Unsupervised Machine Learning

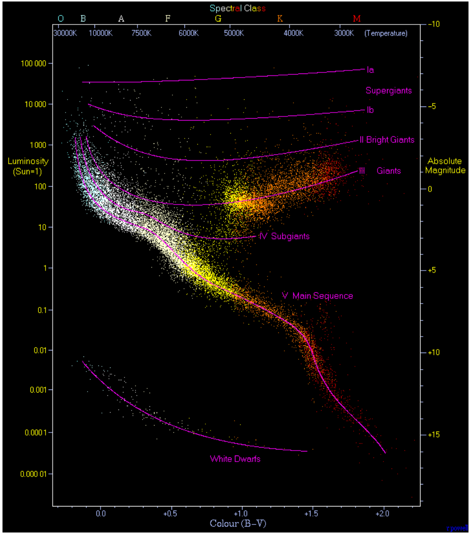
Martin Žonda and Pavel Baláž

Seminář Astronomického ústavu UK

Supervised Learning

Supervised

- Typical tasks: classification, target predictions, regression

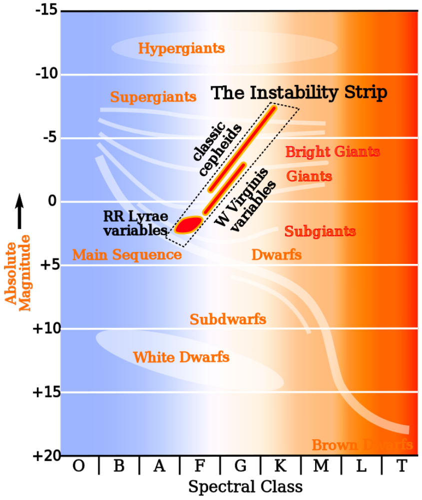


https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

Supervised Learning

Supervised

- Typical tasks: classification, target predictions, regression
- Training set contains labels, i.e., the desired result

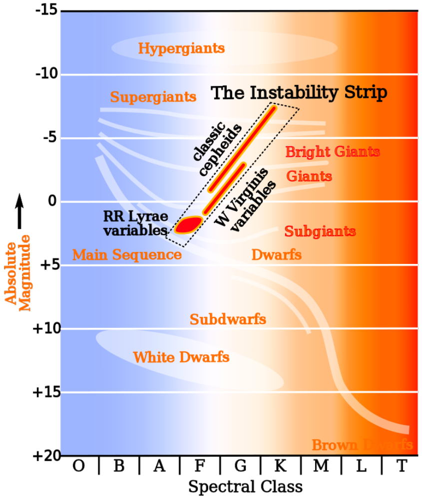


https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

Supervised Learning

Supervised

- Typical tasks: classification, target predictions, regression
- Training set contains labels, i.e., the desired result
- Some supervised learning techniques: **S**upport **V**ector **M**achines, Decision Trees and Random Forest, Supervised Neural Networks

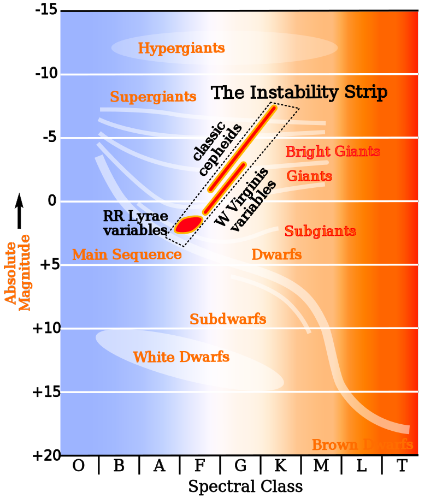


https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

Supervised Learning

Supervised

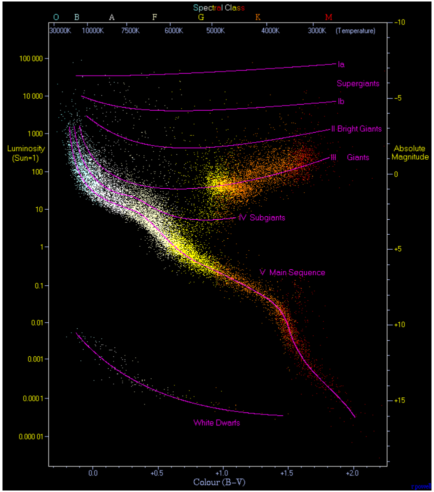
- Typical tasks: classification, target predictions, regression
- Training set contains labels, i.e., the desired result
- Some supervised learning techniques: **S**upport **V**ector **M**achines, Decision Trees and Random Forest, Supervised Neural Networks
- All of the above are used for **Stellar Classification**



https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

The main ideas behind unsupervised learning

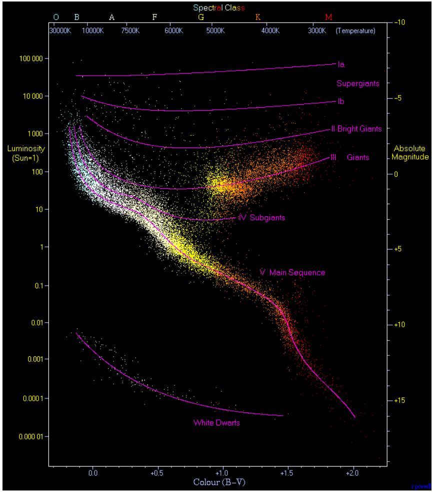
- Even **without labels** data have structure



https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

The main ideas behind unsupervised learning

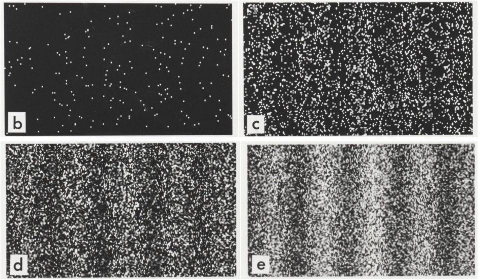
- Even **without labels** data have structure
- They represent an **underlying distribution**



https://en.wikipedia.org/wiki/Hertzsprung-Russell-diagram

The main ideas behind unsupervised learning

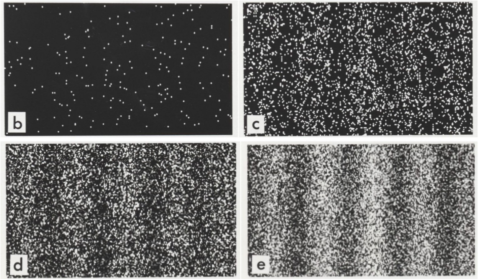
- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern



Dr. Tonomura

The main ideas behind unsupervised learning

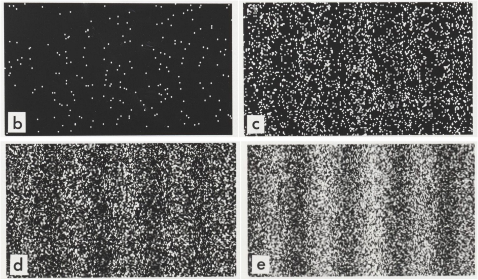
- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern
- We can construct **a model** of the data much less complex than the original set



Dr. Tonomura

The main ideas behind unsupervised learning

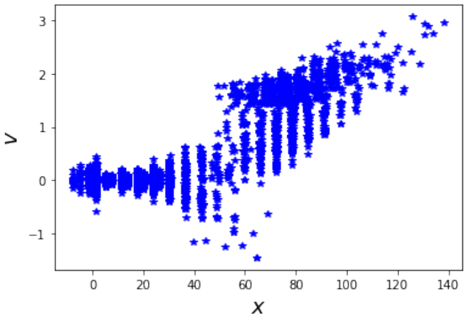
- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern
- We can construct **a model** of the data much less complex than the original set
- A perfect example in physics is thermodynamics



Dr. Tonomura

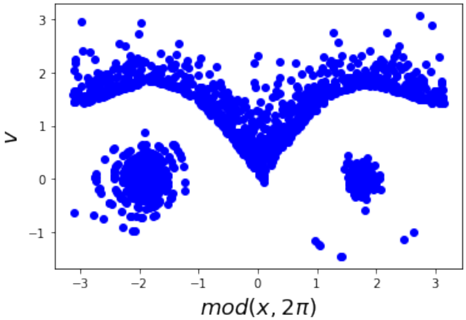
The main ideas behind unsupervised learning

- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern
- We can construct **a model** of the data much less complex than the original set
- A perfect example in physics is thermodynamics
- It helps a lot to find the right "angle" or "natural" basis of the data



The main ideas behind unsupervised learning

- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern
- We can construct **a model** of the data much less complex than the original set
- A perfect example in physics is thermodynamics
- It helps a lot to find the right "angle" or "natural" basis of the data

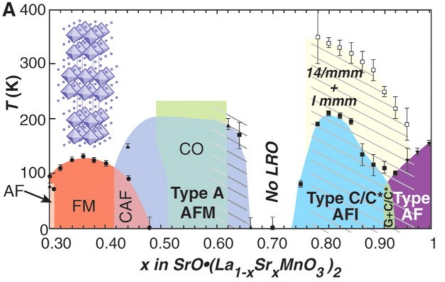


The main ideas behind unsupervised learning

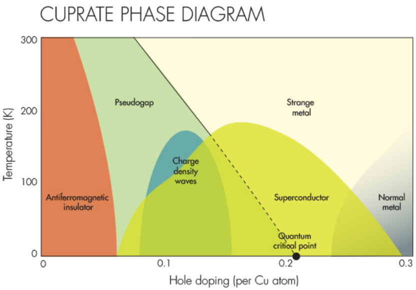
- Even **without labels** data have structure
- They represent an **underlying distribution**
- We often need a lot of data to notice the pattern
- We can construct **a model** of the data much less complex than the original set
- A perfect example in physics is thermodynamics
- It helps a lot to find the right "angle" or "natural" basis of the data

Unsupervised phase classification

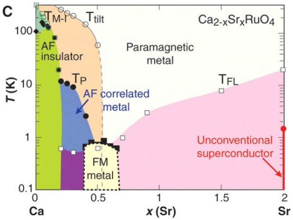
Bilayer Manganites [Dagotto, Science (2005)]



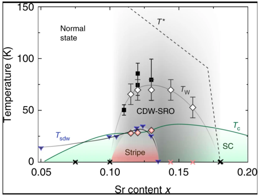
Cuprates [Shmahalo, Quanta Magazine, (2016)]



Single layered Ruthenates [Dagotto, Science (2005)]



LASCO [Wen at al.,Nature Comm. (2019)]



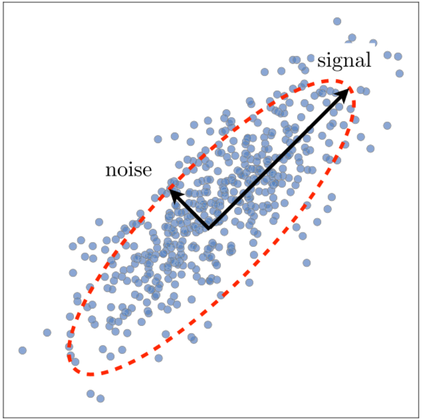
Outline

- **Principal Component Analysis (PCA)**
  - Dimensional reduction and visualization
  - Unsupervised phase classification
  - Kernel PCA
- **Clustering**
  - K-Means
  - Density-based (DB) clustering
- **Unsupervised phase classification**
  - Complicated phase diagrams
  - *Interpretability*

Dimensional reduction, data visualization and phase transitions

Why to reduce dimensionality

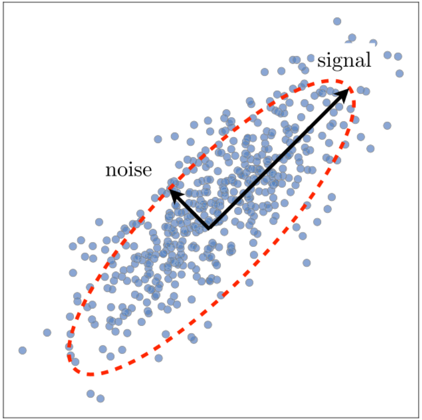
- By dimension we mean measured property



Physics Reports 810,1 (2019)

Why to reduce dimensionality

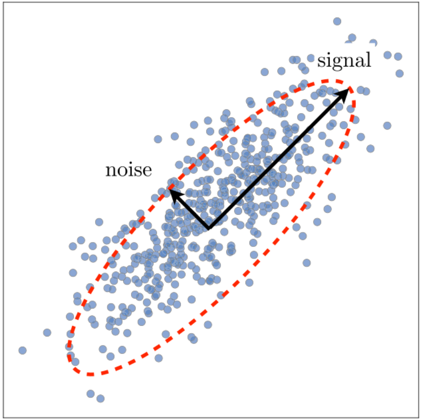
- By dimension we mean measured property
- There is a structure in (unlabeled) scientific data



Physics Reports 810,1 (2019)

Why to reduce dimensionality

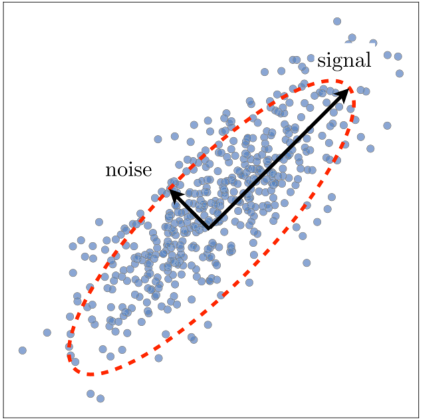
- By dimension we mean measured property
- There is a structure in (unlabeled) scientific data
- There are (probably) correlations between the measured properties



Physics Reports 810,1 (2019)

Why to reduce dimensionality

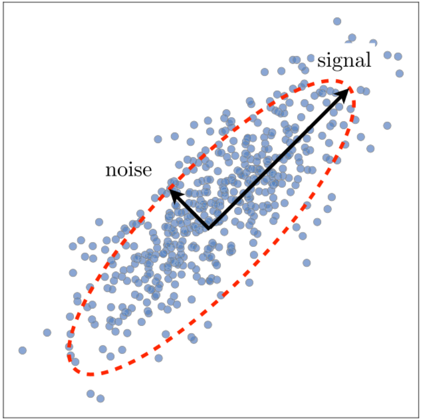
- By dimension we mean measured property
- There is a structure in (unlabeled) scientific data
- There are (probably) correlations between the measured properties
- Astronomical number of degrees of freedom can be often replaced by **order parameters** or **effective variables**



Physics Reports 810,1 (2019)

Why to reduce dimensionality

- By dimension we mean measured property
- There is a structure in (unlabeled) scientific data
- There are (probably) correlations between the measured properties
- Astronomical number of degrees of freedom can be often replaced by **order parameters** or **effective variables**
- **Intrinsic dimensionality** - a minimum number of dimensions required to capture the signal



Physics Reports 810,1 (2019)

Singular Value Decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T = \begin{bmatrix} \check{\boldsymbol{U}} & \check{\boldsymbol{U}}^{\perp} \end{bmatrix} \begin{bmatrix} \check{\Sigma} \\ \mathbf{0} \end{bmatrix} \boldsymbol{V}^T = \check{\boldsymbol{U}}\check{\Sigma}\boldsymbol{V}^T$$

Singular Value Decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T = \begin{bmatrix}\check{\boldsymbol{U}} & \check{\boldsymbol{U}}^{\perp}\end{bmatrix}\begin{bmatrix}\check{\Sigma} \\ \mathbf{O}\end{bmatrix}\boldsymbol{V}^T = \check{\boldsymbol{U}}\check{\Sigma}\boldsymbol{V}^T$$

```
U,S,V=np.linalg.svd(X,full_matrices=True)
U,S,V=np.linalg.svd(X,full_matrices=False)
```

Singular Value Decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \begin{bmatrix} \check{\boldsymbol{U}} & \check{\boldsymbol{U}}^\perp \end{bmatrix} \begin{bmatrix} \check{\boldsymbol{\Sigma}} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{V}^T = \check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}\boldsymbol{V}^T$$

```
U,S,V=np.linalg.svd(X,full_matrices=True)
U,S,V=np.linalg.svd(X,full_matrices=False)
```

Eckart-Young Theorem

The optimal rank-r approximation to $\boldsymbol{X}$, in a least-squares sense, is given by the rank-r SVD truncation $\tilde{\boldsymbol{X}}$.

Singular Value Decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T = \begin{bmatrix}\check{\boldsymbol{U}} & \check{\boldsymbol{U}}^{\perp}\end{bmatrix}\begin{bmatrix}\check{\Sigma} \\ \mathbf{0}\end{bmatrix}\boldsymbol{V}^T = \check{\boldsymbol{U}}\check{\Sigma}\boldsymbol{V}^T$$

```
U,S,V=np.linalg.svd(X,full_matrices=True)
U,S,V=np.linalg.svd(X,full_matrices=False)
```

Eckart-Young Theorem

The optimal rank-r approximation to $\boldsymbol{X}$, in a least-squares sense, is given by the rank-r SVD truncation $\tilde{\boldsymbol{X}}$.

$$\tilde{\boldsymbol{X}}_r = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \ldots$$

Singular Value Decomposition (SVD)

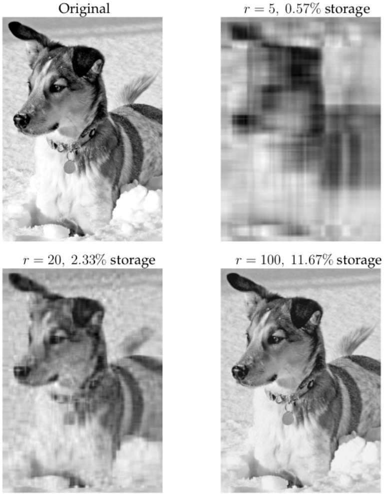
$$X = U\Sigma V^T = \begin{bmatrix} \check{U} & \check{U}^{\perp} \end{bmatrix} \begin{bmatrix} \check{\Sigma} \\ \mathbf{0} \end{bmatrix} V^T = \check{U}\check{\Sigma}V^T$$

```
U,S,V=np.linalg.svd(X,full_matrices=True)
U,S,V=np.linalg.svd(X,full_matrices=False)
```

Eckart-Young Theorem

The optimal rank-*r* approximation to $\boldsymbol{X}$, in a least-squares sense, is given by the rank-*r* SVD truncation $\tilde{\boldsymbol{X}}$.

$$\tilde{X}_r = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \ldots$$

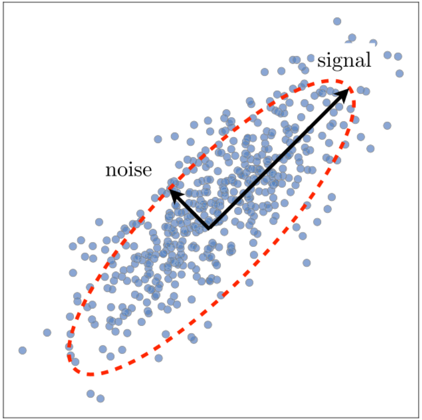


Original image resolution is 2000 × 1500

Brunton and Kutz, Data Driven Science & Engineering

Principal component analysis (PCA)

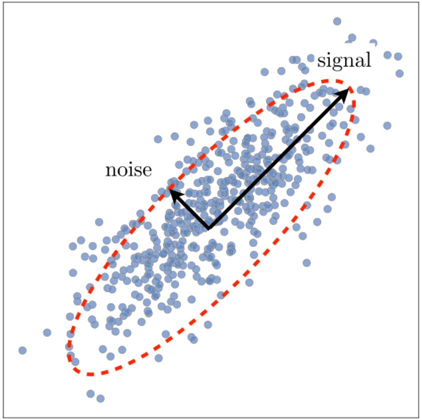
- **PCA** is the most important application of the SVD in ML
  (SVD is related to eigenvalue problem of the covariance matrix matrix $\frac{1}{n-1}\boldsymbol{X_c}\boldsymbol{X_c}^\dagger = \boldsymbol{V}\frac{\Sigma^2}{l-1}\boldsymbol{V}^\dagger$)



Physics Reports 810,1 (2019)

Principal component analysis (PCA)

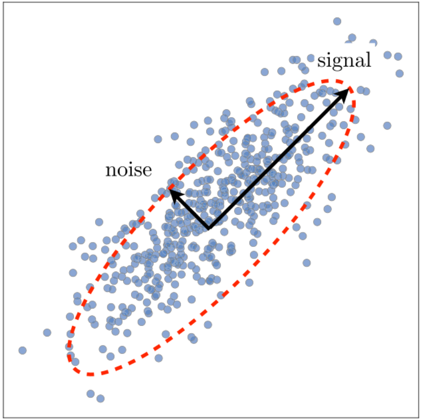
- **PCA** is the most important application of the SVD in ML
  (SVD is related to eigenvalue problem of the covariance matrix matrix $\frac{1}{n-1}\boldsymbol{X_c}\boldsymbol{X_c}^\dagger = \boldsymbol{V}\frac{\Sigma^2}{l-1}\boldsymbol{V}^\dagger$)
- The main goal of PCA is to identify the most meaningful basis!



Physics Reports 810,1 (2019)

Principal component analysis (PCA)

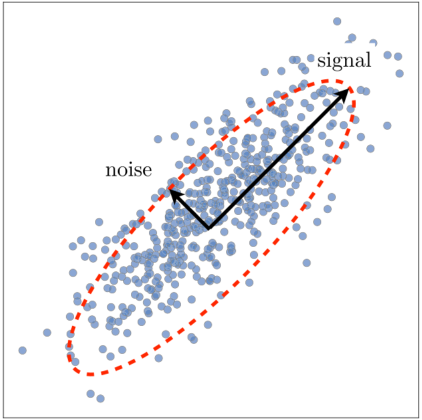
- **PCA** is the most important application of the SVD in ML
  (SVD is related to eigenvalue problem of the covariance matrix matrix $\frac{1}{n-1}\boldsymbol{X_c}\boldsymbol{X_c}^\dagger = \boldsymbol{V}\frac{\Sigma^2}{l-1}\boldsymbol{V}^\dagger$)
- The main goal of PCA is to identify the most meaningful basis!
- **What does the "most meaningful" even mean?**



Physics Reports 810,1 (2019)

Principal component analysis (PCA)

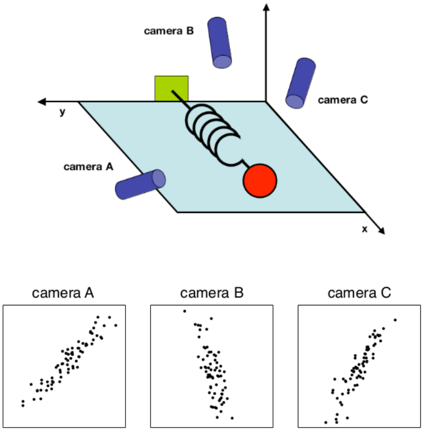
- **PCA** is the most important application of the SVD in ML
  (SVD is related to eigenvalue problem of the covariance matrix matrix $\frac{1}{n-1}\mathbf{X_c}\mathbf{X_c}^{\dagger} = \mathbf{V}\frac{\Sigma^2}{l-1}\mathbf{V}^{\dagger}$)
- The main goal of PCA is to identify the most meaningful basis!
- **What does the "most meaningful" even mean?**
- We assume that large variances means signal, i.e., that there is a large Signal to Noise ratio in our data!



Physics Reports 810,1 (2019)

Principal component analysis (PCA)

- **PCA** is the most important application of the SVD in ML
  (SVD is related to eigenvalue problem of the covariance matrix matrix $\frac{1}{n-1}\boldsymbol{X_c}\boldsymbol{X_c}^\dagger = \boldsymbol{V}\frac{\Sigma^2}{l-1}\boldsymbol{V}^\dagger$)
- The main goal of PCA is to identify the most meaningful basis!
- **What does the "most meaningful" even mean?**
- We assume that large variances means signal, i.e., that there is a large Signal to Noise ratio in our data!
- Let me explain [Check notebook pca_blobs]



J. Shlens: A Tutorial on Principal Component Analysis

Principal component analysis (PCA)

Summary:

- **We need to:**

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance
  - rotate the data so this direction becomes the first PC_0

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance
  - rotate the data so this direction becomes the first PC_0
  - find next direction perpendicular to PC_0 with the largest variance

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance
  - rotate the data so this direction becomes the first PC_0
  - find next direction perpendicular to PC_0 with the largest variance
  - rotate the data so this direction becomes the second PC_1

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance
  - rotate the data so this direction becomes the first PC_0
  - find next direction perpendicular to PC_0 with the largest variance
  - rotate the data so this direction becomes the second PC_1
  - $\ldots$

Principal component analysis (PCA)

Summary:

- **We need to:**
  - store data in $n \times m$ matrix, where m is the number of measured properties and n is the number of samples.
- **PCA will:**
  - center each dimension to zero
  - find direction with the largest variance
  - rotate the data so this direction becomes the first PC_0
  - find next direction perpendicular to PC_0 with the largest variance
  - rotate the data so this direction becomes the second PC_1
  - $\ldots$
  - profit

Why does PCA work and when it does not?

PCA assumptions:

Linearity

The new basis is build as a linear combination of the components of the original one.

Large Signal to Noise ratio

Principal components with larger associated variances represent the droids we are looking.

The principal components are ortogonal

This allows us to use SVD and we can be sure that we will get the optimal result (If the three assumptions are true!)

Check notebook pca_blobs on Kernel PCA.

PCA and phase transitions

- The PCA is in physics usually used as a first step towards supervised learning.
- But (for me) there is a much more exciting application of PCA.
- Automatic identification of phase-boundaries without a supervision.

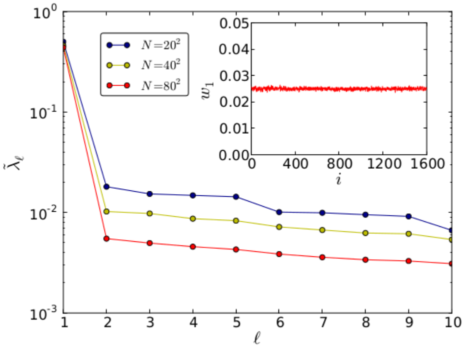
Test case the Ising model:

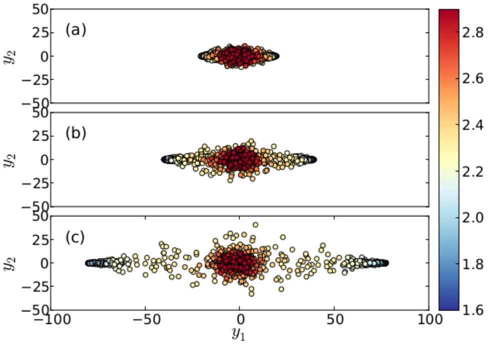
$$H = -J \sum_{\{ij\}} S_i S_j + h \sum_j S_j \tag{1}$$

Its a paradigmatic model for phase transitions and defines its universality class.

PCA and phase transitions in Ising model

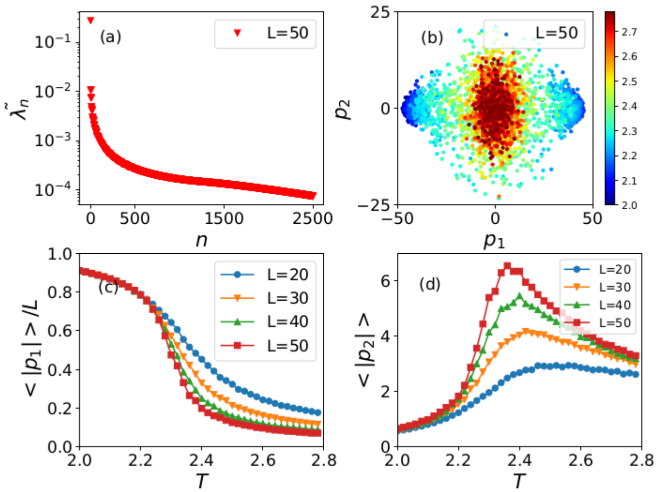
Let's say that we don't know what we should measure. Therefore we will store snapshots of spin configurations. They contain all the information necessary for investigation of order and phase transitions. PCA can be used to extract it.





Wang, Phys. Rev. B **94**, 195105 (2016)

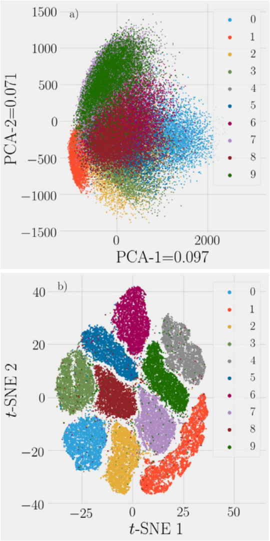
PCA and phase transitions in Ising model



Hu et al., Phys. Rev. E **95**, 062122 (2017)

Other dimension reduction techniques

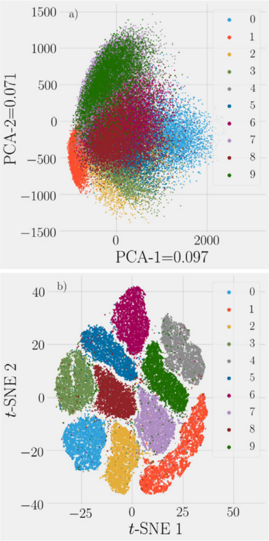
- t-SNE The basic idea is to associate a probability distribution to the neighborhood of each data point and keep similar instances together.



Physics Reports 810 1 (2019)

Other dimension reduction techniques

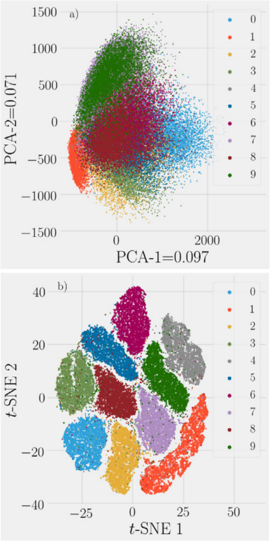
- t-SNE The basic idea is to associate a probability distribution to the neighborhood of each data point and keep similar instances together.
- Isomap - Preserves the number of nodes between two data points.



Physics Reports 810 1 (2019)

Other dimension reduction techniques

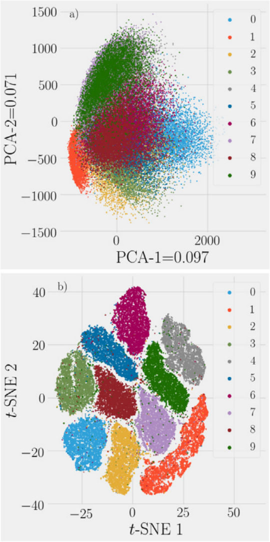
- t-SNE The basic idea is to associate a probability distribution to the neighborhood of each data point and keep similar instances together.
- Isomap - Preserves the number of nodes between two data points.
- Random Projection - Yep, it uses a random linear projection and it works. World is a strange place.



Physics Reports 810 1 (2019)

Other dimension reduction techniques

- t-SNE The basic idea is to associate a probability distribution to the neighborhood of each data point and keep similar instances together.
- Isomap - Preserves the number of nodes between two data points.
- Random Projection - Yep, it uses a random linear projection and it works. World is a strange place.
- ...



Physics Reports 810 1 (2019)

Clustering

Basic concepts

- The aim of clustering is to group unlabeled data into clusters according to some similarity or distance measure

Basic concepts

- The aim of clustering is to group unlabeled data into clusters according to some similarity or distance measure
- Probably the simplest way to seek a hidden structure

Basic concepts

- The aim of clustering is to group unlabeled data into clusters according to some similarity or distance measure
- Probably the simplest way to seek a hidden structure
- Lots of methods

Basic concepts

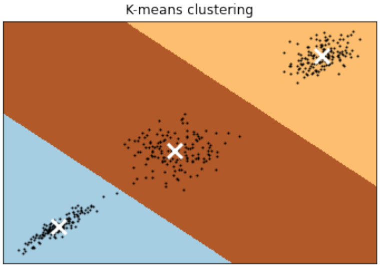
- The aim of clustering is to group unlabeled data into clusters according to some similarity or distance measure
- Probably the simplest way to seek a hidden structure
- Lots of methods
- We will talk about more standard ones: K-means and DB-clustering

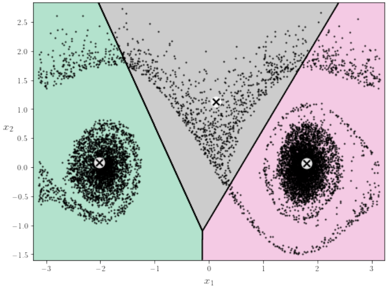
Basic concepts

- The aim of clustering is to group unlabeled data into clusters according to some similarity or distance measure
- Probably the simplest way to seek a hidden structure
- Lots of methods
- We will talk about more standard ones: K-means and DB-clustering
- and one less standard method which requires NN

K-Means

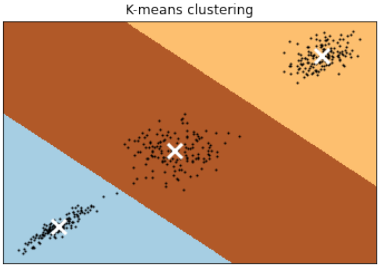
- Let's have N unlabeled measurements $\boldsymbol{x}_i$, where $\boldsymbol{x}_i \in \mathbb{R}^p$

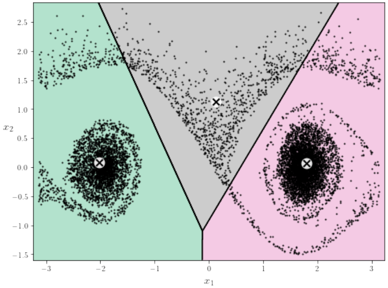




K-Means

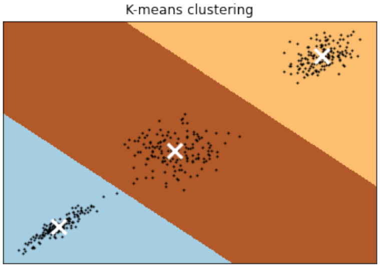
- Let's have N unlabeled measurements $\boldsymbol{x}_i$, where $\boldsymbol{x}_i \in \mathbb{R}^p$
- Lets assume K cluster centers, i.e., cluster means $\boldsymbol{\mu}_k \in \mathbb{R}^p$

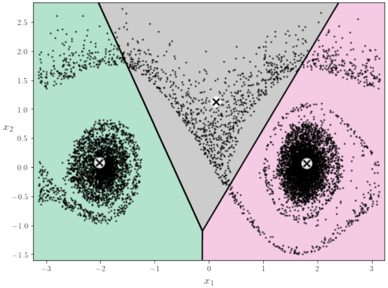




K-Means

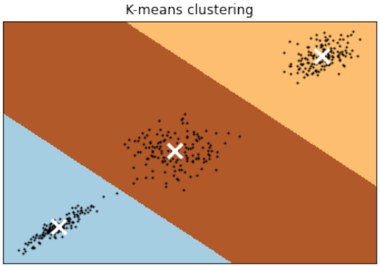
- Let's have N unlabeled measurements $\boldsymbol{x}_i$, where $\boldsymbol{x}_i \in \mathbb{R}^p$
- Lets assume K cluster centers, i.e., cluster means $\boldsymbol{\mu}_k \in \mathbb{R}^p$
- We don't know yet where they are

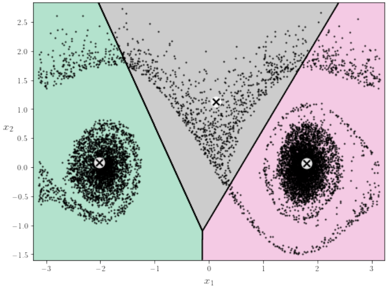




K-Means

- Let's have N unlabeled measurements $\boldsymbol{x}_i$, where $\boldsymbol{x}_i \in \mathbb{R}^p$
- Lets assume K cluster centers, i.e., cluster means $\mu_k \in \mathbb{R}^p$
- We don't know yet where they are
- The actual task is to:



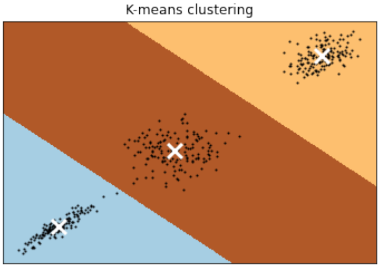


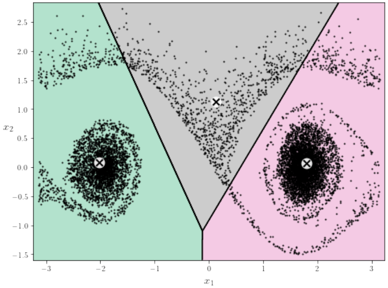
K-Means

- Let's have N unlabeled measurements $\boldsymbol{x}_i$, where $\boldsymbol{x}_i \in \mathbb{R}^p$
- Lets assume K cluster centers, i.e., cluster means $\boldsymbol{\mu}_k \in \mathbb{R}^p$
- We don't know yet where they are
- The actual task is to:

  Find the cluster means (positions) and the data point assignments to them in order to minimize the following cost function:

$$\mathcal{C}(\{\mathbf{x}, \boldsymbol{\mu}\}) = \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)^2 \text{, where } r_{ik} \in \{0, 1\}$$

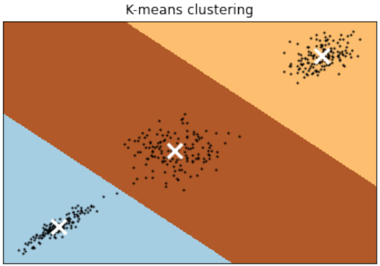


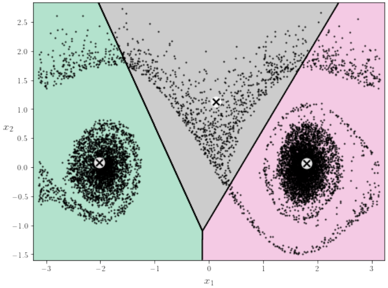


K-Means : optimization procedure

- Take assignments r, minimize $\mathcal{C}$ with respect to μ_k

$$\mu_k = \frac{1}{N_k} \sum_i r_{ik} \mathbf{x}_i$$



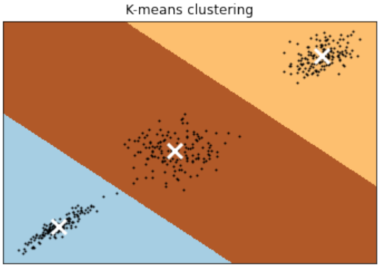


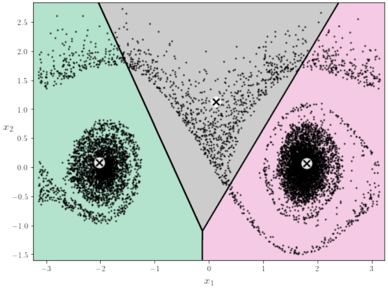
K-Means : optimization procedure

- Take assignments r, minimize $\mathcal{C}$ with respect to μ_k

$$\mu_k = \frac{1}{N_k} \sum_i r_{ik} \mathbf{x}_i$$

- Take means μ, minimize $\mathcal{C}$ with respect to r_{ik}



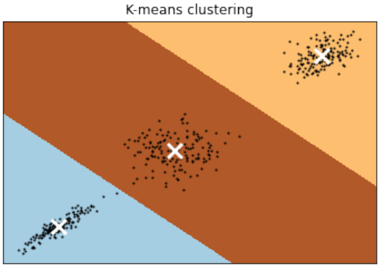


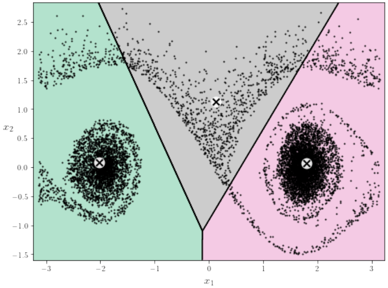
K-Means : optimization procedure

- Take assignments r, minimize $\mathcal{C}$ with respect to μ_k

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_i r_{ik} \mathbf{x}_i$$

- Take means μ, minimize $\mathcal{C}$ with respect to r_{ik}
- This is achieved by assigning each data point to its nearest cluster mean



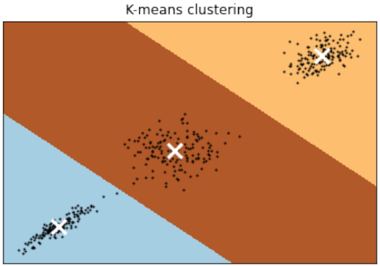


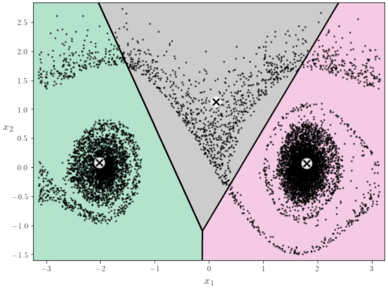
K-Means : optimization procedure

- Take assignments r, minimize $\mathcal{C}$ with respect to μ_k

$$\mu_k = \frac{1}{N_k} \sum_i r_{ik} \mathbf{x}_i$$

- Take means μ, minimize $\mathcal{C}$ with respect to r_{ik}
- This is achieved by assigning each data point to its nearest cluster mean
- Repeat! K-means algorithm is guaranteed to converge.



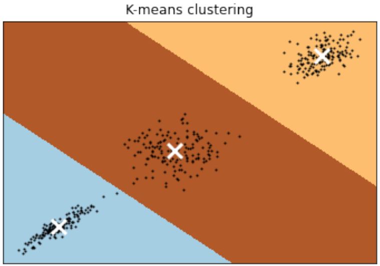


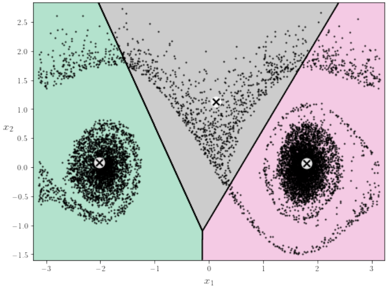
K-Means : optimization procedure

- Take assignments r, minimize $\mathcal{C}$ with respect to μ_k

$$\mu_k = \frac{1}{N_k} \sum_i r_{ik} \mathbf{x}_i$$

- Take means μ, minimize $\mathcal{C}$ with respect to r_{ik}
- This is achieved by assigning each data point to its nearest cluster mean
- Repeat! K-means algorithm is guaranteed to converge.
- Be aware that K-means can lead to spurious results





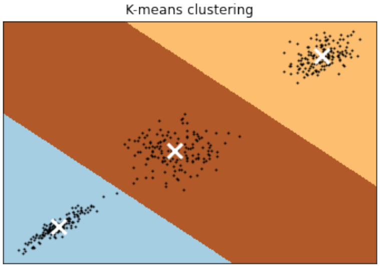
K-Means: Pros. and Cons.

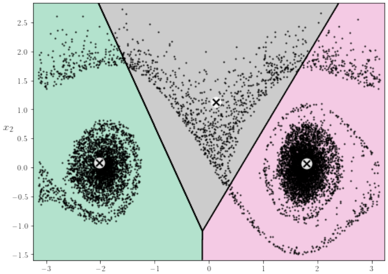
Advantages

- It is fast and scalable
- It converges (it will finish)
- Can be improved

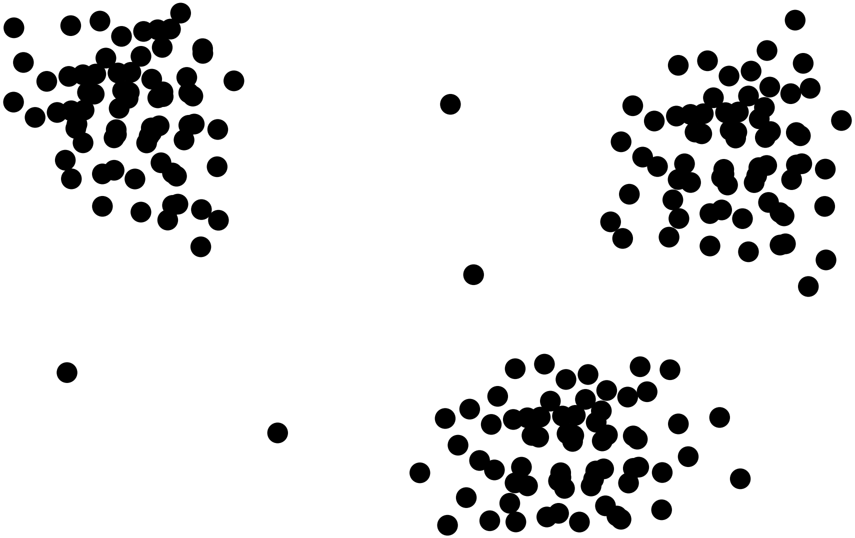
Disadvantages

- It must be run several times
- The number of clusters must be specify
- Does not behave well when the clusters have significantly
    - different size,
    - different densities,
    - nonspherical shapes

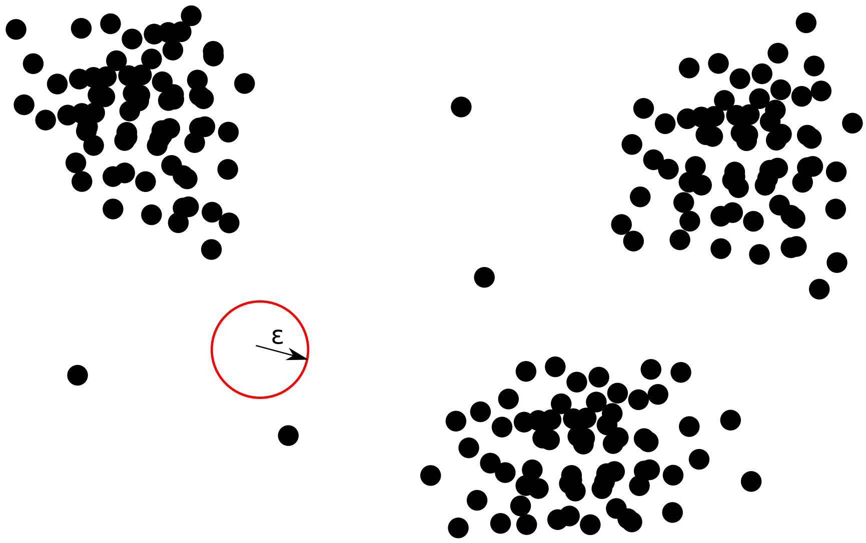




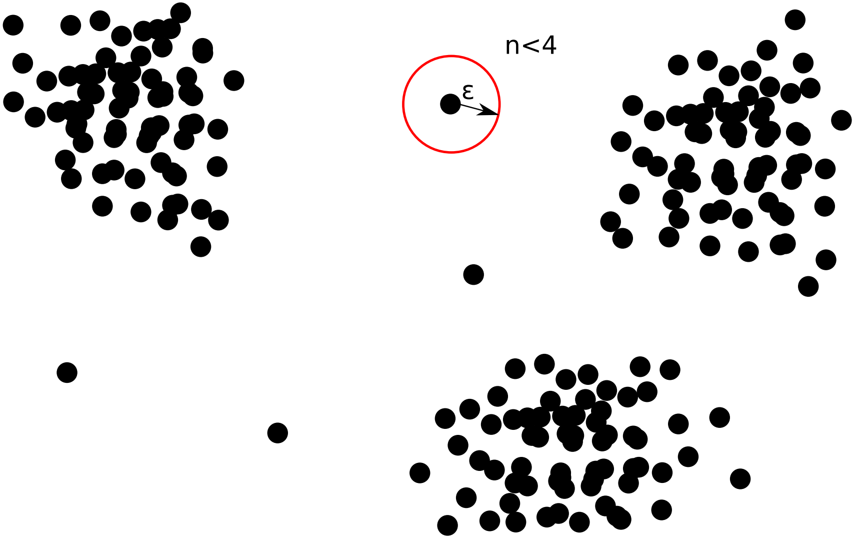
Density-based (DB) clustering DBSCAN



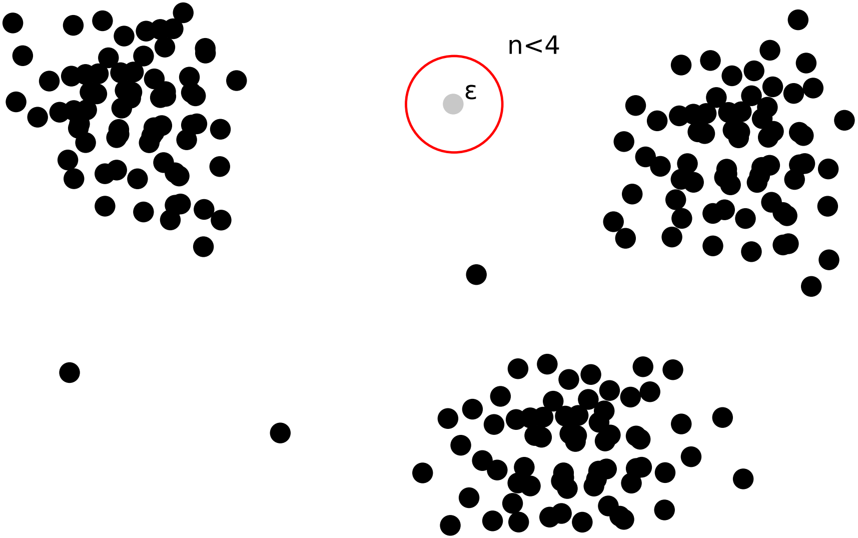
Density-based (DB) clustering DBSCAN



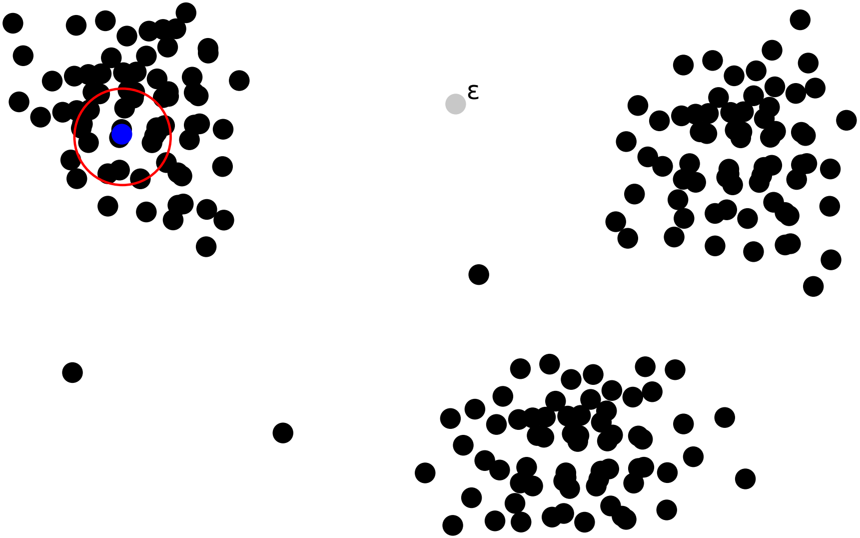
Density-based (DB) clustering DBSCAN



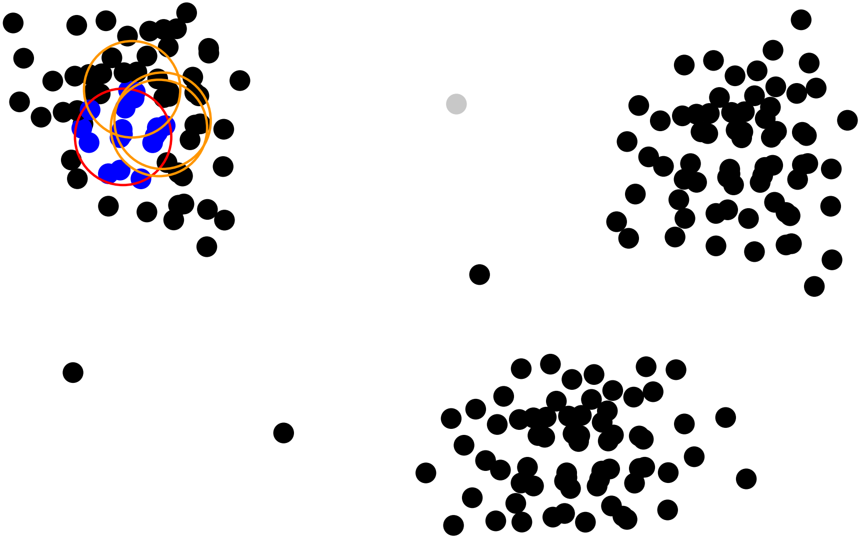
Density-based (DB) clustering DBSCAN



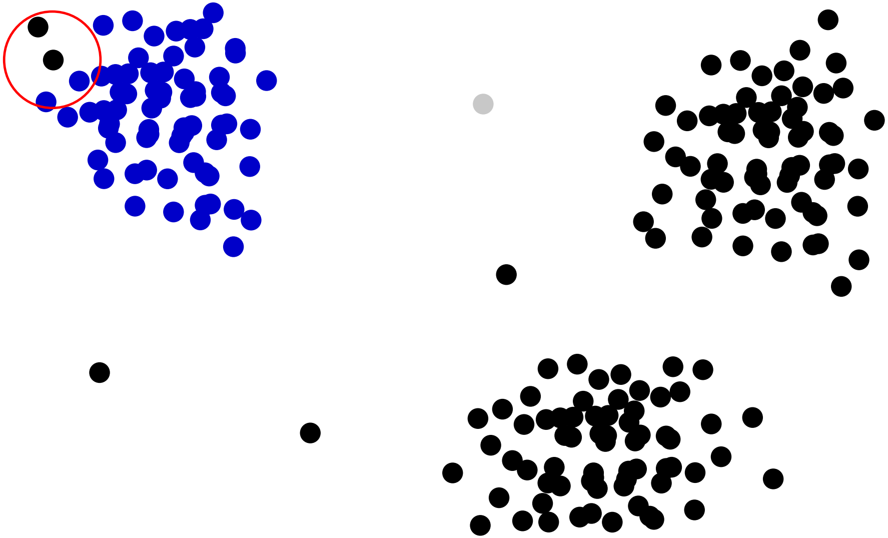
Density-based (DB) clustering DBSCAN



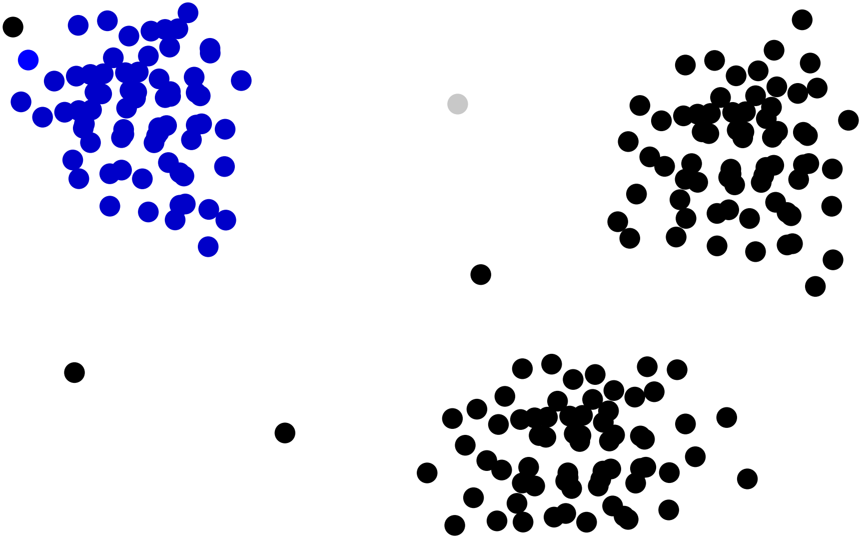
Density-based (DB) clustering DBSCAN



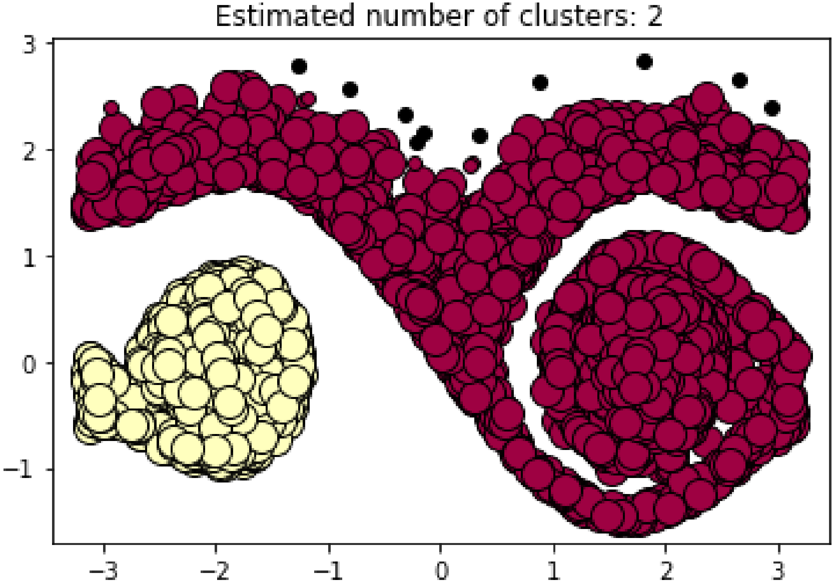
Density-based (DB) clustering DBSCAN



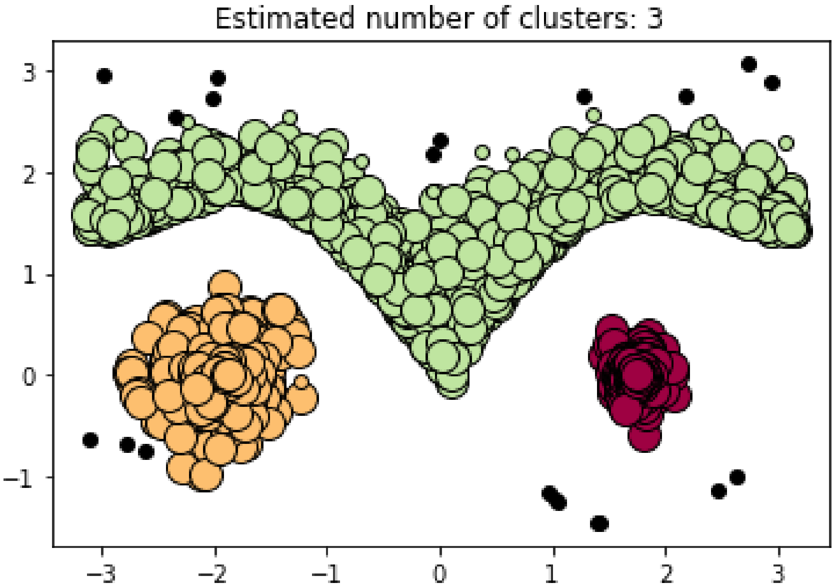
Density-based (DB) clustering DBSCAN



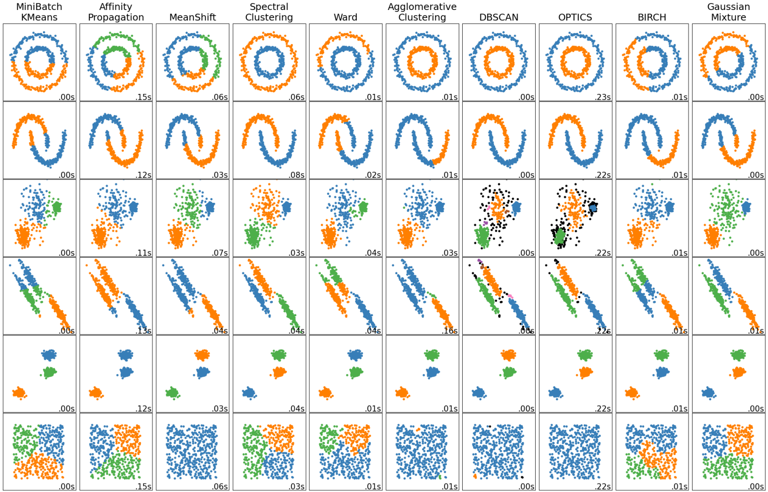
Density-based (DB) clustering DBSCAN



Density-based (DB) clustering DBSCAN



Other clustering techniques

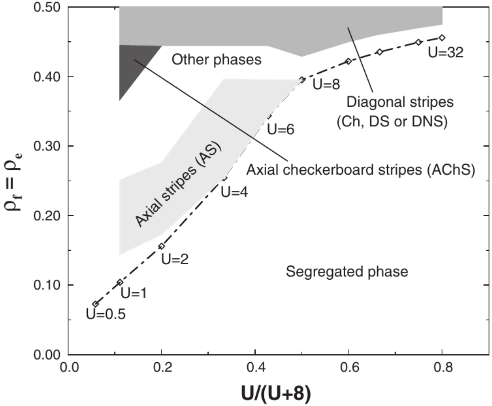


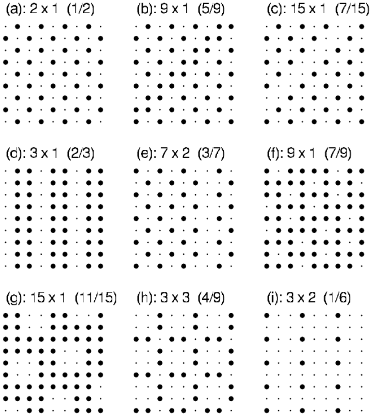
Comparing different clustering algorithms in sklearn

Unsupervised phase classification

More complicated systems, e.g., Phase diagram of correlated electrons

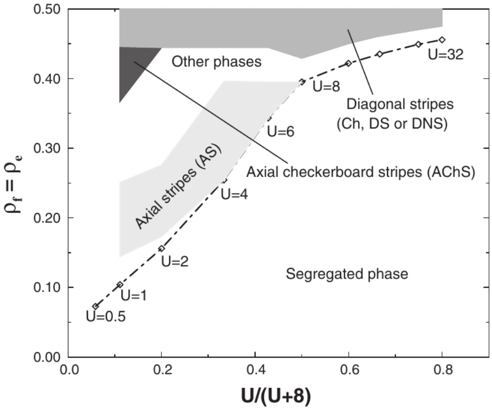
$$H_{FK} = -t \sum_{\langle i,j \rangle} \hat{d}_i^\dagger \hat{d}_j + U \sum_i \hat{f}_i^\dagger \hat{f}_i \, \hat{d}_i^\dagger \hat{d}_i$$

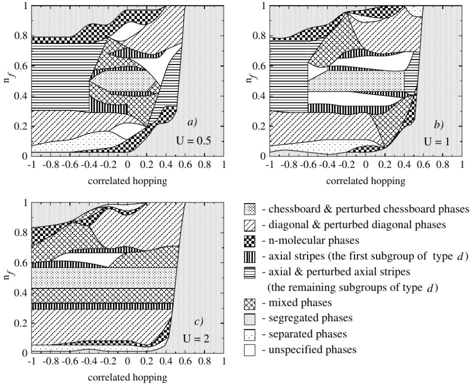




More complicated systems, e.g., Phase diagram of correlated electrons

$$H_{FK} = -t \sum_{\langle i,j \rangle} \hat{d}_i^\dagger \hat{d}_j + U \sum_i \hat{f}_i^\dagger \hat{f}_i \, \hat{d}_i^\dagger \hat{d}_i$$

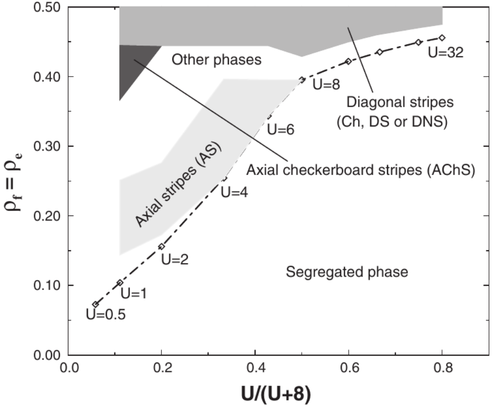




Phase diagram of generalized FKM [Čenčariková et al., (2011)]

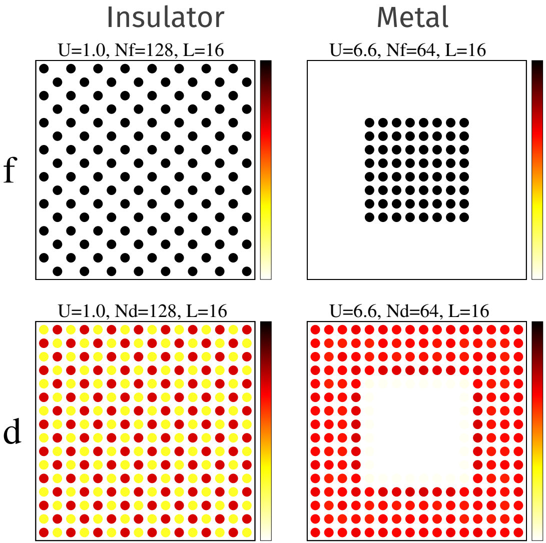
More complicated systems, e.g., Phase diagram of correlated electrons

$$H_{FK} = -t \sum_{\langle i,j \rangle} \hat{d}_i^\dagger \hat{d}_j + U \sum_i \hat{f}_i^\dagger \hat{f}_i \hat{d}_i^\dagger \hat{d}_i$$



- The classification of GS phases in the FKM was for years a manual, lengthy and cumbersome task
- Yet, it seems to be suited for the modern Machine learning (ML) techniques
- But we need something better then standard techniques

Different ordering (phases) have different physical properties



Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don’t know:

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are
  - how many there are

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
    - where the phases are
    - what type they are
    - how many there are
- We know:

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are
  - how many there are
- We know:
  - GS configuration at any $p \equiv (U, \rho)$

Automatic classification - basic principles

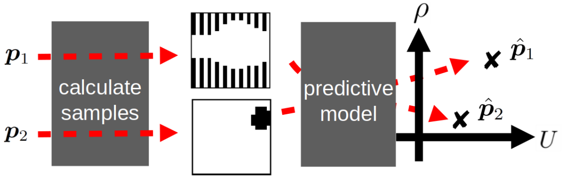
We want to construct the phase diagram without supervision!

- We don't know:
    - where the phases are
    - what type they are
    - how many there are
- We know:
    - GS configuration at any $p \equiv (U, \rho)$

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are
  - how many there are
- We know:
  - GS configuration at any $p \equiv (U, \rho)$



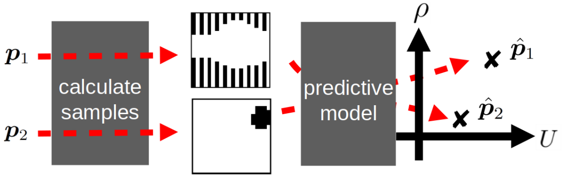
We can train neural network to infer $\overline{\boldsymbol{p}} \equiv (\overline{U}, \overline{\rho})$ by minimizing $\delta \boldsymbol{p} = ||\boldsymbol{p} - \overline{\boldsymbol{p}}||$

- Input is image-like

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are
  - how many there are
- We know:
  - GS configuration at any $p \equiv (U, \rho)$



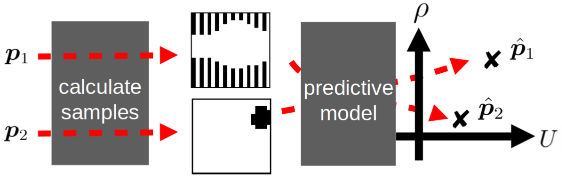
We can train neural network to infer $\bar{\boldsymbol{p}} \equiv (\bar{U}, \bar{\rho})$ by minimizing $\delta \boldsymbol{p} = ||\boldsymbol{p} - \bar{\boldsymbol{p}}||$

- Input is image-like
- We needed DNN (namely CNN), but other predictive models can be used

Automatic classification - basic principles

We want to construct the phase diagram without supervision!

- We don't know:
  - where the phases are
  - what type they are
  - how many there are
- We know:
  - GS configuration at any $p \equiv (U, \rho)$

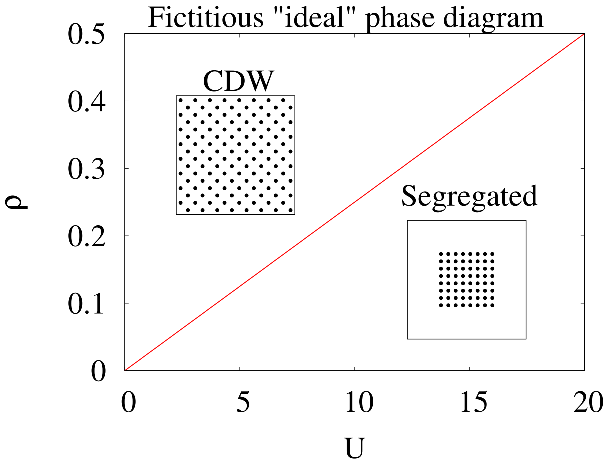


We can train neural network to infer $\overline{\boldsymbol{p}} \equiv (\overline{U}, \overline{\rho})$ by minimizing $\delta \boldsymbol{p} = ||\boldsymbol{p} - \overline{\boldsymbol{p}}||$

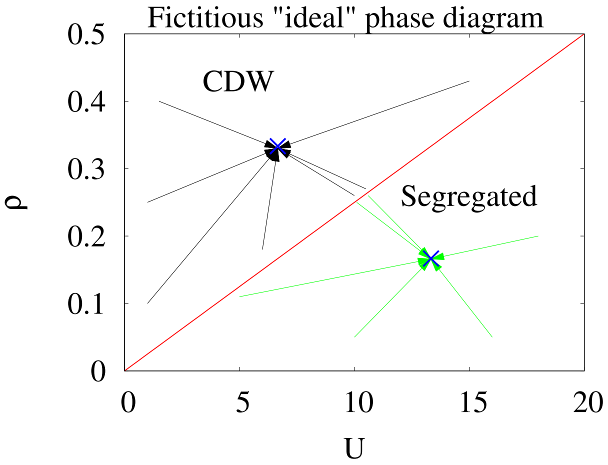
- Input is image-like
- We needed DNN (namely CNN), but other predictive models can be used
- The MSE loss function is defined as

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{p}} N_{\mathrm{x}}} \sum_{\boldsymbol{p}} \sum_{\boldsymbol{x}} \left|\left| \boldsymbol{p} - \overline{\boldsymbol{p}(x)} \right|\right|^2$$

Automatic classification - basic principles



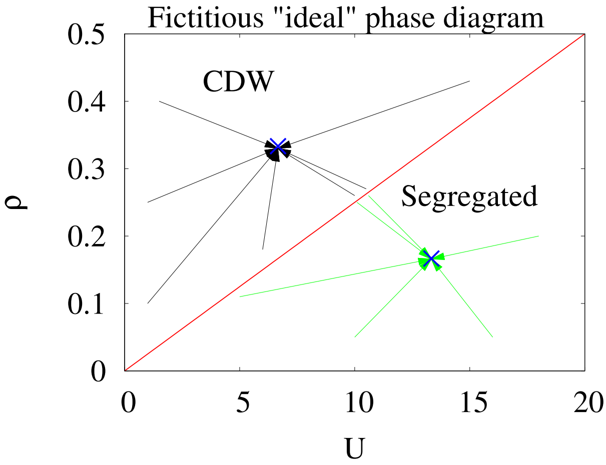
Automatic classification - basic principles



When is MSE loss function minimal?

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{p}} N_{\mathrm{x}}} \sum_{\boldsymbol{p}} \sum_{\boldsymbol{x}} \left|\left|\delta \boldsymbol{p}(x)\right|\right|^2$$

Automatic classification - basic principles



When is MSE loss function minimal?

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{p}} N_{\mathrm{x}}} \sum_{\boldsymbol{p}} \sum_{\boldsymbol{x}} ||\delta \boldsymbol{p}(x)||^2$$

The vector-field divergence signals phase boundaries

$$\nabla_{\boldsymbol{p}} \cdot \delta \boldsymbol{p} = \frac{\partial \delta U}{\partial U} + \frac{\partial \delta \rho}{\partial \rho}$$

Final phase diagram