Unsupervised Learning

COMP9417 Machine Learning and Data Mining

May 2, 2017

Acknowledgements

Material derived from slides for the book
"Elements of Statistical Learning (2nd Ed.)" by T. Hastie,
R. Tibshirani & J. Friedman. Springer (2009)
http://statweb.stanford.edu/~tibs/ElemStatLearn/

Material derived from slides for the book
"Machine Learning: A Probabilistic Perspective" by P. Murphy
MIT Press (2012)
http://www.cs.ubc.ca/~murphyk/MLbook

Material derived from slides for the book
"Machine Learning" by P. Flach
Cambridge University Press (2012)
http://cs.bris.ac.uk/~flach/mlbook

Material derived from slides for the book
"Bayesian Reasoning and Machine Learning" by D. Barber
Cambridge University Press (2012)
http://www.cs.ucl.ac.uk/staff/d.barber/brml

Material derived from figures for the book
"Python Data Science Handbook" by J. VanderPlas
O'Reilly Media (2017)
http://shop.oreilly.com/product/0636920034919.do

Material derived from slides for the course
"Machine Learning" by A. Srinivasan

BITS Pilani, Goa, India (2016)

Contents

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning
- outline a number of frameworks for unsupervised learning

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning
- outline a number of frameworks for unsupervised learning
- describe k-means clustering

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning
- outline a number of frameworks for unsupervised learning
- describe k-means clustering
- describe the role of the EM algorithm in k-means clustering

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning
- outline a number of frameworks for unsupervised learning
- describe k-means clustering
- describe the role of the EM algorithm in k-means clustering
- describe hierarchical clustering

Aims

This lecture will develop your understanding of unsupervised learning methods. Following it you should be able to:

- describe the problem of dimensionality reduction
- outline the method of Principal Component Analysis
- describe the problem of unsupervised learning
- outline a number of frameworks for unsupervised learning
- describe k-means clustering
- describe the role of the EM algorithm in k-means clustering
- describe hierarchical clustering
- outline methods of evaluation for unsupervised learning

Contents

Supervised vs. Unsupervised Learning

Supervised vs. Unsupervised Learning

Supervised learning — classes are *predefined* and need a “definition”, in terms of the data. Methods are known as: classification, discriminant analysis, class prediction, supervised pattern recognition.

Supervised vs. Unsupervised Learning

Supervised learning — classes are *predefined* and need a "definition", in terms of the data. Methods are known as: classification, discriminant analysis, class prediction, supervised pattern recognition.

Unsupervised learning — classes are initially *unknown* and their definitions need to be "discovered" from the data. Methods are known as: cluster analysis, class discovery, unsupervised pattern recognition.

Supervised vs. Unsupervised Learning

Supervised learning — classes are *predefined* and need a "definition", in terms of the data. Methods are known as: classification, discriminant analysis, class prediction, supervised pattern recognition.

Unsupervised learning — classes are initially *unknown* and their definitions need to be "discovered" from the data. Methods are known as: cluster analysis, class discovery, unsupervised pattern recognition.

So: *unsupervised learning* methods address the problem of assigning instances to classes *given only observations about the instances*, i.e., without being given class "labels" for instances by a "teacher".

Unsupervised learning

Why do we need unsupervised learning ?

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)
  - time-consuming (labelling requires thinking)

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)
  - time-consuming (labelling requires thinking)
  - expensive (see above)

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)
  - time-consuming (labelling requires thinking)
  - expensive (see above)
  - error-prone (mistakes, ambiguity)

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)
  - time-consuming (labelling requires thinking)
  - expensive (see above)
  - error-prone (mistakes, ambiguity)
- in principle, can use any feature as the "label"

Unsupervised learning

Why do we need unsupervised learning ?

- most of the world's data is *unlabelled*
- getting a human to label data is often
  - difficult (what are the classes ?)
  - time-consuming (labelling requires thinking)
  - expensive (see above)
  - error-prone (mistakes, ambiguity)
- in principle, can use any feature as the "label"
- unfortunately, often the class is not a known feature

Unsupervised learning

What is unsupervised learning good for ?

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem
- to group data instances into subsets

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem
- to group data instances into subsets
- to discover structure like hierarchies of subconcepts

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem
- to group data instances into subsets
- to discover structure like hierarchies of subconcepts
- to learn new "features" for later use in classification

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem
- to group data instances into subsets
- to discover structure like hierarchies of subconcepts
- to learn new “features” for later use in classification
- to track “concept drift” over time

Unsupervised learning

What is unsupervised learning good for ?

- exploratory data analysis, e.g., with visualization
- data transformation to simplify a classification problem
- to group data instances into subsets
- to discover structure like hierarchies of subconcepts
- to learn new "features" for later use in classification
- to track "concept drift" over time
- to learn generative models for images, text, video, speech, etc.

Contents

Dimensionality Reduction

What is this and why would we need it ?

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
    - including them is all unlikely to improve models

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features
  - should be fewer than the original set

Dimensionality Reduction

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
    - including them is all unlikely to improve models
    - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features
    - should be fewer than the original set
    - should preserve information in original set (as far as possible)

Principal Component Analysis (PCA)

What is this and why would we need it ?

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
    - including them is all unlikely to improve models
    - feature selection may return arbitrary features

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features
  - should be fewer than the original set

Principal Component Analysis (PCA)

What is this and why would we need it ?

- each numeric feature in a dataset is a dimension
- in general, no restrictions on the number of dimensions
- however, many features could be related
- do we need them all in our dataset ?
  - including them is all unlikely to improve models
  - feature selection may return arbitrary features
- so, what to do ?
- one solution would be to find set of *new* features
  - should be fewer than the original set
  - should preserve information in original set (as far as possible)

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

Key idea: look for features in a transformed space so that each dimension in the new space captures the most variation in the original data when it is projected onto that dimension.

Principal Component Analysis (PCA)

Key idea: look for features in a transformed space so that each dimension in the new space captures the most variation in the original data when it is projected onto that dimension.
Any new features should be high;y correlated with (some of) the original features, but not with any of the other new features.

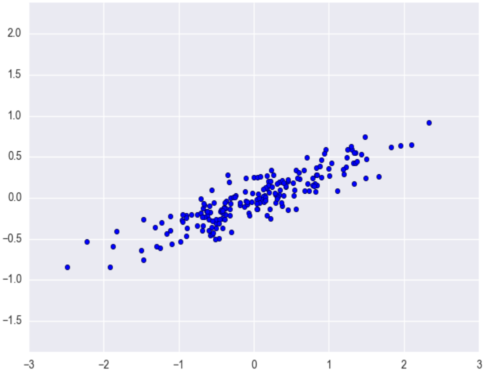
Principal Component Analysis (PCA)

Key idea: look for features in a transformed space so that each dimension in the new space captures the most variation in the original data when it is projected onto that dimension.

Any new features should be high;y correlated with (some of) the original features, but not with any of the other new features.

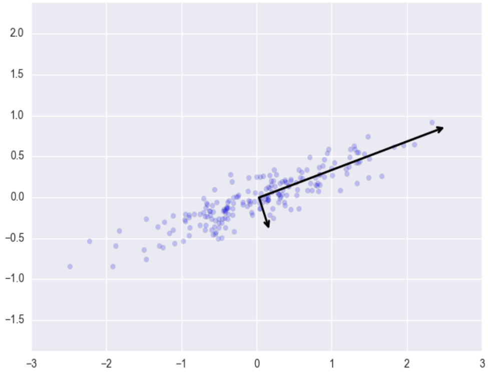
This suggests an approach: look into using the variance-covariance matrix we recall from correlation and regression

PCA Example



PCA looks for linear combinations of the original features. This dataset of 200 points seems to show such a relationship between two feature dimensions.

PCA Example



PCA finds two new features on which the original data can be projected, rotated and scaled. These explain respectively 0.75 and 0.02 of the variance.

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

1. take the data as an $n \times m$ matrix $\mathbf{X}$

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

1. take the data as an $n \times m$ matrix $\mathbf{X}$
2. "centre" the data by subtracting the mean of each column

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

1. take the data as an $n \times m$ matrix $\mathbf{X}$
2. "centre" the data by subtracting the mean of each column
3. construct covariance matrix $\mathbf{C}$ from centred matrix

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

1. take the data as an $n \times m$ matrix $\mathbf{X}$
2. "centre" the data by subtracting the mean of each column
3. construct covariance matrix $\mathbf{C}$ from centred matrix
4. compute eigenvector matrix $\mathbf{V}$ (rotation) and eigenvalue matrix $\mathbf{S}$ (scaling) such that $\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{S}$, and $\mathbf{S}$ is a diagonal $m \times m$ matrix

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

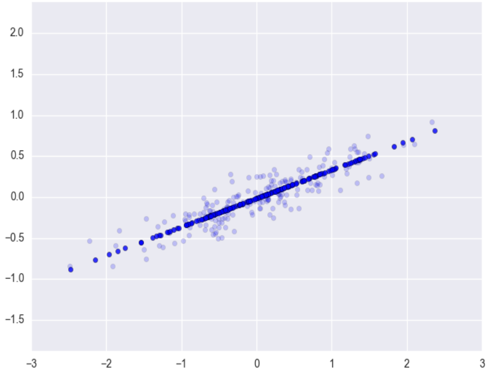
1. take the data as an $n \times m$ matrix $\mathbf{X}$
2. "centre" the data by subtracting the mean of each column
3. construct covariance matrix $\mathbf{C}$ from centred matrix
4. compute eigenvector matrix $\mathbf{V}$ (rotation) and eigenvalue matrix $\mathbf{S}$ (scaling) such that $\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{S}$, and $\mathbf{S}$ is a diagonal $m \times m$ matrix
5. sort columns of $\mathbf{S}$ in decreasing order (decreasing variance)

PCA Algorithm

This algorithm can be presented in several ways. Here are the basic steps in terms of the variance reduction idea:

1. take the data as an $n \times m$ matrix $\mathbf{X}$
2. "centre" the data by subtracting the mean of each column
3. construct covariance matrix $\mathbf{C}$ from centred matrix
4. compute eigenvector matrix $\mathbf{V}$ (rotation) and eigenvalue matrix $\mathbf{S}$ (scaling) such that $\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{S}$, and $\mathbf{S}$ is a diagonal $m \times m$ matrix
5. sort columns of $\mathbf{S}$ in decreasing order (decreasing variance)
6. remove columns of $\mathbf{S}$ below some minimum threshold

PCA Example



By rejecting the second component we reduce the dimensionality by 50% while preserving much of the original variance, seen by plotting the inverse transform of this component along with the original data.

PCA and friends

PCA and friends

- PCA complexity is cubic in number of original features

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets
- alternatively, approximate the sort of projection found by PCA

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets
- alternatively, approximate the sort of projection found by PCA
- for example, can use Random Projections

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets
- alternatively, approximate the sort of projection found by PCA
- for example, can use Random Projections
- more scalable, but what about quality of components ?

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets
- alternatively, approximate the sort of projection found by PCA
- for example, can use Random Projections
- more scalable, but what about quality of components ?
- can be shown to preserve distance relations from the original data

PCA and friends

- PCA complexity is cubic in number of original features
- this is not feasible for high-dimensional datasets
- alternatively, approximate the sort of projection found by PCA
- for example, can use Random Projections
- more scalable, but what about quality of components ?
- can be shown to preserve distance relations from the original data
- many other methods use essentially the same matrix decomposition idea, such as finding “topics” in text using Latent Semantic Analysis (next slide), finding hidden “sub-groups” for recommender systems, and so on

Finding Topics in Text

$$\underset{t \times d}{\left[\ \tilde{\mathbf{X}}\ \right]} = \underset{t \times k}{\left[\ \mathbf{U}_k\ \right]} \underset{k \times k}{\begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_k \end{bmatrix}} \underset{k \times d}{\left[\ \mathbf{V}_k^{\mathrm{T}}\ \right]}$$

Latent Semantic Analysis (LSA) factorizes a word count matrix for t terms from d documents using the Singular Value Decomposition (SVD) to find a number $k < d$ of topics. Here the diagonal matrix $\mathbf{S}$ has the topic "strength" sorted in decreasing order.

Dimensionality Reduction Summary

Dimensionality Reduction Summary

- PCA will transform original features to new space

Dimensionality Reduction Summary

- PCA will transform original features to new space
- every new feature is a linear combination of original features

Dimensionality Reduction Summary

- PCA will transform original features to new space
- every new feature is a linear combination of original features
- aim for new dimensions to maximise variance

Dimensionality Reduction Summary

- PCA will transform original features to new space
- every new feature is a linear combination of original features
- aim for new dimensions to maximise variance
- order by decreasing variance and remove those below a threshold

Dimensionality Reduction Summary

- PCA will transform original features to new space
- every new feature is a linear combination of original features
- aim for new dimensions to maximise variance
- order by decreasing variance and remove those below a threshold
- this reduces dimensionality

Dimensionality Reduction Summary

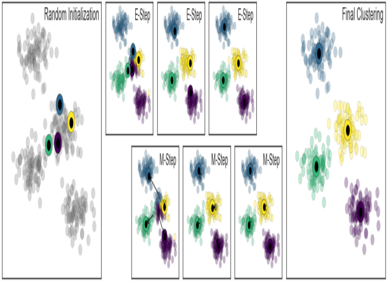
- PCA will transform original features to new space
- every new feature is a linear combination of original features
- aim for new dimensions to maximise variance
- order by decreasing variance and remove those below a threshold
- this reduces dimensionality
- algorithm applies matrix operations to translate, rotate and scale

Dimensionality Reduction Summary

- PCA will transform original features to new space
- every new feature is a linear combination of original features
- aim for new dimensions to maximise variance
- order by decreasing variance and remove those below a threshold
- this reduces dimensionality
- algorithm applies matrix operations to translate, rotate and scale
- many alternatives, e.g., Random Projections, Independent Component Analysis, etc.

Contents

k-Means Clustering Example



Data.

Contents

Summary: Unsupervised Learning

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample
  - train with large unlabelled sample then learn classes with small labelled sample

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample
  - train with large unlabelled sample then learn classes with small labelled sample
- reinforcement learning can be viewed as unsupervised

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
    - train with small labelled sample then improve with large unlabelled sample
    - train with large unlabelled sample then learn classes with small labelled sample
- reinforcement learning can be viewed as unsupervised
    - "reward" is a signal from the "environment"

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample
  - train with large unlabelled sample then learn classes with small labelled sample
- reinforcement learning can be viewed as unsupervised
  - "reward" is a signal from the "environment"
  - learning is designed to optimize function of reward

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample
  - train with large unlabelled sample then learn classes with small labelled sample
- reinforcement learning can be viewed as unsupervised
  - "reward" is a signal from the "environment"
  - learning is designed to optimize function of reward
- active learning

Other approaches

Unsupervised and supervised learning are at different ends of a continuum of "degrees of supervision". Between these extremes many other approaches are possible.

- semi-supervised learning, e.g.,
  - train with small labelled sample then improve with large unlabelled sample
  - train with large unlabelled sample then learn classes with small labelled sample
- reinforcement learning can be viewed as unsupervised
  - "reward" is a signal from the "environment"
  - learning is designed to optimize function of reward
- active learning
  - learning system acts to generate its own examples

Note: unsupervised learing an increasingly active research area, particularly in neural nets, e.g., Yann LeCun: "Unsupervised Learning: The Next Frontier in AI"
`https://www.youtube.com/watch?v=XTbLOjVF-y4`