Introduction to Machine Learning (67577)
Lecture 11

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

Dimensionality Reduction

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space
- Why?

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space
- Why?
  - Reduces training (and testing) time

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space
- Why?
  - Reduces training (and testing) time
  - Reduces estimation error

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space
- Why?
  - Reduces training (and testing) time
  - Reduces estimation error
  - Interpretability of the data, finding meaningful structure in data, illustration

Dimensionality Reduction

- Dimensionality Reduction = taking data in high dimensional space and mapping it into a low dimensional space
- Why?
  - Reduces training (and testing) time
  - Reduces estimation error
  - Interpretability of the data, finding meaningful structure in data, illustration
- Linear dimensionality reduction: $\mathbf{x} \mapsto W\mathbf{x}$ where $W \in \mathbb{R}^{n,d}$ and $n < d$

Outline

1. Principal Component Analysis (PCA)
2. Random Projections
3. Compressed Sensing

Principal Component Analysis (PCA)

$$\mathbf{x} \mapsto W\mathbf{x}$$

- What makes W a good matrix for dimensionality reduction ?

Principal Component Analysis (PCA)

$$\mathbf{x} \mapsto W\mathbf{x}$$

- What makes W a good matrix for dimensionality reduction ?
- Natural criterion: we want to be able to approximately recover $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$

Principal Component Analysis (PCA)

$$\mathbf{x} \mapsto W\mathbf{x}$$

- What makes W a good matrix for dimensionality reduction ?
- Natural criterion: we want to be able to approximately recover $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- PCA:

Principal Component Analysis (PCA)

$$\mathbf{x} \mapsto W\mathbf{x}$$

- What makes W a good matrix for dimensionality reduction ?
- Natural criterion: we want to be able to approximately recover $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- PCA:
  - Linear recovery: $\tilde{\mathbf{x}} = U\mathbf{y} = UW\mathbf{x}$

Principal Component Analysis (PCA)

$$\mathbf{x} \mapsto W\mathbf{x}$$

- What makes W a good matrix for dimensionality reduction ?
- Natural criterion: we want to be able to approximately recover $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- PCA:
  - Linear recovery: $\tilde{\mathbf{x}} = U\mathbf{y} = UW\mathbf{x}$
  - Measures "approximate recovery" by averaged squared norm: given examples $\mathbf{x}_1, \ldots, \mathbf{x}_m$, solve

$$\operatorname*{argmin}_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|^2$$

Solving the PCA Problem

$$\operatorname*{argmin}_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|^2$$

Solving the PCA Problem

$$\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|^2$$

Theorem

Let $A = \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top$ *and let* $\mathbf{u}_1, \ldots, u_n$ *be the* n *leading eigenvectors of* A*. Then, the solution to the PCA problem is to set the columns of* U *to be* $\mathbf{u}_1, \ldots, \mathbf{u}_n$ *and to set* $W = U^\top$

Proof main ideas

- UW is of rank n, therefore its range is n dimensional subspace, denoted S

Proof main ideas

- UW is of rank n, therefore its range is n dimensional subspace, denoted S
- The transformation $\mathbf{x} \mapsto UW\mathbf{x}$ moves $\mathbf{x}$ to this subspace

Proof main ideas

- UW is of rank n, therefore its range is n dimensional subspace, denoted S
- The transformation $\mathbf{x} \mapsto UW\mathbf{x}$ moves $\mathbf{x}$ to this subspace
- The point in S which is closest to $\mathbf{x}$ is $VV^\top\mathbf{x}$, where columns of V are orthonormal basis of S

Proof main ideas

- UW is of rank n, therefore its range is n dimensional subspace, denoted S
- The transformation $\mathbf{x} \mapsto UW\mathbf{x}$ moves $\mathbf{x}$ to this subspace
- The point in S which is closest to $\mathbf{x}$ is $VV^\top\mathbf{x}$, where columns of V are orthonormal basis of S
- Therefore, we can assume w.l.o.g. that $W = U^\top$ and that columns of U are orthonormal

Proof main ideas

Observe:

$$\begin{aligned}\|\mathbf{x} - UU^\top \mathbf{x}\|^2 &= \|\mathbf{x}\|^2 - 2\mathbf{x}^\top UU^\top \mathbf{x} + \mathbf{x}^\top UU^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathbf{x}^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathrm{trace}(U^\top \mathbf{x}\mathbf{x}^\top U)\ ,\end{aligned}$$

Proof main ideas

Observe:

$$\begin{aligned}\|\mathbf{x} - UU^\top \mathbf{x}\|^2 &= \|\mathbf{x}\|^2 - 2\mathbf{x}^\top UU^\top \mathbf{x} + \mathbf{x}^\top UU^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathbf{x}^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathrm{trace}(U^\top \mathbf{x}\mathbf{x}^\top U) ~,\end{aligned}$$

Therefore, an equivalent PCA problem is

$$\operatorname*{argmax}_{U \in \mathbb{R}^{d,n} : U^\top U = I} \mathrm{trace}\left(U^\top \left(\sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top\right) U\right) ~.$$

Proof main ideas

Observe:

$$\begin{aligned}\|\mathbf{x} - UU^\top \mathbf{x}\|^2 &= \|\mathbf{x}\|^2 - 2\mathbf{x}^\top UU^\top \mathbf{x} + \mathbf{x}^\top UU^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathbf{x}^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathrm{trace}(U^\top \mathbf{x}\mathbf{x}^\top U) ~,\end{aligned}$$

Therefore, an equivalent PCA problem is

$$\operatorname*{argmax}_{U \in \mathbb{R}^{d,n} : U^\top U = I} \mathrm{trace}\left(U^\top \left(\sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top \right) U \right) ~.$$

The solution is to set U to be the leading eigenvectors of $A = \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top$.

Value of the objective

It is easy to see that

$$\min_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|^2 = \sum_{i=n+1}^{d} \lambda_i(A)$$

Centering

- It is a common practice to "center" the examples before applying PCA, namely:
- First calculate $\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$
- Then apply PCA on the vectors $(\mathbf{x}_1 - \boldsymbol{\mu}), \ldots, (\mathbf{x}_m - \boldsymbol{\mu})$
- This is also related to the interpretation of PCA as variance maximization (will be given in exercise)

Efficient implementation for $d \gg m$ and kernel PCA

- Recall: $A = \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top = X^\top X$ where $X \in \mathbb{R}^{m,d}$ is a matrix whose i'th row is $\mathbf{x}_i^\top$.
- Let $B = X X^\top$. That is, $B_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$
- If $B\mathbf{u} = \lambda \mathbf{u}$ then

$$A(X^\top \mathbf{u}) = X^\top X X^\top \mathbf{u} = X^\top B \mathbf{u} = \lambda (X^\top \mathbf{u})$$

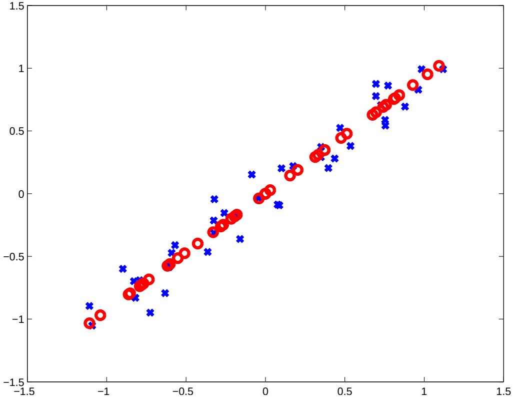
- So, $\frac{X^\top \mathbf{u}}{\|X^\top \mathbf{u}\|}$ is an eigenvector of A with eigenvalue λ
- We can therefore calculate the PCA solution by calculating the eigenvalues of B instead of A
- The complexity is $O(m^3 + m^2 d)$
- And, it can be computed using a kernel function

Pseudo code

PCA

input
A matrix of m examples $X \in \mathbb{R}^{m,d}$
number of components n
if $(m > d)$
$A = X^\top X$
Let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be the eigenvectors of A with largest eigenvalues
else
$B = XX^\top$
Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the eigenvectors of B with largest eigenvalues
for $i = 1, \ldots, n$ set $\mathbf{u}_i = \frac{1}{\|X^\top \mathbf{v}_i\|} X^\top \mathbf{v}_i$
output: $\mathbf{u}_1, \ldots, \mathbf{u}_n$

Demonstration

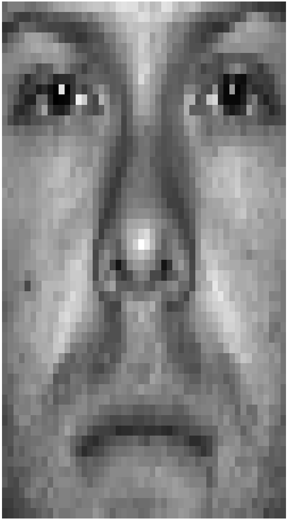


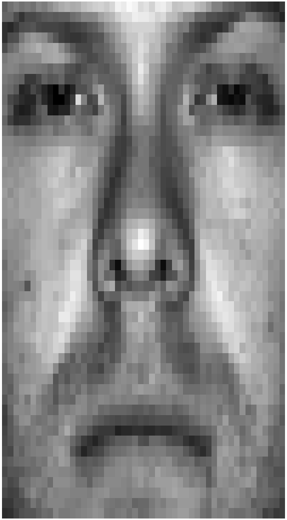
Demonstration

- 50×50 images from Yale dataset
- Before (left) and after reconstruction (right) to 10 dimensions

Demonstration

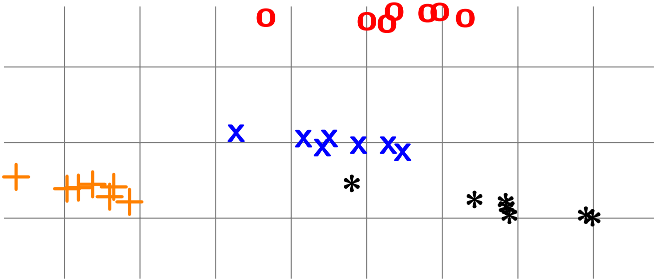
- Before and after





Demonstration

- Images after dim reduction to $\mathbb{R}^2$
- Different marks indicate different individuals



Outline

1. Principal Component Analysis (PCA)
2. Random Projections
3. Compressed Sensing

What is a successful dimensionality reduction?

- In PCA, we measured succes as squared distance between $\mathbf{x}$ and a reconstruction of $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$

What is a successful dimensionality reduction?

- In PCA, we measured succes as squared distance between $\mathbf{x}$ and a reconstruction of $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- In some cases, we don't care about reconstruction, all we care is that $\mathbf{y}_1, \ldots, \mathbf{y}_m$ will retain certain properties of $\mathbf{x}_1, \ldots, \mathbf{x}_m$

What is a successful dimensionality reduction?

- In PCA, we measured succes as squared distance between $\mathbf{x}$ and a reconstruction of $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- In some cases, we don't care about reconstruction, all we care is that $\mathbf{y}_1, \ldots, \mathbf{y}_m$ will retain certain properties of $\mathbf{x}_1, \ldots, \mathbf{x}_m$
- One option: do not distort distances. That is, we'd like that for all i, j, $\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\|$

What is a successful dimensionality reduction?

- In PCA, we measured succes as squared distance between $\mathbf{x}$ and a reconstruction of $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- In some cases, we don't care about reconstruction, all we care is that $\mathbf{y}_1, \ldots, \mathbf{y}_m$ will retain certain properties of $\mathbf{x}_1, \ldots, \mathbf{x}_m$
- One option: do not distort distances. That is, we'd like that for all i, j, $\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\|$
- Equivalently, we'd like that for all i, j, $\frac{\|W\mathbf{x}_i - W\mathbf{x}_j\|}{\|\mathbf{x}_i - \mathbf{x}_j\|} \approx 1$

What is a successful dimensionality reduction?

- In PCA, we measured succes as squared distance between $\mathbf{x}$ and a reconstruction of $\mathbf{x}$ from $\mathbf{y} = W\mathbf{x}$
- In some cases, we don't care about reconstruction, all we care is that $\mathbf{y}_1, \ldots, \mathbf{y}_m$ will retain certain properties of $\mathbf{x}_1, \ldots, \mathbf{x}_m$
- One option: do not distort distances. That is, we'd like that for all i, j, $\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\|$
- Equivalently, we'd like that for all i, j, $\frac{\|W\mathbf{x}_i - W\mathbf{x}_j\|}{\|\mathbf{x}_i - \mathbf{x}_j\|} \approx 1$
- Equivalently, we'd like that for all $\mathbf{x} \in Q$, where $Q = \{\mathbf{x}_i - \mathbf{x}_j : i, j \in [m]\}$, we'll have $\frac{\|W\mathbf{x}\|}{\|x\|} \approx 1$

Random Projections do not distort norms

- Random projection: The transformation $\mathbf{x} \mapsto W\mathbf{x}$, where W is a random matrix

Random Projections do not distort norms

- Random projection: The transformation $\mathbf{x} \mapsto W\mathbf{x}$, where W is a random matrix
- We'll analyze the distortion due to W s.t. $W_{i,j} \sim N(0, 1/n)$

Random Projections do not distort norms

- Random projection: The transformation $\mathbf{x} \mapsto W\mathbf{x}$, where W is a random matrix
- We'll analyze the distortion due to W s.t. $W_{i,j} \sim N(0, 1/n)$
- Let $\mathbf{w}_i$ be the i'th row of W. Then:

$$\mathbb{E}[\|W\mathbf{x}\|^2] = \sum_{i=1}^{n} \mathbb{E}[(\langle \mathbf{w}_i, \mathbf{x} \rangle)^2] = \sum_{i=1}^{n} \mathbf{x}^\top \mathbb{E}[\mathbf{w}_i \mathbf{w}_i^\top] \mathbf{x}$$

$$= n\mathbf{x}^\top \left(\frac{1}{n} I\right) \mathbf{x} = \|\mathbf{x}\|^2$$

Random Projections do not distort norms

- Random projection: The transformation $\mathbf{x} \mapsto W\mathbf{x}$, where W is a random matrix
- We'll analyze the distortion due to W s.t. $W_{i,j} \sim N(0, 1/n)$
- Let $\mathbf{w}_i$ be the i'th row of W. Then:

$$\mathbb{E}[\|W\mathbf{x}\|^2] = \sum_{i=1}^{n} \mathbb{E}[(\langle \mathbf{w}_i, \mathbf{x} \rangle)^2] = \sum_{i=1}^{n} \mathbf{x}^\top \mathbb{E}[\mathbf{w}_i \mathbf{w}_i^\top]\mathbf{x}$$

$$= n\mathbf{x}^\top \left(\frac{1}{n} I\right) \mathbf{x} = \|\mathbf{x}\|^2$$

- In fact, $\|W\mathbf{x}\|^2$ has a χ^2_n distribution, and using a measure concentration inequality it can be shown that

$$\mathbb{P}\left[\left| \frac{\|W\mathbf{x}\|^2}{\|\mathbf{x}\|^2} - 1 \right| > \epsilon \right] \le 2\, e^{-\epsilon^2 n/6}$$

Random Projections do not distort norms

- Applying the union bound over all vectors in Q we obtain:

Lemma (Johnson-Lindenstrauss lemma)

Let Q be a finite set of vectors in $\mathbb{R}^d$. Let $\delta \in (0,1)$ and n be an integer such that

$$\epsilon = \sqrt{\frac{6\,\log(2|Q|/\delta)}{n}} \le 3 \;.$$

Then, with probability of at least $1-\delta$ over a choice of a random matrix $W \in \mathbb{R}^{n,d}$ with $W_{i,j} \sim N(0,1/n)$, we have

$$\max_{\mathbf{x}\in Q} \left| \frac{\|W\mathbf{x}\|^2}{\|\mathbf{x}\|^2} - 1 \right| < \epsilon \;.$$

Outline

1. Principal Component Analysis (PCA)
2. Random Projections
3. Compressed Sensing

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis
- How to "store" $\mathbf{x}$?

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis
- How to "store" $\mathbf{x}$?
  - We can find $\boldsymbol{\alpha} = U^\top \mathbf{x}$ and then save the non-zero elements of $\boldsymbol{\alpha}$

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis
- How to "store" $\mathbf{x}$?
  - We can find $\boldsymbol{\alpha} = U^\top \mathbf{x}$ and then save the non-zero elements of $\boldsymbol{\alpha}$
  - Requires order of $s \log(d)$ storage

Compressed Sensing

- Prior assumption: $\mathbf{x} \approx U\boldsymbol{\alpha}$ where U is orthonormal and $\|\boldsymbol{\alpha}\|_0 \stackrel{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis
- How to "store" $\mathbf{x}$?
  - We can find $\boldsymbol{\alpha} = U^\top \mathbf{x}$ and then save the non-zero elements of $\boldsymbol{\alpha}$
  - Requires order of $s \log(d)$ storage
  - Why go to so much effort to acquire all the d coordinates of $\mathbf{x}$ when most of what we get will be thrown away? Can't we just directly measure the part that won't end up being thrown away?

Compressed Sensing

Informally, the main premise of compressed sensing is the following three "surprising" results:

1. It is possible to fully reconstruct any sparse signal if it was compressed by $\mathbf{x} \mapsto W\mathbf{x}$, where W is a matrix which satisfies a condition called Restricted Isoperimetric Property (RIP). A matrix that satisfies this property is guaranteed to have a low distortion of the norm of any sparse representable vector.

Compressed Sensing

Informally, the main premise of compressed sensing is the following three "surprising" results:

1. It is possible to fully reconstruct any sparse signal if it was compressed by $\mathbf{x} \mapsto W\mathbf{x}$, where W is a matrix which satisfies a condition called Restricted Isoperimetric Property (RIP). A matrix that satisfies this property is guaranteed to have a low distortion of the norm of any sparse representable vector.
2. The reconstruction can be calculated in polynomial time by solving a linear program.

Compressed Sensing

Informally, the main premise of compressed sensing is the following three "surprising" results:

1. It is possible to fully reconstruct any sparse signal if it was compressed by $\mathbf{x} \mapsto W\mathbf{x}$, where W is a matrix which satisfies a condition called Restricted Isoperimetric Property (RIP). A matrix that satisfies this property is guaranteed to have a low distortion of the norm of any sparse representable vector.
2. The reconstruction can be calculated in polynomial time by solving a linear program.
3. A random $n \times d$ matrix is likely to satisfy the RIP condition provided that n is greater than order of $s \log(d)$.

Restricted Isoperimetric Property (RIP)

A matrix $W \in \mathbb{R}^{n,d}$ is (ϵ, s)-RIP if for all $\mathbf{x} \neq 0$ s.t. $\|\mathbf{x}\|_0 \leq s$ we have

$$\left| \frac{\|W\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} - 1 \right| \leq \epsilon \ .$$

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let $\mathbf{x}$ be a vector s.t. $\|\mathbf{x}\|_0 \leq s$, let $\mathbf{y} = W\mathbf{x}$ and let $\tilde{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let $\mathbf{x}$ be a vector s.t. $\|\mathbf{x}\|_0 \le s$, let $\mathbf{y} = W\mathbf{x}$ and let $\tilde{\mathbf{x}} \in \mathrm{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

Proof.

- Assume, by way of contradiction, that $\tilde{\mathbf{x}} \neq \mathbf{x}$.

□

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ *and let* W *be a* $(\epsilon, 2s)$*-RIP matrix. Let* $\mathbf{x}$ *be a vector s.t.* $\|\mathbf{x}\|_0 \leq s$*, let* $\mathbf{y} = W\mathbf{x}$ *and let* $\tilde{\mathbf{x}} \in \mathrm{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$*. Then,* $\tilde{\mathbf{x}} = \mathbf{x}$*.*

Proof.

- Assume, by way of contradiction, that $\tilde{\mathbf{x}} \neq \mathbf{x}$.
- Since $\mathbf{x}$ satisfies the constraints in the optimization problem that defines $\tilde{\mathbf{x}}$ we clearly have that $\|\tilde{\mathbf{x}}\|_0 \leq \|\mathbf{x}\|_0 \leq s$.

□

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ *and let* W *be a* $(\epsilon, 2s)$*-RIP matrix. Let* $\mathbf{x}$ *be a vector s.t.* $\|\mathbf{x}\|_0 \leq s$*, let* $\mathbf{y} = W\mathbf{x}$ *and let* $\tilde{\mathbf{x}} \in \mathrm{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$*. Then,* $\tilde{\mathbf{x}} = \mathbf{x}$*.*

Proof.

- Assume, by way of contradiction, that $\tilde{\mathbf{x}} \neq \mathbf{x}$.
- Since $\mathbf{x}$ satisfies the constraints in the optimization problem that defines $\tilde{\mathbf{x}}$ we clearly have that $\|\tilde{\mathbf{x}}\|_0 \leq \|\mathbf{x}\|_0 \leq s$.
- Therefore, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \leq 2s$.

□

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let $\mathbf{x}$ be a vector s.t. $\|\mathbf{x}\|_0 \le s$, let $\mathbf{y} = W\mathbf{x}$ and let $\tilde{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

Proof.

- Assume, by way of contradiction, that $\tilde{\mathbf{x}} \neq \mathbf{x}$.
- Since $\mathbf{x}$ satisfies the constraints in the optimization problem that defines $\tilde{\mathbf{x}}$ we clearly have that $\|\tilde{\mathbf{x}}\|_0 \le \|\mathbf{x}\|_0 \le s$.
- Therefore, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \le 2s$.
- By RIP on $\mathbf{x} - \tilde{\mathbf{x}}$ we have $\left| \frac{\|W(\mathbf{x}-\tilde{\mathbf{x}})\|^2}{\|\mathbf{x}-\tilde{\mathbf{x}}\|^2} - 1 \right| \le \epsilon$

□

RIP matrices yield lossless compression for sparse vectors

Theorem

Let $\epsilon < 1$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let $\mathbf{x}$ be a vector s.t. $\|\mathbf{x}\|_0 \le s$, let $\mathbf{y} = W\mathbf{x}$ and let $\tilde{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{v}: W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0$. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

Proof.

- Assume, by way of contradiction, that $\tilde{\mathbf{x}} \neq \mathbf{x}$.
- Since $\mathbf{x}$ satisfies the constraints in the optimization problem that defines $\tilde{\mathbf{x}}$ we clearly have that $\|\tilde{\mathbf{x}}\|_0 \le \|\mathbf{x}\|_0 \le s$.
- Therefore, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \le 2s$.
- By RIP on $\mathbf{x} - \tilde{\mathbf{x}}$ we have $\left| \frac{\|W(\mathbf{x}-\tilde{\mathbf{x}})\|^2}{\|\mathbf{x}-\tilde{\mathbf{x}}\|^2} - 1 \right| \le \epsilon$
- But, since $W(\mathbf{x} - \tilde{\mathbf{x}}) = \mathbf{0}$ we get that $|0 - 1| \le \epsilon$. Contradiction.

□

Efficient reconstruction

- If we further assume that $\epsilon < \frac{1}{1+\sqrt{2}}$ then

$$\mathbf{x} \;=\; \underset{\mathbf{v}:W\mathbf{v}=\mathbf{y}}{\operatorname{argmin}} \|\mathbf{v}\|_0 \;=\; \underset{\mathbf{v}:W\mathbf{v}=\mathbf{y}}{\operatorname{argmin}} \|\mathbf{v}\|_1 \;.$$

Efficient reconstruction

- If we further assume that $\epsilon < \frac{1}{1+\sqrt{2}}$ then

$$\mathbf{x} = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0 = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_1 .$$

- The right-hand side is a linear programming problem

Efficient reconstruction

- If we further assume that $\epsilon < \frac{1}{1+\sqrt{2}}$ then

$$\mathbf{x} = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0 = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_1 .$$

- The right-hand side is a linear programming problem
- Summary: we can reconstruct all sparse vector efficiently based on $O(s\log(d))$ measurements

PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors

PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors
- PCA guarantee perfect recovery if all examples are in an n-dimensional subspace

PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors
- PCA guarantee perfect recovery if all examples are in an n-dimensional subspace
- Different prior knowledge:

PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors
- PCA guarantee perfect recovery if all examples are in an n-dimensional subspace
- Different prior knowledge:
  - If the data is $\mathbf{e}_1, \ldots, \mathbf{e}_d$, random projections will be perfect but PCA will fail

PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors
- PCA guarantee perfect recovery if all examples are in an n-dimensional subspace
- Different prior knowledge:
    - If the data is $\mathbf{e}_1, \ldots, \mathbf{e}_d$, random projections will be perfect but PCA will fail
    - If d is very large and data is exactly on an n-dim subspace. Then, PCA will be perfect but random projections might fail

Summary

- Linear dimensionality reduction $\mathbf{x} \mapsto W\mathbf{x}$
  - PCA: optimal if reconstruction is linear and error is squared distance
  - Random projections: preserves disctances
  - Random projections: exact reconstruction for sparse vectors (but with a non-linear reconstruction)
- Not covered: non-linear dimensionality reduction