm and other methods to solve the dictionary update step

Wavelet, curvelet, and contourlet: fixed representations

Applied and computational harmonic analysis community developed representations with optimal approximation properties for piecewise smooth functions.



Most notable are the Daubechies wavelets and Curvelets/Contourlets pioneered by Candes and Donoho.
While optimal, in a certain sense, for a specific class of functions, they can typically be improved upon for any particular data set.

Optimality of curvelets in 2D

Theorem (Candes and Donoho 02'[a])

[a] http://www.curvelet.org/papers/CurveEdges.pdf

Let f be a two dimensional function that is piecewise C^2 with a boundary that is also C^2. Let f_n^F, f_n^W, and f_n^C be the best approximation of f using n terms of the Fourier, Wavelet and Curvelet representation respectively. Then their approximation error satisfy $\|f - f_n^F\|_{L^2}^2 = \mathcal{O}(n^{-1/2})$, $\|f - f_n^W\|_{L^2}^2 = \mathcal{O}(n^{-1})$, and $\|f - f_n^C\|_{L^2}^2 = \mathcal{O}(n^{-2}\log^3(n))$; moreover, no fixed representation can have a rate exceeding $\mathcal{O}(n^{-2})$.

Near optimality of such representation suggest a good first layer.

Dictionary learning

While there are representations that are near optimal for realistic classes of functions, one can usually improve upon them for a particular data set; that is, one can learn a better dictionary for that data.

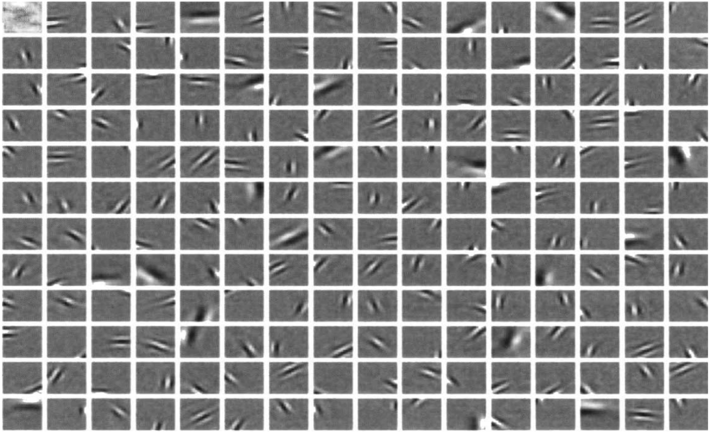
Let $Y \in \mathbb{R}^{m \times p}$ be a collection of p data elements in $\mathbb{R}^m$. Each data element y_i can be well represented by a dictionary $D \in \mathbb{R}^{m \times n}$ if there exists an x_i with at most k nonzeros such that $\|y_i - Dx_i\| \leq \epsilon(k)$. This can be combined in matrix notation as $\min_X \|Y - DX\|$ subject to $\|x_i\|_0 \leq k$ for $i = 1, \ldots, p$.

Note that solving for the optimal x_i for each y_i is in general NP hard, but for well behaved D it is easy.

Dictionary learning does a step further and learns the optimal D

$$\min_{D,X} \|Y - DX\| \quad \text{subject to} \quad \|x_i\|_0 \leq k, \ \|d_i\| = 1$$

Dictionary learned from natural scenes (Olshausen and Field 96'[1]



Note the similarity to curvelets and the first layer of deep CNNs.

[1] https://www.nature.com/articles/381607a0.pdf

Dictionary learning through ADMM

Alternating direction method of multipliers (ADMM) holds all but one component of a problem fixed and solves the other, then iterates through the variables to be solved for.
For dictionary learning this is iteratively solving:

$$\min_{X:\|x_i\|_0 \le k} \|Y - DX\| \quad \text{then} \quad \min_{D:\|d_i\|=1} \|Y - DX\|$$

There are many methods for solving each of these subproblems.
Solving for X is more challenging, and will be our focus for now.
While better solutions exist, if X is held fixed one can solve for $YX^T = DXX^T$ as $X \in \mathbb{R}^{n \times p}$ for $p > n$ allowing $D = YX^T(XX^T)^{-1}$ followed by normalising the columns.

Coherence

- With $n > m$ the columns of $D \in \mathbb{R}^{m \times n}$ can't be orthogonal, we measure their dependence by the coherence of the columns.

$$\mu_2(D) := \max_{i \neq j} |d_i^* d_j|$$

- The collection of columns which are minimally coherent are called Grassman Frames and satisfy:

$$\mu_2(A_{m,n}) \geq \left(\frac{n-m}{m(n-1)} \right)^{1/2} \sim m^{-1/2}$$

- We can use coherence to analyse a number of algorithms to try and solve the sparse coding problem

$$\min_x \|x\|_0 \quad \text{subject to} \quad \|y_i - Dx_i\| \leq \tau$$

which in its worst case is NP-hard to solve.

One step thresholding

Input: y, D and k (number of non-zeros in output vector).
Algorithm: Set Λ the index set of the $k \leq m$ largest in $|D^* b|$
Output the n-vector x whose entries are

$$x_\Lambda = (D_\Lambda^* D_\Lambda)^{-1} D_\Lambda^* y \quad \text{and} \quad x(i) = 0 \text{ for } i \notin \Lambda.$$

Theorem

Let $y = Dx_0$, with the columns of D having unit ℓ^2 norm, and

$$\|x_0\|_0 < \frac{1}{2}\left(\nu_\infty(x_0) \cdot \mu_2(D)^{-1} + 1\right),$$

then the Thresholding decoder with $k = \|x_0\|_0$ will return x_0, with $\nu_p(x) := \min_{j \in \text{supp}(x)} |x(j)| / \|x\|_p$.

One step thresholding (proof)

Proof.

With $y = Dx_0$, denote $w = D^* b = D^* D x_0$.
The i^{th} entry in w is equal to $w_i = \sum_{j \in \text{supp}(x_0)} x_0(j) d_i^* d_j$.
For $i \notin \text{supp}(x_0)$ the magnitude of w_i is bounded above as:

$$|w_i| \leq \sum_{j \in \text{supp}(x_0)} |x_0(j)| \cdot |d_i^* d_j| \leq k\mu_2(D) \|x_0\|_\infty.$$

For $i \in \text{supp}(x_0)$ the magnitude of w_i is bounded below as:

$$\begin{aligned} |w_i| &\geq |x_0(i)| - \left| \sum_{j \in \text{supp}(x_0), j \neq i} x_0(j) d_i^* d_j \right| \\ &\geq |x_0(i)| - (k-1)\mu_2(D) \|x_0\|_\infty. \end{aligned}$$

Recovery if $\max_{i \notin \text{supp}(x_0)} |w_i| < \min_{i \in \text{supp}(x_0)} |w_i|$. □

Matching Pursuit (Tropp 04'[2])

Input: y, D and k (number of nonzeros in output vector).
Algorithm: Let $r^j := y - Dx^j$.
Set $x^0 = 0$, and let $i := \text{argmax}_\ell |d_\ell^* r^j|$ and define
$x^{j+1} = x^j + (d_i^* r^j) e_i$ where e_i is the i^{th} coordinate vector.
Output x^j when a termination criteria is obtained.

Theorem

Let $y = Dx_0$, with the columns of D having unit ℓ^2 norm, and

$$\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right),$$

then Matching Pursuit will have $\text{supp}(x^j) \subseteq \text{supp}(x_0)$ for all j.

$*$ Preferable over one step thresholding: no dependence on $\nu_p(x_0)$.

[2] https://ieeexplore.ieee.org/document/1337101

Matching Pursuit (proof)

Proof.

Assume $\text{supp}(x^j) \subset \text{supp}(x_0)$ for some j, which is true for $j = 0$.
Let $r^j = y - Dx^j$, and $w_i = \sum_{\ell \in \text{supp}(x_0)} (x_0 - x^j)(\ell) \cdot d_i^* d_\ell$.
For $i \notin \text{supp}(x_0)$ the magnitude of w_i is bounded above as:

$$|w_i| \leq \sum_{\ell \in \text{supp}(x_0)} |(x_0 - x^j)(\ell)| \cdot |d_i^* d_\ell| \leq k\mu_2(D) \| |x_0 - x^j\|_\infty.$$

For $i \in \text{supp}(x_0)$ the magnitude of w_i is bounded below as:

$$\begin{aligned} |w_i| &\geq |(x_0 - x^j)(i)| - \left| \sum_{\ell \in \text{supp}(x_0), \ell \neq i} (x_0 - x^j)(\ell) \cdot d_i^* d_\ell \right| \\ &\geq |(x_0 - x^j)(i)| - (k-1)\mu_2(D) \|x_0 - x^j\|_\infty. \end{aligned}$$

Recovery if $\max_{i \in \text{supp}(x_0)} |w_i| > \max_{i \notin \text{supp}(x_0)} |w_i|$.

□

Orthogonal Matching Pursuit (Tropp 04'[3])

Input: y, D and k (number of nonzeros in output vector).
Algorithm: Let $r^j := y - Dx^j$.
Set $x^0 = 0$ and Λ^0 to be the empty set, and set $j = 0$.
Let $r^j := y - Dx^j$, $i := \text{argmax}_\ell |d_\ell^* r^j|$, and $\Lambda^{j+1} = i \bigcup \Lambda^j$.
Set $x_{\Lambda^{j+1}}^{j+1} = (D_{\Lambda^{j+1}}^* D_{\Lambda^{j+1}})^{-1} D_{\Lambda^{j+1}}^* y$
and $x^{j+1}(\ell) = 0$ for $\ell \notin \Lambda^{j+1}$, and set $j = j + 1$.
Output x^j when a termination criteria is obtained.

Theorem

Let $y = Dx_0$, with the columns of D having unit ℓ^2 norm, and

$$\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right),$$

then after $\|x_0\|_{\ell^0}$ steps, Orthogonal Matching Pursuit recovers x_0.

$*$ Proof, same as Matching Pursuit. Finite number of steps.

[3] https://ieeexplore.ieee.org/document/1337101

ℓ^1-regularization (Tropp 06'[4])

Input: y and D.
"Algorithm": Return $\text{argmin}\|x\|_1$ subject to $y = Dx$.

Theorem

Let $y = A_{m,n}x_0$, with

$$\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right),$$

then the solution of ℓ^1-regularization is x_0.

$*$ Preferable over OMP: faster if use good ℓ^1 solver.

[4] http://users.cms.caltech.edu/~jtropp/papers/Tro06-Just-Relax.pdf

ℓ^1-regularization (proof, page 1)

Proof.

Let $\Lambda_0 := supp(x_0)$ and $\Lambda_1 := supp(x_1)$ with $y = Dx_0 = Dx_1$, and $\exists i$ with $i \in \Lambda_1$ with $i \notin \Lambda_0$.

Note that because $y = D_{\Lambda_0} x_0 = D_{\Lambda_1} x_1$,

$$\begin{aligned} \|x_0\|_1 &= \|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} D_{\Lambda_0} x_0\|_1 \\ &= \|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} y\|_1 \\ &= \|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} D_{\Lambda_1} x_1\|_1. \end{aligned}$$

Establish bounds on $(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} d_i$.

To establish proof need bounds for $i \in \Lambda$ and $i \notin \Lambda$.

For $i \in \Lambda_0$: $\|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} d_i\|_1$
$= \|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} D_{\Lambda_0} e_i\|_1 = \|e_i\|_1 = 1$

□

ℓ^1-regularization (proof, page 2)

Proof.

For any $i \notin \Lambda_0$ we establish the bound in two parts; first,

$$\|D^*_{\Lambda_0} d_i\|_1 \leq \sum_{\ell \in \Lambda_0} |d^*_\ell d_i| \leq k\mu_2(D).$$

Noting $D^*_{\Lambda_0} D_{\Lambda_0} = I_{k,k} + B$ where $B_{i,i} = 0$ and $|B_{i,j}| \leq \mu_2(D)$, then

$$\|(I_{k,k}+B)^{-1}\|_1 = \left\|\sum_{\ell=0}^{\infty}(-B)^\ell\right\|_1 \leq \sum_{\ell=0}^{\infty} \|B\|_1^\ell = \frac{1}{1-\|B\|_1} \leq \frac{1}{1-(k-1)\mu}$$

Therefore, for $i \notin \Lambda_0$:

$$\|(D^*_{\Lambda_0} D_{\Lambda_0})^{-1} D^*_{\Lambda_0} d_i\|_1 \leq \frac{k\mu_2(D)}{(1-(k-1)\mu_2(D))} < 1$$

□

ℓ^1-regularization (proof, page 3)

Proof.

Proof concludes through triangle inequality and use that:

- For $i \in \Lambda_0$: $\|(D_{\Lambda_0}^* D_{\Lambda_0})^{-1} D_{\Lambda_0}^* d_i\|_1 = 1$
- For $i \notin \Lambda_0$: $\|(D_{\Lambda_0}^* D_{\Lambda_0})^{-1} D_{\Lambda_0}^* d_i\|_1 < 1$
- And $\exists i$ with $i \in \Lambda_1$ and $i \notin \Lambda_0$.

Then,

$$\begin{aligned} \|x_0\|_1 &= \left\| \sum_{i \in \Lambda_1} (D_{\Lambda_0}^* D_{\Lambda_0})^{-1} D_{\Lambda_0}^* d_i x_1(i) \right\|_1 \\ &\leq \sum_{i \in \Lambda_1} |x_1(i)| \cdot \left\| (D_{\Lambda_0}^* D_{\Lambda_0})^{-1} D_{\Lambda_0}^* d_i \right\|_1 \\ &< \sum_{i \in \Lambda_1} |x_1(i)| = ||x_1||_1. \end{aligned}$$

□

But, is the solution even unique?

The sparsity of the sparsest vector in the nullspace of D,

$$spark(D) := \min_z \|z\|_{\ell^0} \quad \text{subject to} \quad Dz = 0.$$

Theorem (Coherence and Spark)

$$\text{spark}(D) \geq \min(m+1, \mu_2(D)^{-1} + 1)$$

If $\|x_0\| < (\mu_2(D)^{-1} + 1)/2$ unique satisfying $y = Dx_0$.

Proof.

Gershgorin disc theorem for $D^*_\Lambda D_\Lambda$ with $|\Lambda| = k$:
1 on diagonal, off diagonals bounded by $\mu_2(D)$.
If $k < \mu_2(D)^{-1} + 1$, smallest singular value of $D^*_\Lambda D_\Lambda$ is > 0 □

How to interpret these results, is better possible?

- When is $\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right)$?
  Grassman Frames: $\mu_2(D) \geq \left(\frac{n-m}{m(n-1)}\right)^{1/2} \sim m^{-1/2}$
  "Sqrt bottleneck" $\|x_0\|_{\ell^0} \lesssim \sqrt{m}$

How to interpret these results, is better possible?

- When is $\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right)$?
  Grassman Frames: $\mu_2(D) \geq \left(\frac{n-m}{m(n-1)}\right)^{1/2} \sim m^{-1/2}$
  "Sqrt bottleneck" $\|x_0\|_{\ell^0} \lesssim \sqrt{m}$
- Is better possible? (not without more)
  Fourier & Dirac: $D = [F\ I]$ for m the square of an integer:
  Let $\Lambda = [\sqrt{m},\ 2\sqrt{m},\ \cdots, m]$, then
  $\sum_{j\in\Lambda} e_j = \sum_{j\in\Lambda} f_j \Longrightarrow \text{spark}(D) = 2\sqrt{m}$.

How to interpret these results, is better possible?

- When is $\|x_0\|_{\ell^0} < \frac{1}{2}\left(\mu_2(D)^{-1} + 1\right)$?
  Grassman Frames: $\mu_2(D) \geq \left(\frac{n-m}{m(n-1)}\right)^{1/2} \sim m^{-1/2}$
  "Sqrt bottleneck" $\|x_0\|_{\ell^0} \lesssim \sqrt{m}$
- Is better possible? (not without more)
  Fourier & Dirac: $D = [F\ I]$ for m the square of an integer:
  Let $\Lambda = [\sqrt{m},\ 2\sqrt{m},\ \cdots, m]$, then
  $\sum_{j\in\Lambda} e_j = \sum_{j\in\Lambda} f_j \Longrightarrow \text{spark}(D) = 2\sqrt{m}$.
- Slightly more accurate sometimes with cumulative coherence:
  $\max_{i\in\Lambda}\max_{\Lambda'} \sum_{j\in\Lambda'} d_i^* d_j$
- To avoid pathological cases introduce randomness

One step thresholding: average sign pattern [ScVa07]

Input: y, D and k (number of nonzeros in output vector).
Algorithm: Set Λ the index set of the $k \leq m$ largest in $|D^* y|$
Output the n-vector x whose entries are

$$x_\Lambda = (D_\Lambda^* D_\Lambda)^{-1} D_\Lambda y \quad \text{and} \quad x(i) = 0 \text{ for } i \notin \Lambda.$$

One step thresholding: average sign pattern [ScVa07]

Input: y, D and k (number of nonzeros in output vector).
Algorithm: Set Λ the index set of the $k \leq m$ largest in $|D^*y|$
Output the n-vector x whose entries are

$$x_\Lambda = (D_\Lambda^* D_\Lambda)^{-1} D_\Lambda y \quad \text{and} \quad x(i) = 0 \text{ for } i \notin \Lambda.$$

Theorem

Let $y = Dx_0$, with the columns of D having unit ℓ^2 norm, the sign of the nonzeros in x_0 selected randomly from ± 1 independent of D, and

$$\|x_0\|_{\ell^0} < (128 \log(2n/\epsilon))^{-1} \nu_\infty^2(x_0) \mu_2^{-2}(D),$$

then, with probability greater than $1 - \epsilon$, the Thresholding decoder with $k = \|x_0\|_{\ell^0}$ will return x_0.

One step thresholding: average sign pattern (proof, pg. 1)

Theorem (Rademacher concentration)

Fix a vector α. Let ϵ be a Rademacher series, vector with entries drawn uniformly from ± 1, of the same length as α, then

$$\text{Prob}\left(\left|\sum_i \epsilon_i \alpha_i\right| > t\right) \leq 2\exp\left(\frac{-t^2}{32\|\alpha\|_2^2}\right)$$

Let $\Lambda := \text{supp}(x_0)$. Thresholding fail to recover x_0 if

$$\max_{i\notin\Lambda}|d_i^* y| > \min_{i\in\Lambda}|d_i^* y|.$$

$$\text{Prob}\left(\max_{i\notin\Lambda}|d_i^* y| > p \quad \text{and} \quad \min_{i\in\Lambda}|d_i^* y| < p\right) \quad \leq$$

$$\text{Prob}\left(\max_{i\notin\Lambda}|d_i^* y| > p\right) + \text{Prob}\left(\min_{i\in\Lambda}|d_i^* y| < p\right) \quad =: \quad P_1 + P_2$$

One step thresholding: average sign pattern (proof, pg. 2)

$$\begin{aligned}
P_1 &= \mathrm{Prob}\left(\max_{i\notin\Lambda}|d_i^* y| > p\right) \\
&\leq \sum_{i\notin\Lambda}\mathrm{Prob}\left(|d_i^* y| > p\right) \\
&= \sum_{i\notin\Lambda}\mathrm{Prob}\left(\left|\sum_{j\in\Lambda} x_0(j)(d_i^* d_j)\right| > p\right) \\
&\leq 2\sum_{i\notin\Lambda}\exp\left(\frac{-p^2}{32\sum_{j\in\Lambda}|x_0(j)|^2|d_i^* d_j|^2}\right) \\
&\leq 2(n-k)\exp\left(\frac{-p^2}{32k\|x_0\|_\infty^2\mu_2^2(D)}\right).
\end{aligned}$$

One step thresholding: average sign pattern (proof, pg. 3)

$$
\begin{aligned}
P_2 &= \text{Prob}\left(\min_{i\in\Lambda} |d_i^* y| < p\right) \\
&\leq \text{Prob}\left(\min_{i\in\Lambda} |x_0(i)| - \max_{i\in\Lambda}\left|\sum_{j\in\Lambda, j\neq i} x_0(j)(d_i^* d_j)\right| < p\right) \\
&\leq \sum_{i\in\Lambda} \text{Prob}\left(\left|\sum_{j\in\Lambda, j\neq i} x_0(j)(d_i^* d_j)\right| > \min_{i\in\Lambda} |x_0(i)| - p\right) \\
&\leq 2\sum_{i\in\Lambda} \exp\left(\frac{-(\min_{i\in\Lambda}|x_0(i)| - p)^2}{32\sum_{j\in\Lambda, j\neq i}|x_0(j)|^2 |d_i^* d_j|^2}\right) \\
&\leq 2k\exp\left(\frac{-(\min_{i\in\Lambda}|x_0(i)| - p)^2}{32k\|x_0\|_\infty^2 \mu_2^2(D)}\right).
\end{aligned}
$$

One step thresholding: average sign pattern (proof, pg. 4)

Balance P_1 and P_2 by setting $p := \min_{i\in\Lambda} |x_0(i)|/2$:

$$P_1 + P_2 \leq 2n \exp\left(\frac{-(\min_{i\in\Lambda} |x_0(i)|)^2}{128k\|x_0\|_\infty^2 \mu_2^2(D)}\right) \leq 2n \exp\left(\frac{-\nu_\infty(x_0)^2}{128k\mu_2^2(D)}\right).$$

Setting this bound on the probability of failure equal to ϵ and solving for k yields the conclusion of the proof. □

- Similar work for matching pursuit by Schnass, ℓ^1 by Tropp, and in Statistical RICs
- Stronger uniform statements we need more than coherence.

Dictionary learning through ADMM

Alternating direction method of multipliers (ADMM) holds all but one component of a problem fixed and solves the other, then iterates through the variables to be solved for.
For dictionary learning this is iteratively solving:

$$\min_{X:\|x_i\|_0\leq k} \|Y - DX\| \quad \text{then} \quad \min_{D:\|d_i\|=1} \|Y - DX\|$$

Returning to the dictionary update step. Algorithms include
Method of optimal directions:
solve for $YX^T = DXX^T$ as $X \in \mathbb{R}^{n\times p}$ for $p > n$ allowing $D = YX^T(XX^T)^{-1}$ followed by normalising the columns,
K-SVD, and steepest descent or other gradient updates of D.

Dictionary learning: K-SVD (Aharon et al. '06[5])

For a fixed sparse code one can view $\min_{D:\|d_i\|=1} \|Y - DX\|$ in terms of individual columns:

$$\left\| Y - \sum_{i=1}^{n} d_i \tilde{x}_i^T \right\|$$

where $\tilde{x}_i^T$ is the i^{th} row of X.
Being faithful to the sparsity constraint, we can view d_i as a column used to represent those columns in Y indexed by the support of $\tilde{x}_i^T$. Letting $E_i = [Y - \sum_{j\neq i} d_j \tilde{x}_j^T]_{\mathrm{supp}(\tilde{x}_i^T)}$ our task is to minimize

$$\left\| E_i - d_i \tilde{z}_i^T \right\|$$

where z_i^T is a vector of length $|\mathrm{supp}(\tilde{x}_i^T)|$, and whose solution is given by the best rank 1 approximation of E_i.

[5] https://ieeexplore.ieee.org/document/1710377