Today.

Modelling.

Today.

Modelling.

An Analysis of the Power of PCA.

Today.

Modelling.

An Analysis of the Power of PCA.

Musing (rant?) about algorithms in the real world.

Two populations.

DNA data:

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3 .., x_n$.

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3 .., x_n$.

Which population?

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3.., x_n$.

Which population?

Comment: snps could be movie preferences, populations could be types.

Two populations.

DNA data:

human1: A $\cdots$ C $\cdots$ T $\cdots$ A
human2: C $\cdots$ C $\cdots$ A $\cdots$ T
human3: A $\cdots$ G $\cdots$ T $\cdots$ T

Single Nucleotide Polymorphism.

Same population?

Model: same populution breeds.

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3.., x_n$.

Which population?

Comment: snps could be movie preferences, populations could be types.
E.g., republican/democrat, shopper/saver.

Which population?

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Which population?

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3.., x_n$.

Which population?

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3 .., x_n$.

Population 1: snp i: Pr[$x_i = 1$] = $p_i^{(1)}$
Population 2: snp i: Pr[$x_i = 1$] = $p_i^{(2)}$

Which population?

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3.., x_n$.

Population 1: snp i: Pr[$x_i = 1$] = $p_i^{(1)}$
Population 2: snp i: Pr[$x_i = 1$] = $p_i^{(2)}$

Simpler Calculation:

Which population?

Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3 .., x_n$.

Population 1: snp i: $\Pr[x_i = 1]$ = $p_i^{(1)}$
Population 2: snp i: $\Pr[x_i = 1]$ = $p_i^{(2)}$

Simpler Calculation:

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Which population?

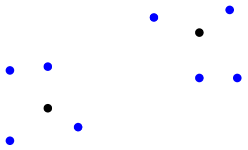
Population 1: snp 843: Pr[A] = .4 , Pr[T] = .6
Population 2: snp 843: Pr[A] = .6 , Pr[T] = .4

Individual: $x_1, x_2, x_3.., x_n$.

Population 1: snp i: Pr$[x_i = 1]$ = $p_i^{(1)}$
Population 2: snp i: Pr$[x_i = 1]$ = $p_i^{(2)}$

Simpler Calculation:

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.



Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
  x in population 1.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?

x in population 1.

$$E[(x-\mu_1)^2] = d\sigma^2$$

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?

x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2$.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
 x in population 1.

 $E[(x-\mu_1)^2] = d\sigma^2$
 $E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1 - \mu_2)^2$.

If $(\mu_1 - \mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
  x in population 1.

$$E[(x-\mu_1)^2] = d\sigma^2$$
$$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2.$$

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
  $\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2$.

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
$\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2.$

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
$\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?
Roughly $d\sigma^4$.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2$.

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
$\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?
Roughly $d\sigma^4$.
Signal is difference between expecations.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1 - \mu_2)^2$.

If $(\mu_1 - \mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
$\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?
Roughly $d\sigma^4$.
Signal is difference between expecations.
roughly $d\varepsilon^2$

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
x in population 1.

$E[(x-\mu_1)^2] = d\sigma^2$
$E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2$.

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
$\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?
Roughly $d\sigma^4$.
Signal is difference between expecations.
roughly $d\varepsilon^2$
Signal $>>$ Noise. $\leftrightarrow d\varepsilon^2 >> \sqrt{d}\sigma^2$.

Gaussians

Population 1: Gaussion with mean $\mu_1 \in R^d$, std dev. σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, std dev. σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

How many snps to collect to determine population for individual x?
  x in population 1.

  $E[(x-\mu_1)^2] = d\sigma^2$
  $E[(x-\mu_2)^2] \geq (d-1)\sigma^2 + (\mu_1-\mu_2)^2$.

If $(\mu_1-\mu_2)^2 = d\varepsilon^2 >> \sigma^2$, then different.
  $\rightarrow$ take $d >> \sigma^2/\varepsilon^2$

Variance of estimator?
  Roughly $d\sigma^4$.
  Signal is difference between expecations.
    roughly $d\varepsilon^2$
  Signal $>>$ Noise. $\leftrightarrow d\varepsilon^2 >> \sqrt{d}\sigma^2$.
Need $d >> \sigma^4/\varepsilon^4$.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x-\mu_1)\cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x-\mu_2)\cdot v)^2] \geq (\mu_1-\mu_2)^2$ if x is population 2.

Std deviation is σ^2!

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

No loss in signal!

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

No loss in signal!

$d\varepsilon^2 >> \sigma^2$.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x-\mu_1)\cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x-\mu_2)\cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

No loss in signal!

$d\varepsilon^2 >> \sigma^2$.
$\rightarrow d >> \sigma^2/\varepsilon^2$

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

No loss in signal!

$d\varepsilon^2 >> \sigma^2$.
$\rightarrow d >> \sigma^2/\varepsilon^2$

Versus $d >> \sigma^4/\varepsilon^4$.

Projection

Population 1: Gaussion with mean $\mu_1 \in R^d$, variance σ in each dim.
Population 2: Gaussion with mean $\mu_2 \in R^d$, variance σ in each dim.

Difference between humans σ per snp.
Difference between populations ε per snp.

Project x onto unit vector v in direction $\mu_2 - \mu_1$.

$E[((x - \mu_1) \cdot v)^2] = \sigma^2$ if x is population 1.

$E[((x - \mu_2) \cdot v)^2] \geq (\mu_1 - \mu_2)^2$ if x is population 2.

Std deviation is σ^2! versus $\sqrt{d}\sigma^2$!

No loss in signal!

$d\varepsilon^2 >> \sigma^2$.
$\rightarrow d >> \sigma^2/\varepsilon^2$

Versus $d >> \sigma^4/\varepsilon^4$.

A quadratic difference in amount of data!

Don’t know much about...

Don’t know μ_1 or μ_2?

Without the means?

Sample of n people.

Without the means?

Sample of n people.

Some (say half) from population 1,

Without the means?

Sample of n people.

Some (say half) from population 1,

some from population 2.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Without the means?

Sample of n people.
  Some (say half) from population 1,
    some from population 2.

Which are which?

Near Neighbors Approach

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.

Without the means?

Sample of *n* people.
  Some (say half) from population 1,
    some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
  Cluster using threshold.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$
Where x's from one population, y's from other.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Noise is proportional to $\sqrt{d}\sigma^2$.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1,x_2)] - E[d(x_1,y_1)]$
should be larger than noise in $d(x,y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Noise is proportional to $\sqrt{d}\sigma^2$.

$d >> \sigma^4/\varepsilon^4$ $\quad \rightarrow$ same type people closer to each other.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1,x_2)] - E[d(x_1,y_1)]$
should be larger than noise in $d(x,y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Noise is proportional to $\sqrt{d}\sigma^2$.

$d >> \sigma^4/\varepsilon^4 \quad \rightarrow$ same type people closer to each other.

$d >> (\sigma^4/\varepsilon^4)\log n$ suffices for threshold clustering.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Noise is proportional to $\sqrt{d}\sigma^2$.

$d >> \sigma^4/\varepsilon^4$ $\rightarrow$ same type people closer to each other.

$d >> (\sigma^4/\varepsilon^4) \log n$ suffices for threshold clustering.

$\log n$ factor for union bound over $\binom{n}{2}$ pairs.

Without the means?

Sample of n people.
Some (say half) from population 1,
some from population 2.

Which are which?

Near Neighbors Approach

Compute Euclidean distance squared.
Cluster using threshold.

Signal $E[d(x_1, x_2)] - E[d(x_1, y_1)]$
should be larger than noise in $d(x, y)$
Where x's from one population, y's from other.

Signal is proportional $d\varepsilon^2$.

Noise is proportional to $\sqrt{d}\sigma^2$.

$d >> \sigma^4/\varepsilon^4 \quad \rightarrow$ same type people closer to each other.

$d >> (\sigma^4/\varepsilon^4) \log n$ suffices for threshold clustering.

$\log n$ factor for union bound over $\binom{n}{2}$ pairs.

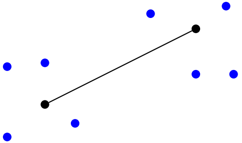
Best one can do?

Principal components analysis.

Remember Projection!

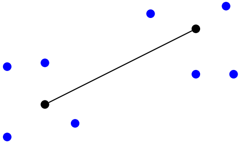
Principal components analysis.

Remember Projection!



Principal components analysis.

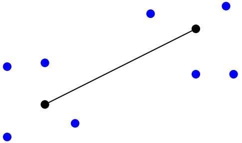
Remember Projection!



Don't know μ_1 or μ_2?

Principal components analysis.

Remember Projection!

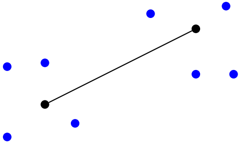


Don't know μ_1 or μ_2?

Principal component analysis:

Principal components analysis.

Remember Projection!



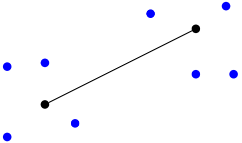
Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

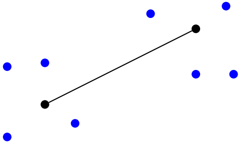
Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

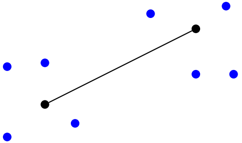
Find direction, v, of maximum variance.

Maximize $\sum (x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

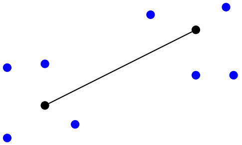
Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

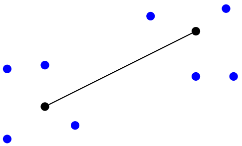
Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

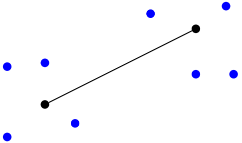
Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

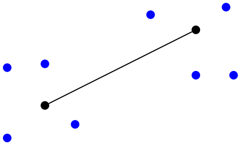
Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

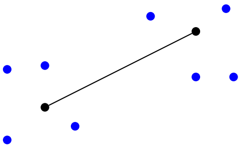
$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

Need $d >> \sigma^2/\varepsilon^2$ at least.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

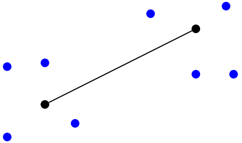
$\propto nd\varepsilon^2$.

Need $d >> \sigma^2/\varepsilon^2$ at least.

When will PCA pick correct direction with good probability?

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

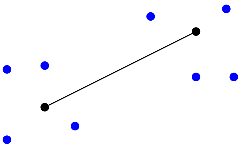
Need $d >> \sigma^2/\varepsilon^2$ at least.

When will PCA pick correct direction with good probability?

Union bound over directions.

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

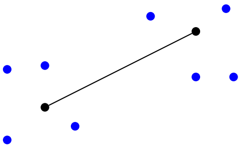
Need $d >> \sigma^2/\varepsilon^2$ at least.

When will PCA pick correct direction with good probability?

Union bound over directions. How many directions?

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum (x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

Need $d >> \sigma^2/\varepsilon^2$ at least.

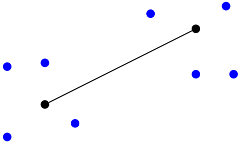
When will PCA pick correct direction with good probability?

Union bound over directions. How many directions?

Infinity

Principal components analysis.

Remember Projection!



Don't know μ_1 or μ_2?

Principal component analysis:

Find direction, v, of maximum variance.

Maximize $\sum(x \cdot v)^2$ (zero center the points)

Recall: $(x \cdot v)^2$ could determine population.

Typical direction variance. $n\sigma^2$.

Direction along $\mu_1 - \mu_2$,

$\propto n(\mu_1 - \mu_2)^2$.

$\propto nd\varepsilon^2$.

Need $d >> \sigma^2/\varepsilon^2$ at least.

When will PCA pick correct direction with good probability?

Union bound over directions. How many directions?

Infinity and beyond!

Nets

“δ - Net”.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Signal (exp. projection): $\propto nd\varepsilon^2$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Signal (exp. projection): $\propto nd\varepsilon^2$.
Noise (std dev.): $\sqrt{n}\sigma^2$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Signal (exp. projection): $\propto nd\varepsilon^2$.
Noise (std dev.): $\sqrt{n}\sigma^2$.

$nd >> (\sigma^4/\varepsilon^4) \log d$ and $d >> \sigma^2/\varepsilon^2$ works.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Signal (exp. projection): $\propto nd\varepsilon^2$.
Noise (std dev.): $\sqrt{n}\sigma^2$.

$nd >> (\sigma^4/\varepsilon^4) \log d$ and $d >> \sigma^2/\varepsilon^2$ works.

Nearest neighbor works with very high $d > \sigma^4/\varepsilon^4$.

Nets

"δ - Net".

Set $\mathscr{D}$ of directions

where all others, v, are close to $x \in \mathscr{D}$.

$x \cdot v \geq 1 - \delta$.

δ- Net:

$[\cdots, i\delta/d, \cdots]$ integers $i \in [-d/\delta, d\delta]$.

Total of $N \propto \left(\frac{d}{\delta}\right)^{O(d)}$ vectors in net.

Signal $>>$ Noise times $\log N = O(d \log \frac{d}{\delta})$ to isolate direction.

$\log N$ is due to union bound over vectors in net.

Signal (exp. projection): $\propto nd\varepsilon^2$.
Noise (std dev.): $\sqrt{n}\sigma^2$.

$nd >> (\sigma^4/\varepsilon^4) \log d$ and $d >> \sigma^2/\varepsilon^2$ works.

Nearest neighbor works with very high $d > \sigma^4/\varepsilon^4$.

PCA can reduce d to "knowing centers" case, with reasonable number of sample points.

PCA calculation.

Matrix A where rows are points.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

> Av are projections onto v.
> $vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

$Bv = \lambda v$ for maximum λ.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

$Bv = \lambda v$ for maximum λ.
$\rightarrow vBv = \lambda$ for unit v.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

- Av are projections onto v.
- $vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

- $Bv = \lambda v$ for maximum λ.
- $\rightarrow vBv = \lambda$ for unit v.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

$Bv = \lambda v$ for maximum λ.
$\rightarrow vBv = \lambda$ for unit v.

Eigenvectors form orthonormal basis.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

- Av are projections onto v.
- $vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

- $Bv = \lambda v$ for maximum λ.
- $\rightarrow vBv = \lambda$ for unit v.

Eigenvectors form orthonormal basis.

Any other vector $av + x$, $x \cdot v = 0$

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

$Bv = \lambda v$ for maximum λ.
$\rightarrow vBv = \lambda$ for unit v.

Eigenvectors form orthonormal basis.

Any other vector $av + x$, $x \cdot v = 0$
x is composed of possibly smaller eigenvalue vectors.

PCA calculation.

Matrix A where rows are points.

First eigenvector of $B = A^T A$ is maximum variance direction.

Av are projections onto v.
$vBv = (vA)^T(Av)$ is $\sum_x (x \cdot v)^2$.

First eigenvector, v, of B maximizes $x^T Bx$.

$Bv = \lambda v$ for maximum λ.
$\rightarrow vBv = \lambda$ for unit v.

Eigenvectors form orthonormal basis.

Any other vector $av + x$, $x \cdot v = 0$
x is composed of possibly smaller eigenvalue vectors.

$\rightarrow vBv \geq (av + x)B(av + x)$ for unit v, $av + x$.

Computing eigenvalues.

Power method:

Computing eigenvalues.

Power method:
Choose random x.

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

Computing eigenvalues.

Power method:
  Choose random x.
  Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Computing eigenvalues.

Power method:
  Choose random x.
  Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector.

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector. Multiply by A^T (direction between means),

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector. Multiply by A^T (direction between means), multiply by A (project points),

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector. Multiply by A^T (direction between means), multiply by A (project points), cluster (round to +1/-1 vector.)

Computing eigenvalues.

Power method:
Choose random x.
Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
Choose random partition.
Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector. Multiply by A^T (direction between means), multiply by A (project points), cluster (round to +1/-1 vector.)

Sort of repeatedly multiplying by AA^T.

Computing eigenvalues.

Power method:
  Choose random x.
  Repeat: Let $x = Bx$. Scale x to unit vector.

$x = a_1 v_1 + a_2 v_2 + \cdots$

$x_t \propto B^t x = a_1 \lambda_1^t v_1 + a_2 \lambda_2^t v_2 + \cdots$

Mostly v_1 after a while since $\lambda_1^t >> \lambda_2^t$.

Cluster Algorithm:
  Choose random partition.
  Repeat: Compute means of partition. Project, cluster.

Choose random $+1/-1$ vector. Multiply by A^T (direction between means), multiply by A (project points), cluster (round to +1/-1 vector.)

Sort of repeatedly multiplying by AA^T. Power method.

Sum up.

Clustering mixture of gaussians.

Sum up.

Clustering mixture of gaussians.

Near Neighbor works with sufficient data.

Sum up.

Clustering mixture of gaussians.

Near Neighbor works with sufficient data.

Projection onto subspace of means is better.

Sum up.

Clustering mixture of gaussians.

Near Neighbor works with sufficient data.

Projection onto subspace of means is better.

Principal compent analysis can find subspace of means.

Sum up.

Clustering mixture of gaussians.

Near Neighbor works with sufficient data.

Projection onto subspace of means is better.

Principal compent analysis can find subspace of means.

Power method computes principal component.

Sum up.

Clustering mixture of gaussians.

Near Neighbor works with sufficient data.

Projection onto subspace of means is better.

Principal compent analysis can find subspace of means.

Power method computes principal component.

Generic clustering algorithm is rounded version of power method.

See you on Tuesday.