Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}$ that should satisfy	"$Y \approx \check{f}(X)$" (Metric to be defined)

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}$ that should satisfy	"$Y \approx \check{f}(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}$ that should satisfy	"$Y \approx \check{f}(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	*Dataset*: $D = \{x_1 \ldots, x_n\}$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}_D$ that should satisfy	"$Y \approx \check{f}_D(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	*Dataset*: $D = \{x_1 \ldots, x_n\}$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised

Look for $\hat{f}_D$ that should satisfy

$$"Y \approx \hat{f}_D(X)"$$

Training dataset:

$$D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$$

Unsupervised

Dataset:

$$D = \{x_1 \ldots, x_n\}$$

Often introduce the *data matrix*:

$$X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised

Look for $\hat{f}_D$ that should satisfy

Training dataset:

$$D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$$

Unsupervised

"$Y \approx \check{f}_D(X)$"

X notation ambiguous!

Dataset:

$$D = \{x_1 \ldots, x_n\}$$

Often introduce the *data matrix*:

$$X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised

Look for $\hat{f}_D$ that should satisfy

$$"Y \approx \hat{f}_D(X)"$$

Training dataset:

$$D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$$

Unsupervised

Dataset:

$$D = \{x_1 \ldots, x_n\}$$

Often introduce the *data matrix*:

$$X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}_D$ that should satisfy	"$Y \approx \check{f}_D(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	*Dataset*: $D = \{x_1 \ldots, x_n\}$
	Often introduce the *data matrix*: $X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$
Regression or Classification	Just Classification*: "Clustering"

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}_D$ that should satisfy	"$Y \approx \check{f}_D(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	*Dataset*: $D = \{x_1 \ldots, x_n\}$
	Often introduce the *data matrix*: $X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$
Regression or Classification	Just Classification*: "Clustering"
Given a new data x: label is $f_D(x)$	Labels: $f_{D'}(x_1), \ldots, f_{D'}(x_n)$

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised	Unsupervised
Look for $\hat{f}_D$ that should satisfy	"$Y \approx \check{f}_D(X)$"
Training dataset: $D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$	*Dataset*: $D = \{x_1 \ldots, x_n\}$
	Often introduce the *data matrix*: $X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$
Regression or Classification	Just Classification*: "Clustering"
Given a new data x: label is $f_D(x)$	Labels: $f_{D'}(x_1), \ldots, f_{D'}(x_n)$

Semi-supervised

*the term "unsupervised regression" appears in dimension reduction problems

Statistical Learning STA4042

SUPERVISED VS. UNSUPERVISED

SCHOOL OF DATA SCIENCE

What is the difference?

Supervised

Look for $\hat{f}_D$ that should satisfy

"$Y \approx \hat{f}_D(X)$"

Training dataset:

$D = \{(x_1, y_n), \ldots, (x_n, y_n)\}$

Regression or Classification

Given a new data x:
label is $f_D(x)$

Unsupervised

"$Y \approx \check{f}_D(X)$"

Dataset:

$D = \{x_1 \ldots, x_n\}$

Often introduce the *data matrix*:

$X \equiv (x_1 \ldots, x_n) \in \mathbb{R}^{p \times n}$

Just Classification*: "Clustering"

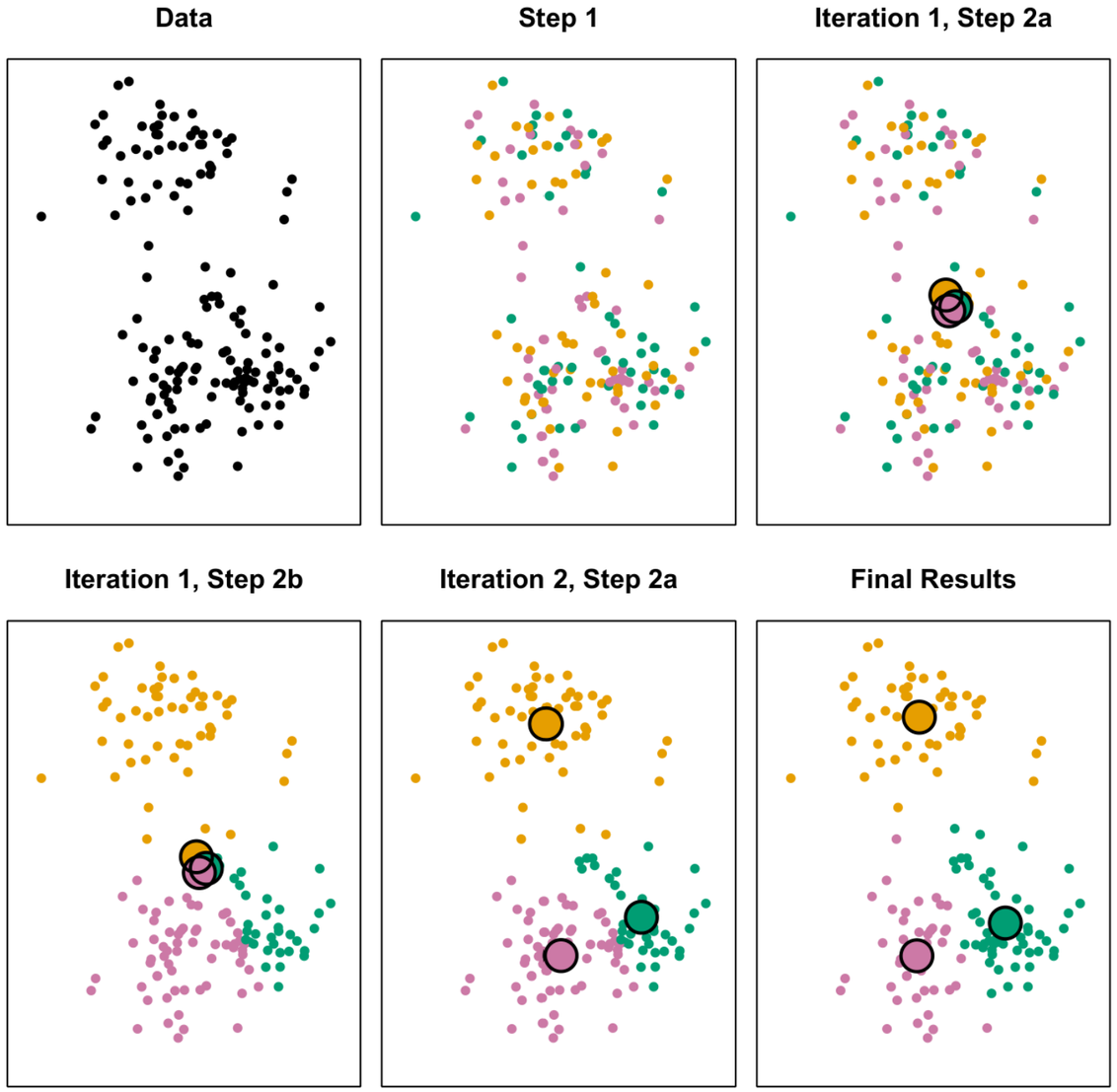
Labels: $f_{D'}(x_1), \ldots, f_{D'}(x_n)$

Semi-supervised

Given a set of labeled data D, and a set of unlabeled data D'
For new test data $x \in \mathbb{R}^p$, label: $f_{D,D'}(x)$.

*the term "unsupervised regression" appears in dimension reduction problems

Clustering method 1: k-means



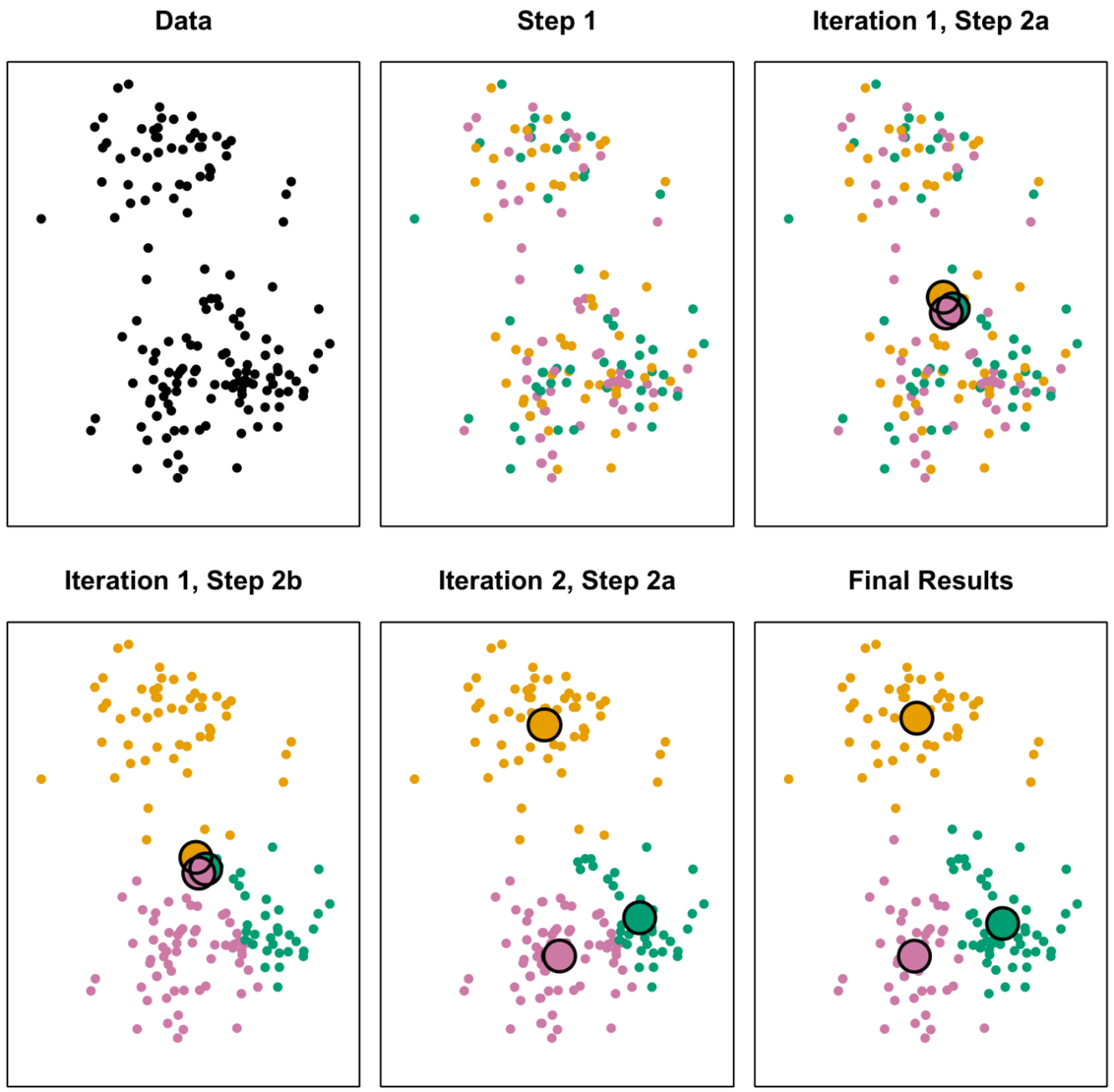
Clustering in iterative steps until convergence:

Step 0 randomly partition the data in k cluster

Step $i \to i+1$**:**

- find the centroid of each cluster
- classify each point along with the closest centroid.

Clustering method 1: k-means

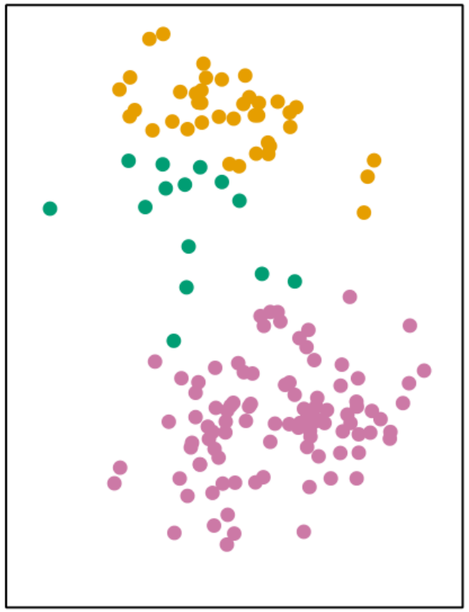


Clustering in iterative steps until convergence:

Step 0 randomly partition the data in k cluster

Step $i \to i+1$**:**

- find the centroid of each cluster
- classify each point along with the closest centroid.



Possibility to fail

Clustering method 2: PCA + k-means

Clustering method 2: PCA + k-means

Principal component analysis (PCA) generally used to reduce dimension.

Clustering method 2: PCA + k-means

Principal component analysis (PCA) generally used to reduce dimension.
$p \longrightarrow d$ with d small
Given $(x_1, \dots, x_n) \in \mathbb{R}^{p \times n}$,

Look for $u_1, \dots, u_d$ orthonormal, such that $(u^T x_i)_{i \in [n]}$ has the *biggest variance*.

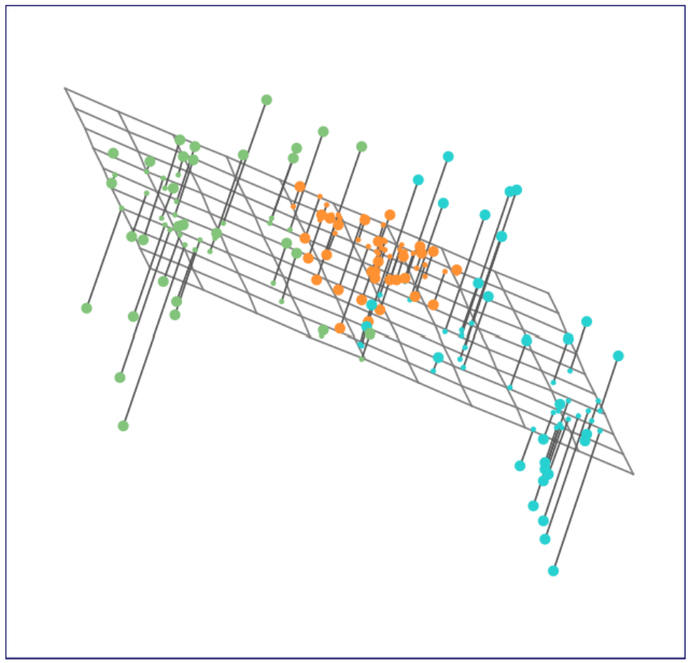
Maximize $u \in \mathbb{R}^p$:

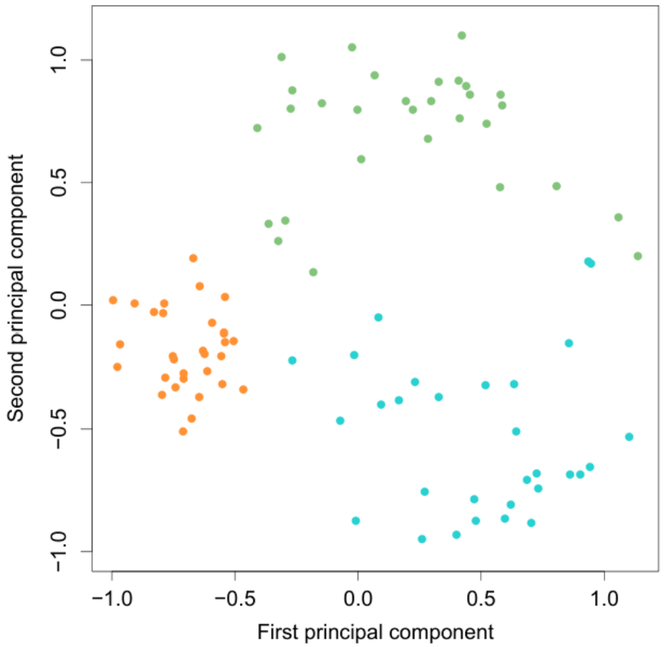
$$u^T \left(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T \right) u$$

Exactly the first d eigenvectors of $(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T)$.

Clustering method 2: PCA + k-means

(Left) 3d representation of the data, the plane is directed by the two first principal component. **(Right)** coordinates of the data projected on the two first principal components.





Principal component analysis (PCA) generally used to reduce dimension.

$p \longrightarrow d$ with d small

Given $(x_1, \dots, x_n) \in \mathbb{R}^{p \times n}$,

Look for $u_1, \dots, u_d$ orthonormal, such that $(u^T x_i)_{i \in [n]}$ has the *biggest variance*.

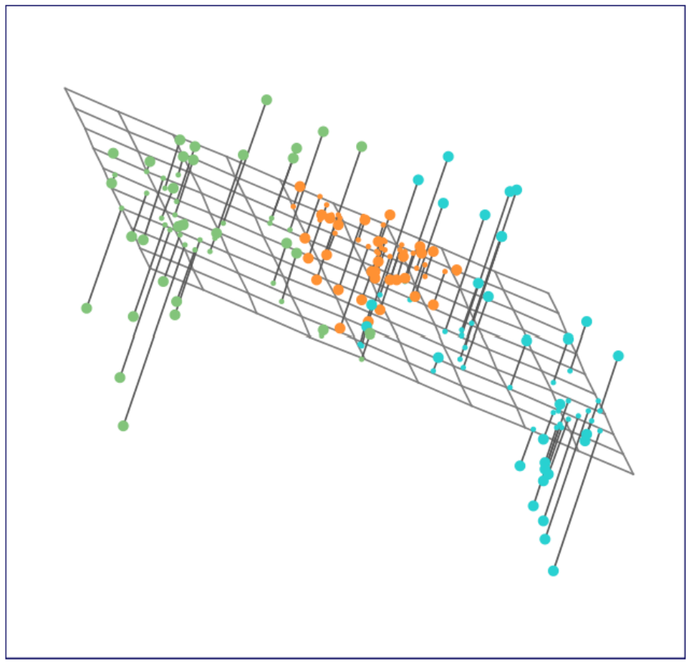
Maximize $u \in \mathbb{R}^p$:

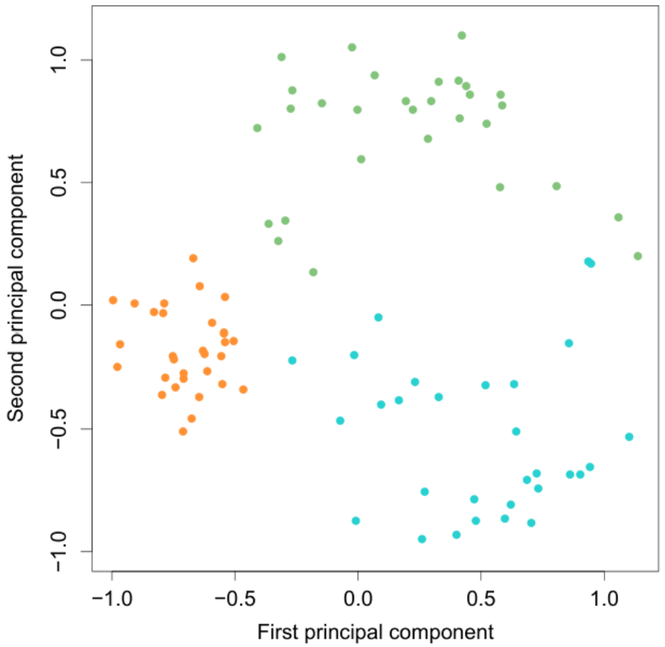
$$u^T \left(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T \right) u$$

Exactly the first d eigenvectors of $(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T)$.

Clustering method 2: PCA + k-means

(Left) 3d representation of the data, the plane is directed by the two first principal component. **(Right)** coordinates of the data projected on the two first principal components.





$\longrightarrow$ For classification, do k-means on $u_1, \ldots, u_d$.

Principal component analysis (PCA) generally used to reduce dimension.

$p \longrightarrow d$ with d small

Given $(x_1, \ldots, x_n) \in \mathbb{R}^{p \times n}$,

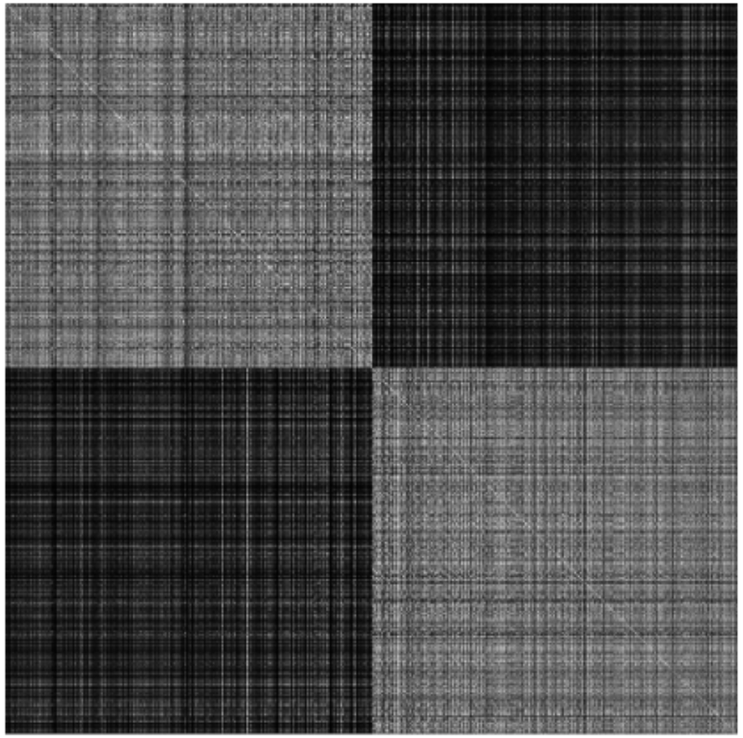
Look for $u_1, \ldots, u_d$ orthonormal, such that $(u^T x_i)_{i \in [n]}$ has the *biggest variance*.

Maximize $u \in \mathbb{R}^p$:

$$u^T \left(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T \right) u$$

Exactly the first d eigenvectors of $(\frac{1}{n} X X^T - \frac{1}{n^2} X 1_n^T 1_n X^T)$.

Clustering method 3: Spectral clustering

$$\mathbf{K} = \left[\right.$$  $$\left. \right] = (K(x_i, x_j))_{i,j\in[n]} \in \mathbb{R}^{n\times n}$$

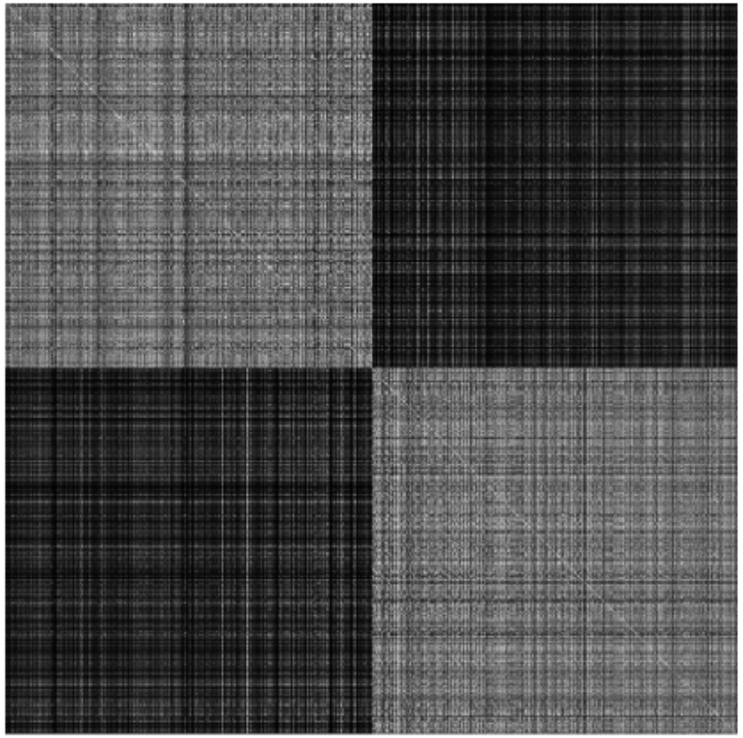
Two classes:

- $x_1, \dots, x_{n/2} \sim \mathcal{N}(\mu, I_p)$
- $x_{n/2+1}, \dots, x_n \sim \mathcal{N}(-\mu, I_p)$

with $\mu = (2, 0, \dots, 0) \in \mathbb{R}^p$, $n = 500$, $p = 5$.

with for ex. $K(x, y) = e^{-\frac{\|x-y\|^2}{2p}}$ ("Heat kernel")

Clustering method 3: Spectral clustering

$$\mathbf{K} = \left[\right.$$  $$\left. \right]$$

Two classes:

- $x_1, \dots, x_{n/2} \sim \mathcal{N}(\mu, I_p)$
- $x_{n/2+1}, \dots, x_n \sim \mathcal{N}(-\mu, I_p)$

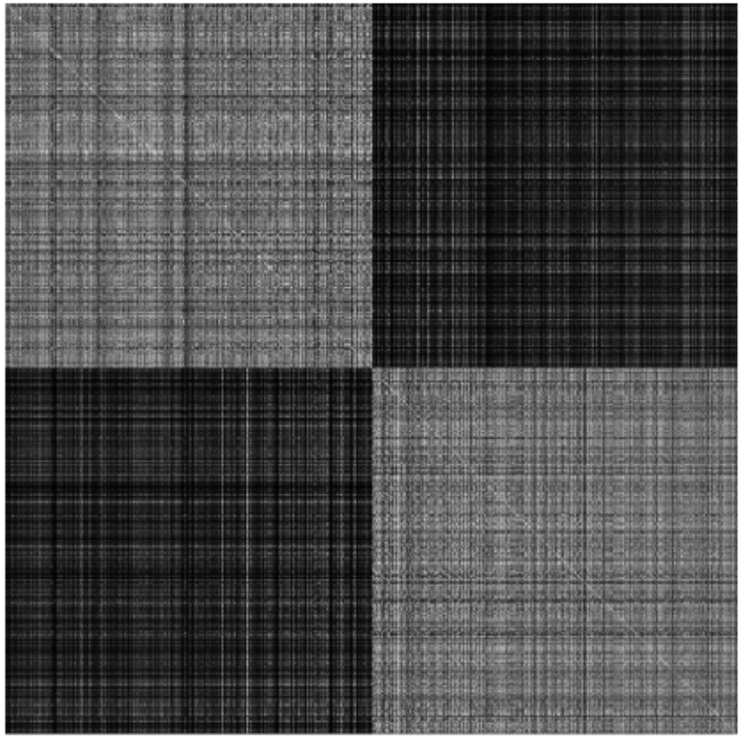
with $\mu = (2, 0, \dots, 0) \in \mathbb{R}^p$, $n = 500$, $p = 5$.

$$= (K(x_i, x_j))_{i,j \in [n]} \in \mathbb{R}^{n \times n}$$

with for ex. $K(x, y) = e^{-\frac{\|x-y\|^2}{2p}}$ ("Heat kernel")

$\Longrightarrow$ Look for first eigenvectors of K should capture class information

Clustering method 3: Spectral clustering

$$\mathbf{K} = \left[\right.$$  $$\left. \right] = (K(x_i, x_j))_{i,j\in[n]} \in \mathbb{R}^{n\times n}$$

$$\mathbf{v}_2 = \left[\right.$$  $$\left. \right]$$

Two classes:

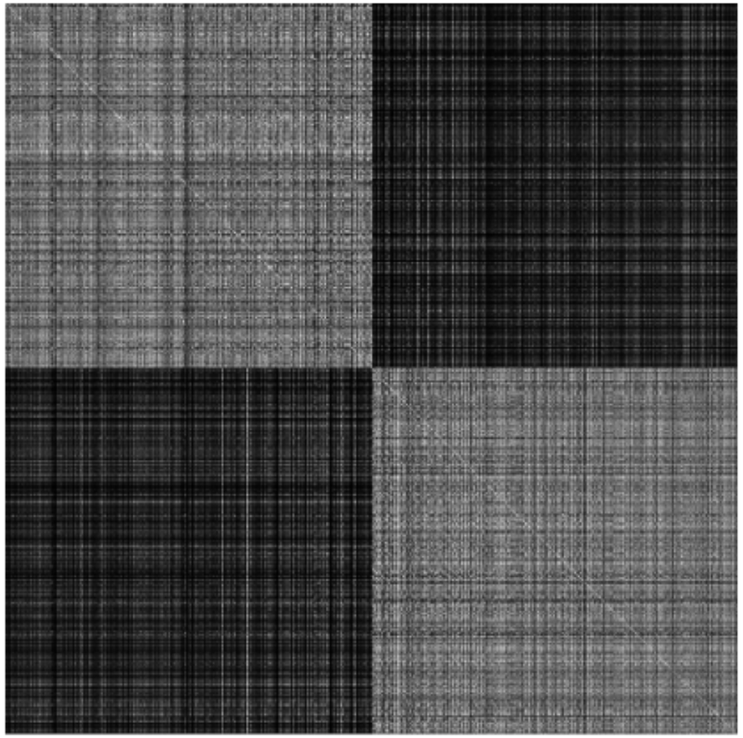
- $x_1, \dots, x_{n/2} \sim \mathcal{N}(\mu, I_p)$
- $x_{n/2+1}, \dots, x_n \sim \mathcal{N}(-\mu, I_p)$

with $\mu = (2, 0, \dots, 0) \in \mathbb{R}^p$, $n = 500$, $p = 5$.

with for ex. $K(x, y) = e^{-\frac{\|x-y\|^2}{2p}}$ ("Heat kernel")

$\Longrightarrow$ Look for first eigenvectors of K should capture class information

Clustering method 3: Spectral clustering

$$\mathbf{K} = \left[\right.$$  $$\left. \right] = (K(x_i, x_j))_{i,j\in[n]} \in \mathbb{R}^{n\times n}$$

$$\mathbf{v}_2 = \left[\right.$$  $$\left. \right]$$

Two classes:

- $x_1, \dots, x_{n/2} \sim \mathcal{N}(\mu, I_p)$
- $x_{n/2+1}, \dots, x_n \sim \mathcal{N}(-\mu, I_p)$

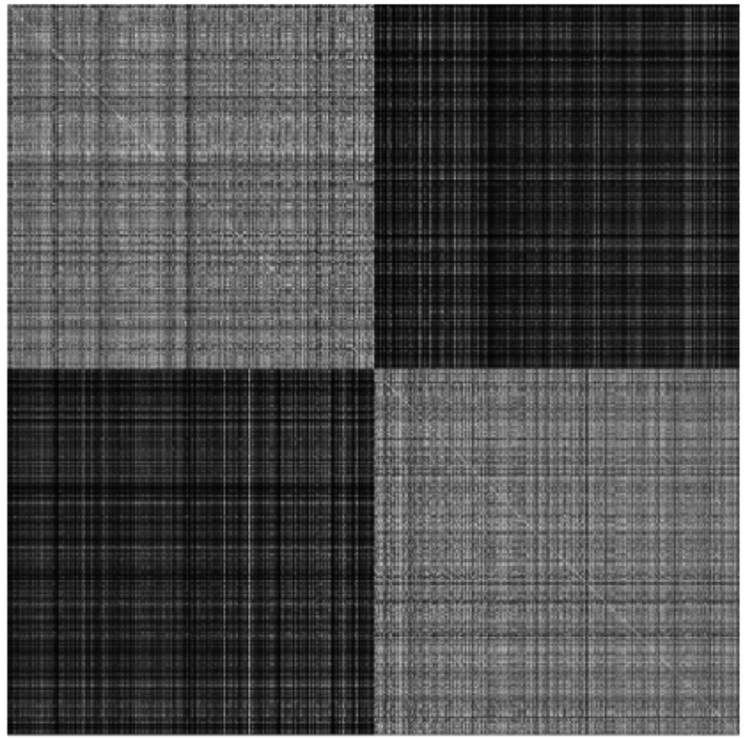
with $\mu = (2, 0, \dots, 0) \in \mathbb{R}^p$, $n = 500$, $p = 5$.

with for ex. $K(x, y) = e^{-\frac{\|x-y\|^2}{2p}}$ ("Heat kernel")

$\Longrightarrow$ Look for first eigenvectors of K should capture class information

Clustering method 3: Spectral clustering

$$\mathbf{K} = \left[\quad \right] = (K(x_i, x_j))_{i,j \in [n]} \in \mathbb{R}^{n \times n}$$



$$\mathbf{v}_2 = \left[\quad \right]$$

Two classes:

- $x_1, \dots, x_{n/2} \sim \mathcal{N}(\mu, I_p)$
- $x_{n/2+1}, \dots, x_n \sim \mathcal{N}(-\mu, I_p)$

with $\mu = (2, 0, \dots, 0) \in \mathbb{R}^p$, $n = 500$, $p = 5$.

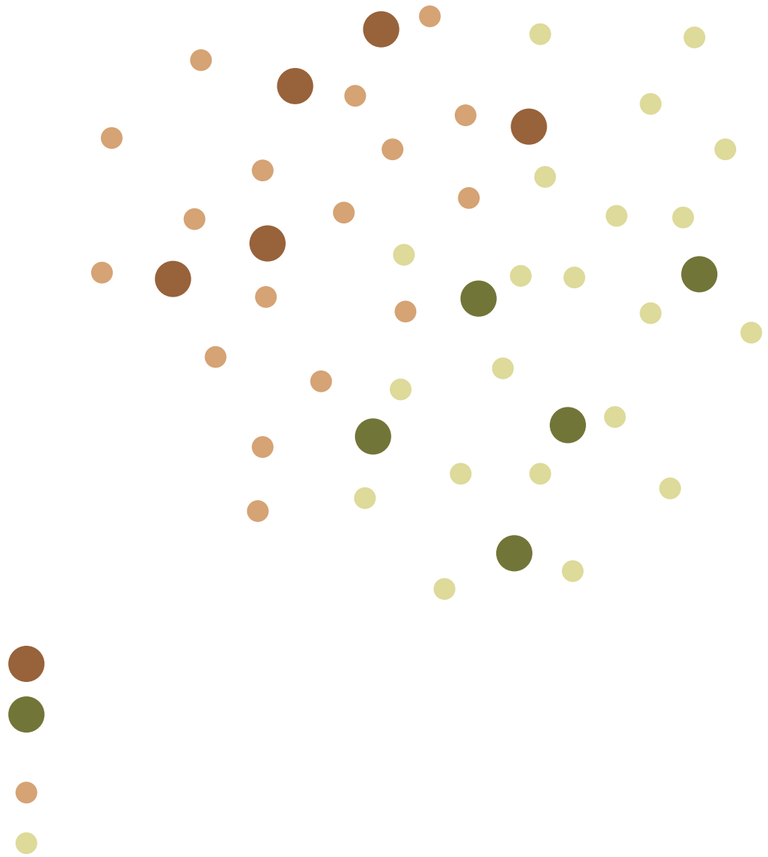
with for ex. $K(x, y) = e^{-\frac{\|x-y\|^2}{2p}}$ ("Heat kernel")

$\Longrightarrow$ Look for first eigenvectors of K should capture class information

When doing PCA, work with $\frac{1}{n} X X^T \in \mathbb{R}^{p \times p}$.

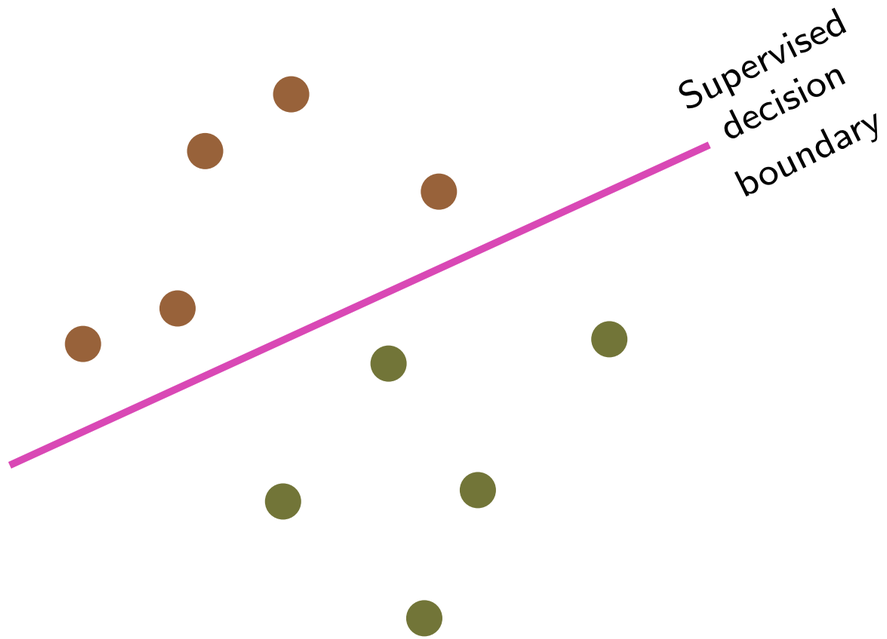
Semi-supervised learning with empirical risk minimization

Setting:



Semi-supervised learning with empirical risk minimization

Setting:

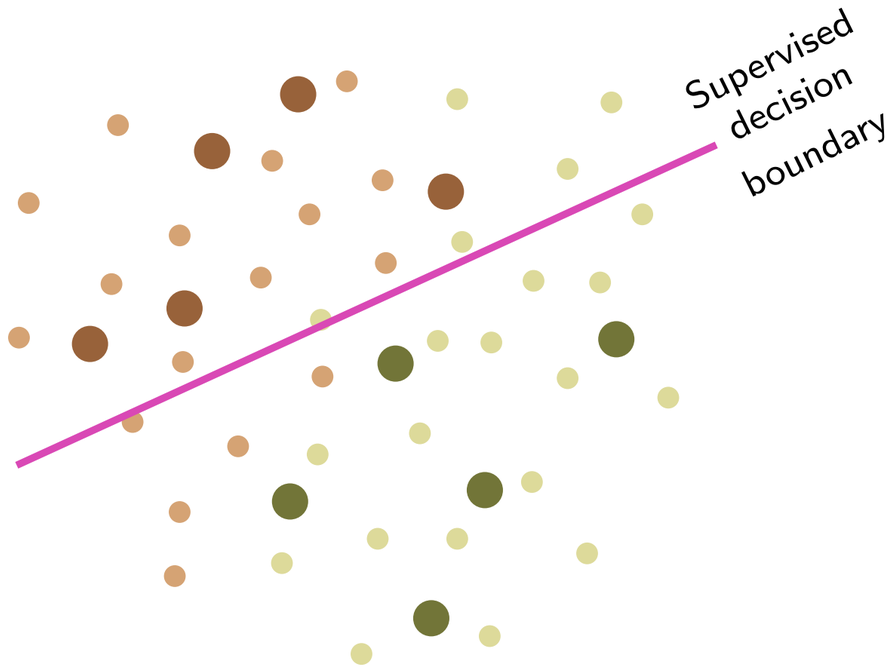


Labeled data

Unlabeled data

Semi-supervised learning with empirical risk minimization

Setting:

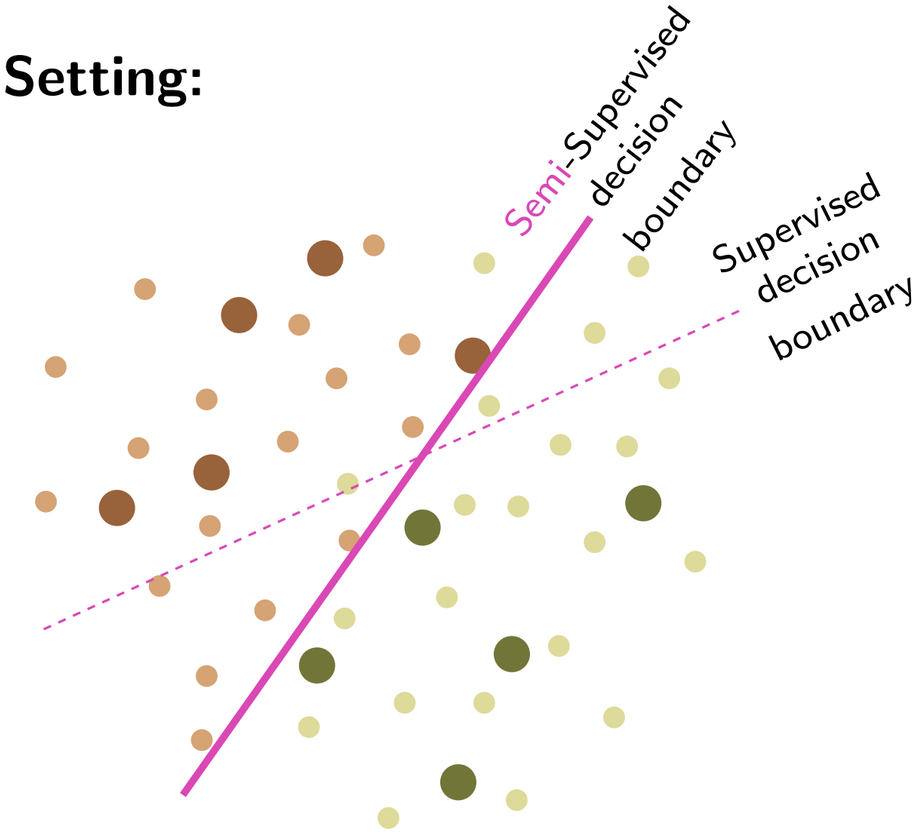


Labeled data

Unlabeled data

Semi-supervised learning with empirical risk minimization

Setting:



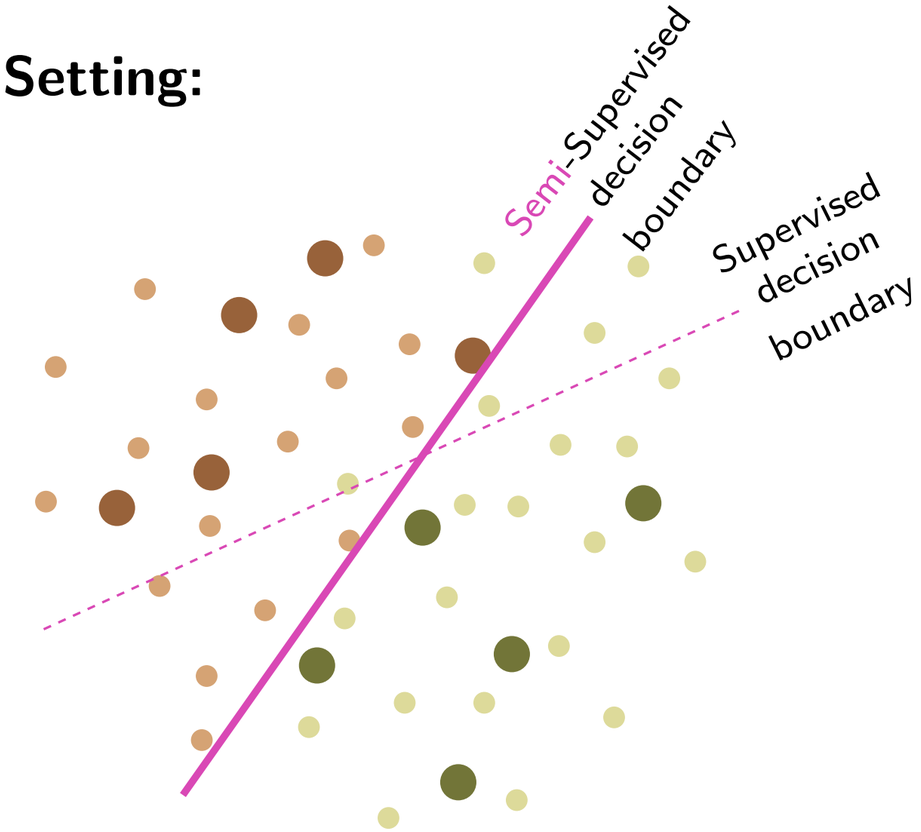
Solution

Labeled data

Unlabeled data

Semi-supervised learning with empirical risk minimization

Setting:



Labeled data

Unlabeled data

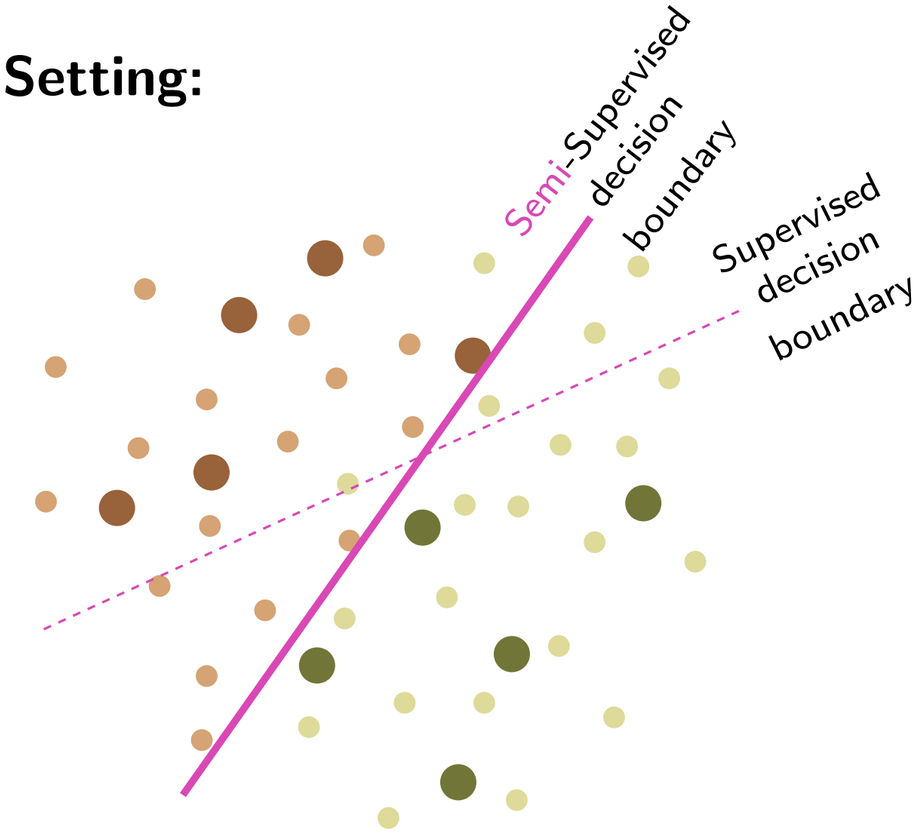
Main idea:

Take advantage of **both** the labeled and the unlabeled data to make the classification

Solution

Semi-supervised learning with empirical risk minimization

Setting:



Labeled data

Unlabeled data

Main idea:

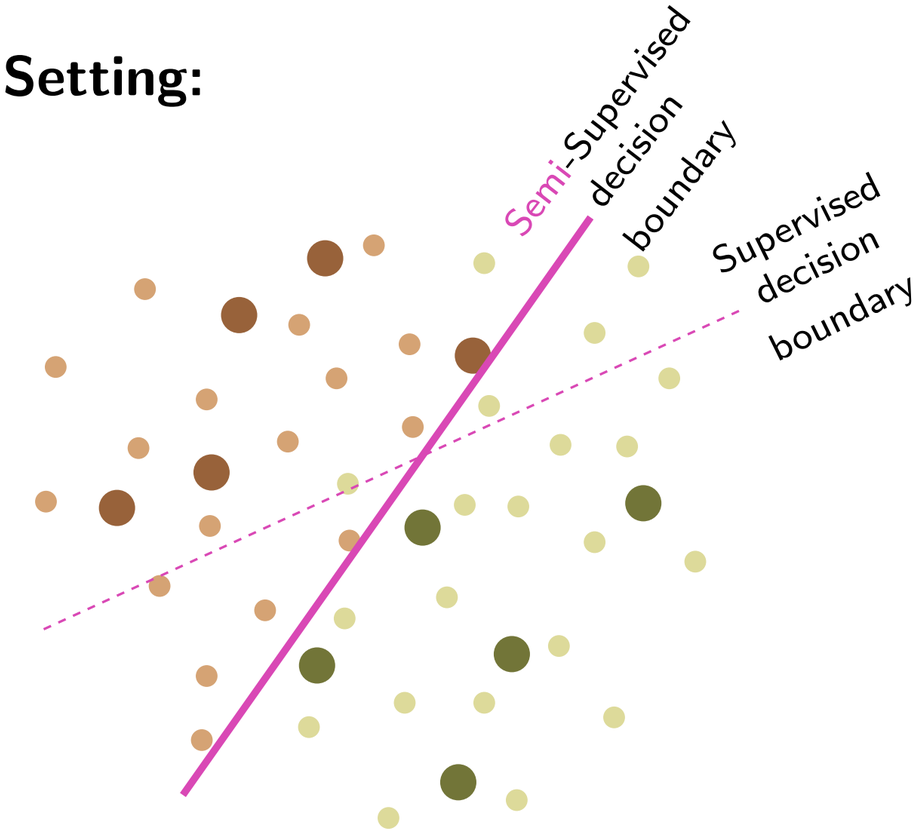
Take advantage of **both** the labeled and the unlabeled data to make the classification

Solution

Combines supervised and unsupervised methods

Semi-supervised learning with empirical risk minimization

Setting:



Labeled data: $(x_1, y_1), \ldots, (x_{n_l}, y_{n_l}) \in \mathbb{R}^p \times \{-1, 1\}$

Unlabeled data: $z_1, \ldots, z_{n_u} \in \mathbb{R}^p$

Main idea:

Take advantage of **both** the labeled and the unlabeled data to make the classification

Solution

Combines supervised and unsupervised methods

Best formulation given by the Empirical risk minimization (ERM):

Minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(w)$$

Supervised Loss

Unsupervised Loss

Regularisation

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

- Look for a linear boundary: $h_w(x) = w^T x$

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

- Look for a linear boundary: $h_w(x) = w^T x$
- Choice for the Supervised loss:

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l}\sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u}\sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

- Look for a linear boundary: $h_w(x) = w^T x$
- Choice for the Supervised loss:

  With $y_i \cdot w^T x_i \to t$:

  Logistic loss: $L_l(t) = \log(1 + e^{-t})$ (LR)

  Hinge loss: $L_l(t) = \max(1 - t, 0)$ (SVM)

  Exponential loss: $L_l(t) = e^{-t}$ (Adaboost)

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

- Look for a linear boundary: $h_w(x) = w^T x$
- Choice for the Supervised loss:

  With $y_i \cdot w^T x_i \to t$:

  Logistic loss: $L_l(t) = \log(1 + e^{-t})$ (LR)

  Hinge loss: $L_l(t) = \max(1 - t, 0)$ (SVM)

  Exponential loss: $L_l(t) = e^{-t}$ (Adaboost)

- Choice for the Unsupervised loss:

  Pb: Unsupervised clustering loss still not provided

  Get inspiration from PCA:

  Minimize $\frac{1}{n_u} \sum_{i=1}^{n} w^T z_i z_i^T w$

Semi-supervised learning with empirical risk minimization

Classification example: minimize on $w \in \mathbb{R}^q$:

$$\frac{1}{n_l}\sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u}\sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

- Look for a linear boundary: $h_w(x) = w^T x$
- Choice for the Supervised loss:

  With $y_i \cdot w^T x_i \to t$:

  Logistic loss: $L_l(t) = \log(1 + e^{-t})$ (LR)

  Hinge loss: $L_l(t) = \max(1 - t, 0)$ (SVM)

  Exponential loss: $L_l(t) = e^{-t}$ (Adaboost)

- Need to work with centered data:

  Set $m = \frac{1}{n_l}\sum_{i=1}^{n_l} x_i + \frac{1}{n_u}\sum_{i=1}^{n_u} z_i$

  $x_i \leftarrow x_i - m$

  $z_i \leftarrow z_i - m$

- Choice for the Unsupervised loss:

  Pb: Unsupervised clustering loss still not provided

  Get inspiration from PCA:

  Minimize $\frac{1}{n_u}\sum_{i=1}^{n} w^T z_i z_i^T w$

  w: first principal component

  $L_u(t) = t^2$ but also possible $L_u(t) = |t|$,
  Entropy loss: $L_u(t) = t\log(t)$ when $Y \in \{0, 1\}$

Semi-supervised learning with empirical risk minimization

Minimize on $w \in \mathbb{R}^q$:

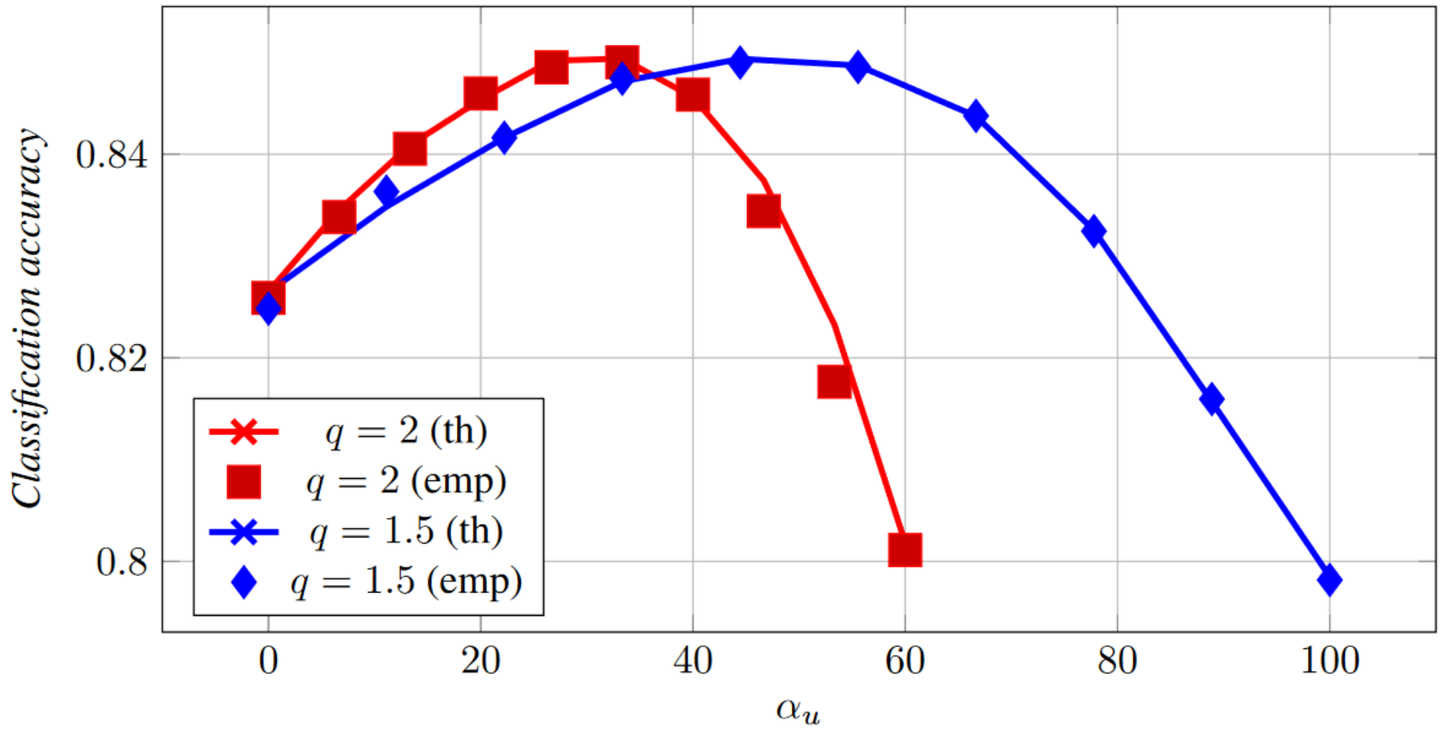
$$\frac{1}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{1}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

Semi-supervised learning with empirical risk minimization

Minimize on $w \in \mathbb{R}^q$:

$$\frac{\alpha_l}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{\alpha_u}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

$\Longrightarrow$ Trade-off between α_l and α_u to weight the contribution of labeled and unlabeled data.

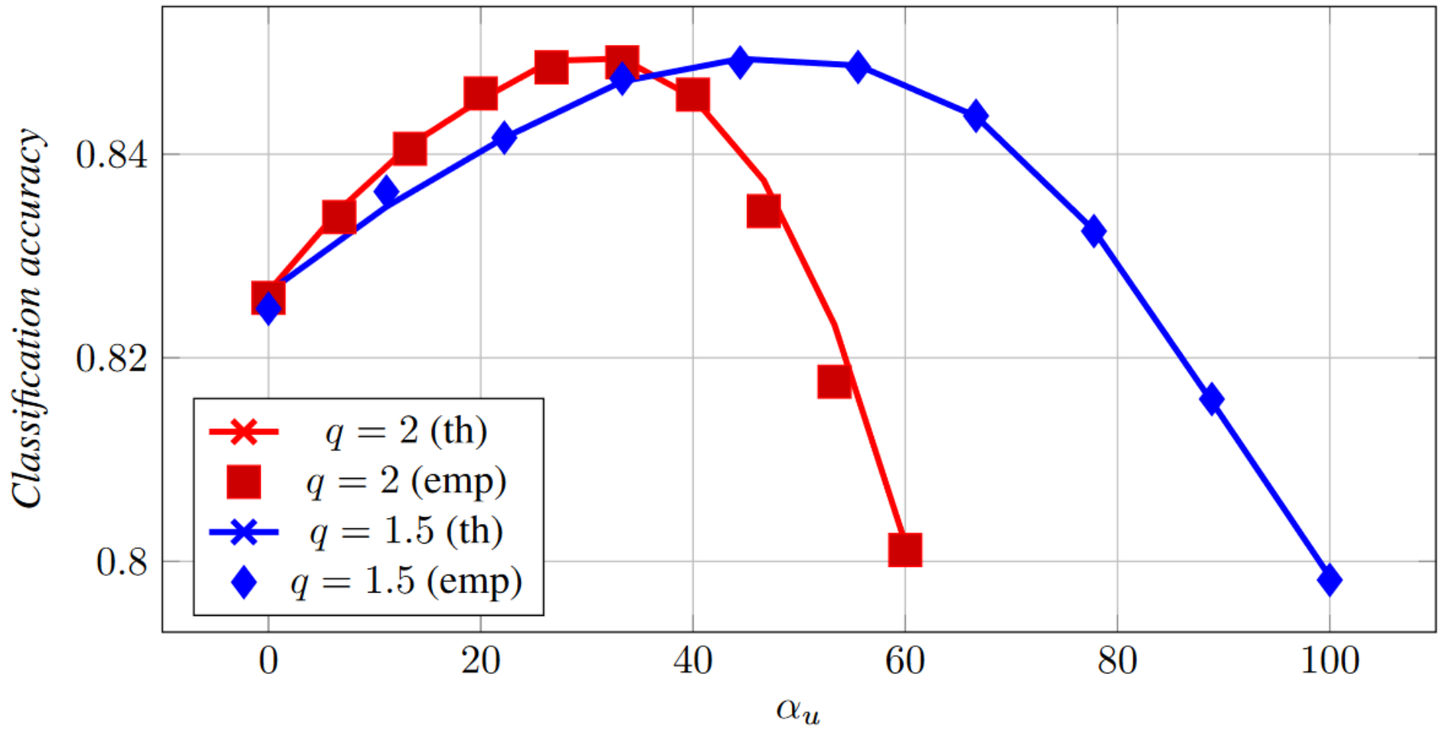


Semi-supervised learning with empirical risk minimization

Minimize on $w \in \mathbb{R}^q$:

$$\frac{\alpha_l}{n_l} \sum_{i=1}^{n_l} L_l(h_w(x_i), y_i) + \frac{\alpha_u}{n_u} \sum_{i=1}^{n_u} L_u(h_w(z_i)) + r(h_w)$$

$\Longrightarrow$ Trade-off between α_l and α_u to weight the contribution of labeled and unlabeled data.



Accuracy for two losses:
$L_l = L_u : t \mapsto t^q$ with $q = 2, 1.5$
Low density of unlabeled data:
$\frac{n_u}{n_l} = 1\%$

No big influence of the loss, curve different but maximum equal.