Dimensionality Reduction
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:
\[ x \rightarrow Wx \]
where \( W \in \mathbb{R}^{n,d} \) and \( n < d \)
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)

\[ x \mapsto Wx \]

- What makes \( W \) a good matrix for dimensionality reduction?
Principal Component Analysis (PCA)

\[ x \mapsto Wx \]

- What makes \( W \) a good matrix for dimensionality reduction?
- Natural criterion: we want to be able to approximately recover \( x \) from \( y = Wx \)
Principal Component Analysis (PCA)

\[ x \mapsto Wx \]

- What makes \( W \) a good matrix for dimensionality reduction?
- Natural criterion: we want to be able to approximately recover \( x \) from \( y = Wx \)
- PCA:
Principal Component Analysis (PCA)

\[ x \mapsto Wx \]

- What makes \( W \) a good matrix for dimensionality reduction?
  - Natural criterion: we want to be able to approximately recover \( x \) from \( y = Wx \)

PCA:
  - Linear recovery: \( \tilde{x} = Uy = UWx \)
Principal Component Analysis (PCA)

\[ \mathbf{x} \mapsto \mathbf{Wx} \]

- What makes \( \mathbf{W} \) a good matrix for dimensionality reduction?

- Natural criterion: we want to be able to approximately recover \( \mathbf{x} \) from \( \mathbf{y} = \mathbf{Wx} \)

PCA:

- Linear recovery: \( \tilde{\mathbf{x}} = \mathbf{Uy} = \mathbf{UWx} \)
- Measures “approximate recovery” by averaged squared norm: given examples \( \mathbf{x}_1, \ldots, \mathbf{x}_m \), solve

\[
\arg\min_{\mathbf{W} \in \mathbb{R}^{n,d}, \mathbf{U} \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \| \mathbf{x}_i - \mathbf{UWx}_i \|^2
\]
Solving the PCA Problem

\[ \arg\min_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \| x_i - UW x_i \|^2 \]

Theorem
Let \( A = \sum_{i=1}^{m} x_i x_i^\top \) and let \( u_1, \ldots, u_n \) be the \( n \) leading eigenvectors of \( A \). Then, the solution to the PCA problem is to set the columns of \( U \) to be \( u_1, \ldots, u_n \) and to set \( W = U^\top \).
Solving the PCA Problem

$$\text{argmin}_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \| x_i - UW x_i \|^2$$

**Theorem**

Let \( A = \sum_{i=1}^{m} x_i x_i^\top \) and let \( u_1, \ldots, u_n \) be the \( n \) leading eigenvectors of \( A \). Then, the solution to the PCA problem is to set the columns of \( U \) to be \( u_1, \ldots, u_n \) 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 $x \mapsto UWx$ moves $x$ to this subspace
Proof main ideas

- $UW$ is of rank $n$, therefore its range is $n$ dimensional subspace, denoted $S$
- The transformation $x \mapsto UWx$ moves $x$ to this subspace
- The point in $S$ which is closest to $x$ is $VV^T x$, 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 $x \mapsto UWx$ moves $x$ to this subspace
- The point in $S$ which is closest to $x$ is $VV^\top x$, 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:

\[
\|\mathbf{x} - \mathbf{U}\mathbf{U}^\top \mathbf{x}\|^2 = \|\mathbf{x}\|^2 - 2\mathbf{x}^\top \mathbf{U}\mathbf{U}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{U}\mathbf{U}^\top \mathbf{U}\mathbf{U}^\top \mathbf{x} \\
= \|\mathbf{x}\|^2 - \mathbf{x}^\top \mathbf{U}\mathbf{U}^\top \mathbf{x} \\
= \|\mathbf{x}\|^2 - \text{trace}(\mathbf{U}^\top \mathbf{xx}^\top \mathbf{U}) ,
\]
Proof main ideas

Observe:

\[
\|x - UU^T x\|^2 = \|x\|^2 - 2x^T UU^T x + x^T UU^T UU^T x
= \|x\|^2 - x^T UU^T x
= \|x\|^2 - \text{trace}(U^T xx^T U),
\]

Therefore, an equivalent PCA problem is

\[
\arg\max_{U \in \mathbb{R}^{d,n}: U^T U = I} \text{trace} \left( U^T \left( \sum_{i=1}^{m} x_i x_i^T \right) U \right).
\]
Observe:

\[
\| \mathbf{x} - \mathbf{U} \mathbf{U}^\top \mathbf{x} \|^2 = \| \mathbf{x} \|^2 - 2 \mathbf{x}^\top \mathbf{U} \mathbf{U}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{U} \mathbf{U}^\top \mathbf{U} \mathbf{U}^\top \mathbf{x}
\]

\[
= \| \mathbf{x} \|^2 - \mathbf{x}^\top \mathbf{U} \mathbf{U}^\top \mathbf{x}
\]

\[
= \| \mathbf{x} \|^2 - \text{trace}(\mathbf{U}^\top \mathbf{x} \mathbf{x}^\top \mathbf{U})
\]

Therefore, an equivalent PCA problem is

\[
\arg\max_{\mathbf{U} \in \mathbb{R}^{d,n}: \mathbf{U}^\top \mathbf{U} = \mathbf{I}} \quad \text{trace} \left( \mathbf{U}^\top \left( \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{U} \right)
\]

The solution is to set \( \mathbf{U} \) to be the leading eigenvectors of \( \mathbf{A} = \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top \).
It is easy to see that

$$\min_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|x_i - UWx_i\|^2 = \sum_{i=n+1}^{d} \lambda_i(A)$$
It is a common practice to “center” the examples before applying PCA, namely:

First calculate \( \mu = \frac{1}{m} \sum_{i=1}^{m} x_i \)

Then apply PCA on the vectors \((x_1 - \mu), \ldots, (x_m - \mu)\)

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: $A = \sum_{i=1}^{m} x_i x_i^\top = X^\top X$ where $X \in \mathbb{R}^{m,d}$ is a matrix whose $i$'th row is $x_i^\top$.

- Let $B = XX^\top$. That is, $B_{i,j} = \langle x_i, x_j \rangle$

- If $Bu = \lambda u$ then

$$A(X^\top u) = X^\top XX^\top u = X^\top Bu = \lambda(X^\top u)$$

- So, $\frac{X^\top u}{\|X^\top u\|}$ is an eigenvector of $A$ with eigenvalue $\lambda$

- We can therefore calculate the PCA solution by calculating the eigenvalues of $B$ instead of $A$

- The complexity is $O(m^3 + m^2d)$

- 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 $u_1, \ldots, u_n$ be the eigenvectors of $A$ with largest eigenvalues
else
  $B = XX^\top$
  Let $v_1, \ldots, v_n$ be the eigenvectors of $B$ with largest eigenvalues
for $i = 1, \ldots, n$ set $u_i = \frac{1}{\|X^\top v_i\|} X^\top v_i$

output: $u_1, \ldots, u_n$
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

![Graph showing images after dimensionality reduction with various symbols representing different individuals.](image-url)
Outline

1. Principal Component Analysis (PCA)

2. Random Projections

3. Compressed Sensing
What is a successful dimensionality reduction?

- In PCA, we measured success as squared distance between $x$ and a reconstruction of $x$ from $y = Wx$
What is a successful dimensionality reduction?

- In PCA, we measured success as squared distance between $x$ and a reconstruction of $x$ from $y = Wx$.
- In some cases, we don’t care about reconstruction, all we care is that $y_1, \ldots, y_m$ will retain certain properties of $x_1, \ldots, x_m$.
What is a successful dimensionality reduction?

- In PCA, we measured success as squared distance between \( x \) and a reconstruction of \( x \) from \( y = Wx \).
- In some cases, we don’t care about reconstruction, all we care is that \( y_1, \ldots, y_m \) will retain certain properties of \( x_1, \ldots, x_m \).
- One option: do not distort distances. That is, we’d like that for all \( i, j, \|x_i - x_j\| \approx \|y_i - y_j\| \).
What is a successful dimensionality reduction?

- In PCA, we measured success as squared distance between $x$ and a reconstruction of $x$ from $y = Wx$
- In some cases, we don’t care about reconstruction, all we care is that $y_1, \ldots, y_m$ will retain certain properties of $x_1, \ldots, x_m$
- One option: do not distort distances. That is, we’d like that for all $i, j$, $\|x_i - x_j\| \approx \|y_i - y_j\|$
- Equivalently, we’d like that for all $i, j$, $\frac{\|Wx_i - Wx_j\|}{\|x_i - x_j\|} \approx 1$
What is a successful dimensionality reduction?

- In PCA, we measured success as squared distance between $x$ and a reconstruction of $x$ from $y = Wx$

- In some cases, we don’t care about reconstruction, all we care is that $y_1, \ldots, y_m$ will retain certain properties of $x_1, \ldots, x_m$

- One option: do not distort distances. That is, we’d like that for all $i, j$, $\|x_i - x_j\| \approx \|y_i - y_j\|$

- Equivalently, we’d like that for all $i, j$, $\frac{\|Wx_i - Wx_j\|}{\|x_i - x_j\|} \approx 1$

- Equivalently, we’d like that for all $x \in Q$, where $Q = \{x_i - x_j : i, j \in [m]\}$, we’ll have $\frac{\|Wx\|}{\|x\|} \approx 1$
Random Projections do not distort norms

- **Random projection**: The transformation $x \mapsto Wx$, 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 $w_i$ be the $i$'th row of $W$. Then:

$$
E[\|Wx\|^2] = n \sum_{i=1}^{\|x\|^2} E[(\langle w_i, x \rangle)^2] = n \sum_{i=1}^{\|x\|^2} x^\top E[w_i w_i^\top] x = n x^\top \left(1/nI\right) x = \|x\|^2
$$

In fact, $\|Wx\|^2$ has a $\chi^2_n$ distribution, and using a measure concentration inequality it can be shown that

$$
P[\|Wx\|^2 / \|x\|^2 - 1 > \epsilon] \leq 2e^{-\epsilon^2 n/6}
$$
Random Projections do not distort norms

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

\[
\mathbb{E}[\|\mathbf{Wx}\|_2^2] = n \sum_{i=1}^{n} \mathbb{E}[(\langle \mathbf{w}_i, \mathbf{x} \rangle )^2] = n \sum_{i=1}^{n} \mathbf{x}^\top \mathbb{E}[\mathbf{w}_i \mathbf{w}_i^\top] \mathbf{x} = \mathbf{x}^\top (\frac{1}{n} \mathbf{I}) \mathbf{x} = \|\mathbf{x}\|_2^2,
\]

In fact, $\|\mathbf{Wx}\|_2^2$ has a $\chi^2_n$ distribution, and using a measure concentration inequality it can be shown that

\[
P[\|\mathbf{Wx}\|_2^2/\|\mathbf{x}\|_2^2 - 1 > \epsilon] \leq 2e^{-\epsilon^2 n / 6}.
\]
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. $W_{i,j} \sim \mathcal{N}(0, 1/n)$.
- Let $\mathbf{w}_i$ be the $i$'th row of $W$. Then:

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

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

$$
P[\|W\mathbf{x}\|^2 / \|\mathbf{x}\|^2 - 1 > \epsilon] \leq 2e^{-\epsilon^2 n / 6}$$
Random Projections do not distort norms

- **Random projection:** The transformation \( \mathbf{x} \mapsto \mathbf{Wx} \), where \( \mathbf{W} \) is a random matrix.

- We’ll analyze the distortion due to \( \mathbf{W} \) s.t. \( W_{i,j} \sim \mathcal{N}(0, 1/n) \).

- Let \( \mathbf{w}_i \) be the \( i \)'th row of \( \mathbf{W} \). Then:

  \[
  \mathbb{E}[\|\mathbf{Wx}\|^2] = \sum_{i=1}^{n} \mathbb{E}[\langle \mathbf{w}_i, \mathbf{x} \rangle^2] = \sum_{i=1}^{n} \mathbf{x}^\top \mathbb{E}[\mathbf{w}_i \mathbf{w}_i^\top] \mathbf{x} \\
  = n \mathbf{x}^\top \left( \frac{1}{n} \mathbf{I} \right) \mathbf{x} = \|\mathbf{x}\|^2
  \]

- In fact, \( \|\mathbf{Wx}\|^2 \) has a \( \chi_n^2 \) distribution, and using a measure concentration inequality it can be shown that

  \[
  \mathbb{P} \left[ \left| \frac{\|\mathbf{Wx}\|^2}{\|\mathbf{x}\|^2} - 1 \right| > \epsilon \right] \leq 2 e^{-\epsilon^2 n/6}
  \]
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

$$\epsilon = \sqrt{6 \log(2|Q|/\delta)} \leq 3.$$  

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

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

Shai Shalev-Shwartz (Hebrew U)
Outline

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

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

  - E.g.: natural images are approximately sparse in a wavelet basis
  - How to “store” $x$?
    - We can find $\alpha = U^\top x$ and then save the non-zero elements of $\alpha$
    - Requires order of $s \log(d)$ storage

  - Why go to so much effort to acquire all the $d$ coordinates of $x$ 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

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

• Prior assumption: \( x \approx U \alpha \) where \( U \) is orthonormal and
  \[ \| \alpha \|_0 \overset{\text{def}}{=} |\{ i : \alpha_i \neq 0 \}| \leq s \text{ for some } s \ll d \]
  
• E.g.: natural images are approximately sparse in a wavelet basis

• How to “store” \( x \)?
  
  We can find \( \alpha = U^\top x \) and then save the non-zero elements of \( \alpha \)
  
  Requires order of \( s \log(d) \) storage

  Why go to so much effort to acquire all the \( d \) coordinates of \( x \) 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

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

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

- Prior assumption: $x \approx U\alpha$ where $U$ is orthonormal and $\|\alpha\|_0 \overset{\text{def}}{=} |\{i : \alpha_i \neq 0\}| \leq s$ for some $s \ll d$
- E.g.: natural images are approximately sparse in a wavelet basis
- How to “store” $x$?
  - We can find $\alpha = U^T x$ and then save the non-zero elements of $\alpha$
  - Requires order of $s \log(d)$ storage
  - Why go to so much effort to acquire all the $d$ coordinates of $x$ 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?
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 $x \mapsto Wx$, 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.
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 $x \mapsto Wx$, 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 $\text{order of } s \log(d)$. 
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 $x \mapsto Wx$, 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 $s \log(d)$. 
A matrix $W \in \mathbb{R}^{n,d}$ is $(\epsilon, s)$-RIP if for all $x \neq 0$ s.t. $\|x\|_0 \leq s$ we have

$$\left| \frac{\|Wx\|_2^2}{\|x\|_2^2} - 1 \right| \leq \epsilon.$$
RIP matrices yield lossless compression for sparse vectors

**Theorem**

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

Proof.

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

By RIP on $x - \tilde{x}$ we have $|\|W(x - \tilde{x})\|_2| \leq \epsilon$. But, since $W(x - \tilde{x}) = 0$ we get that $|0 - 1| \leq \epsilon$. Contradiction.
Theorem

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

Proof.

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

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

Proof.

- Assume, by way of contradiction, that $\tilde{x} \neq x$.
- Since $x$ satisfies the constraints in the optimization problem that defines $\tilde{x}$ we clearly have that $\|\tilde{x}\|_0 \leq \|x\|_0 \leq s$. 
RIP matrices yield lossless compression for sparse vectors

**Theorem**

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

**Proof.**

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

Shai Shalev-Shwartz (Hebrew U)
RIP matrices yield lossless compression for sparse vectors

**Theorem**

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

**Proof.**

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

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

Proof.

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

$$x = \arg\min_{v: Wv = y} \|v\|_0 = \arg\min_{v: Wv = y} \|v\|_1.$$
Efficient reconstruction

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

$$x = \arg\min_{v: Wv = y} \|v\|_0 = \arg\min_{v: Wv = y} \|v\|_1.$$ 

- The right-hand side is a linear programming problem
Efficient reconstruction

- If we further assume that $\epsilon < \frac{1}{1+\sqrt{2}}$ then
  
  $$
  x = \arg\min_v \|v\|_0 = \arg\min_v \|v\|_1.
  $$

- The right-hand side is a linear programming problem

- **Summary:** we can reconstruct all sparse vector efficiently based on $O(s \log(d))$ measurements
Random projections guarantee perfect recovery for all $O(n / \log(d))$-sparse vectors.
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 guarantees perfect recovery if all examples are in an $n$-dimensional subspace.
- Different prior knowledge:
  - If the data is $e_1, \ldots, e_d$, random projections will be perfect but PCA will fail.
  - If $d$ is very large and data is exactly on an $n$-dimensional subspace, then PCA will be perfect but random projections might fail.
PCA vs. Random Projections

- Random projections guarantee perfect recovery for all $O(n/\log(d))$-sparse vectors.
- PCA guarantees perfect recovery if all examples are in an $n$-dimensional subspace.
- Different prior knowledge:
  - If the data is $e_1, \ldots, e_d$, 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 $\mathbf{x} \mapsto \mathbf{Wx}$
  - PCA: optimal if reconstruction is linear and error is squared distance
  - Random projections: preserves distances
  - Random projections: exact reconstruction for sparse vectors (but with a non-linear reconstruction)

- Not covered: non-linear dimensionality reduction