Lecture 6:
More tricks of the trade

Bingqing Cheng

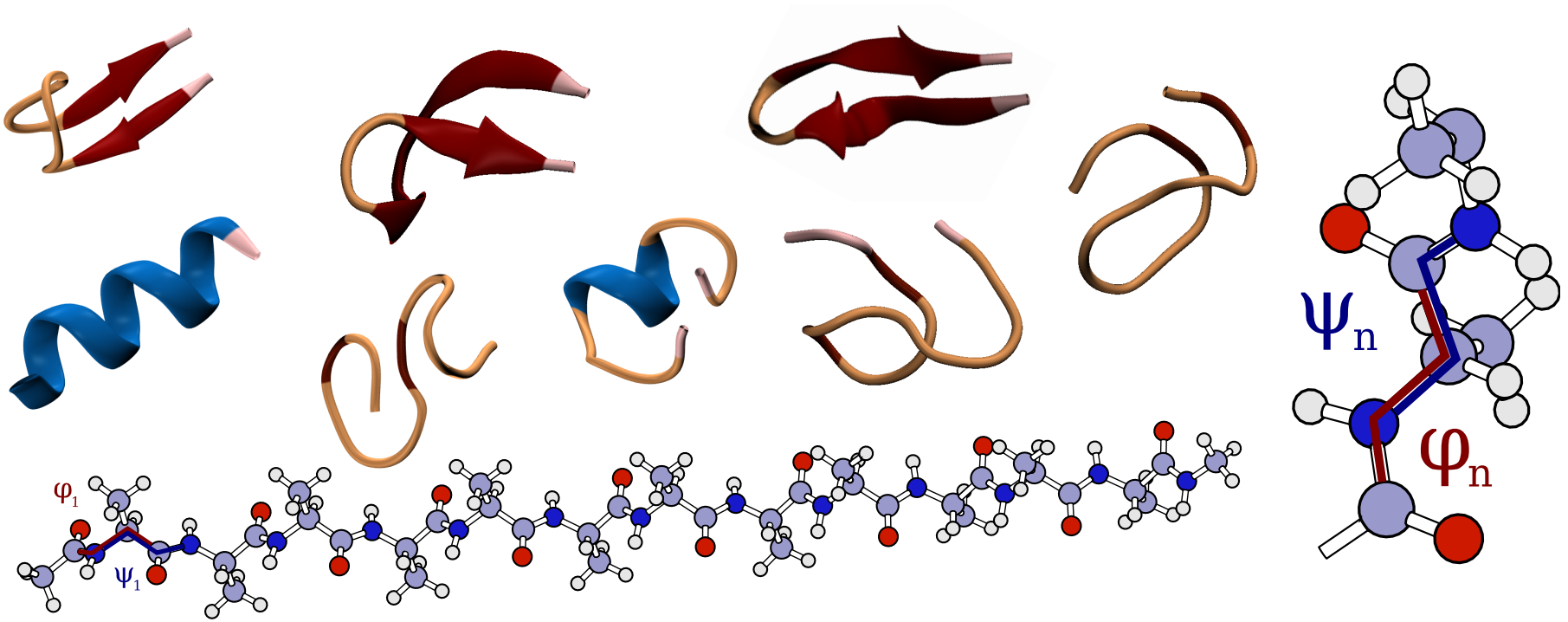
Trinity College, the University of Cambridge

bc509@cam.ac.uk

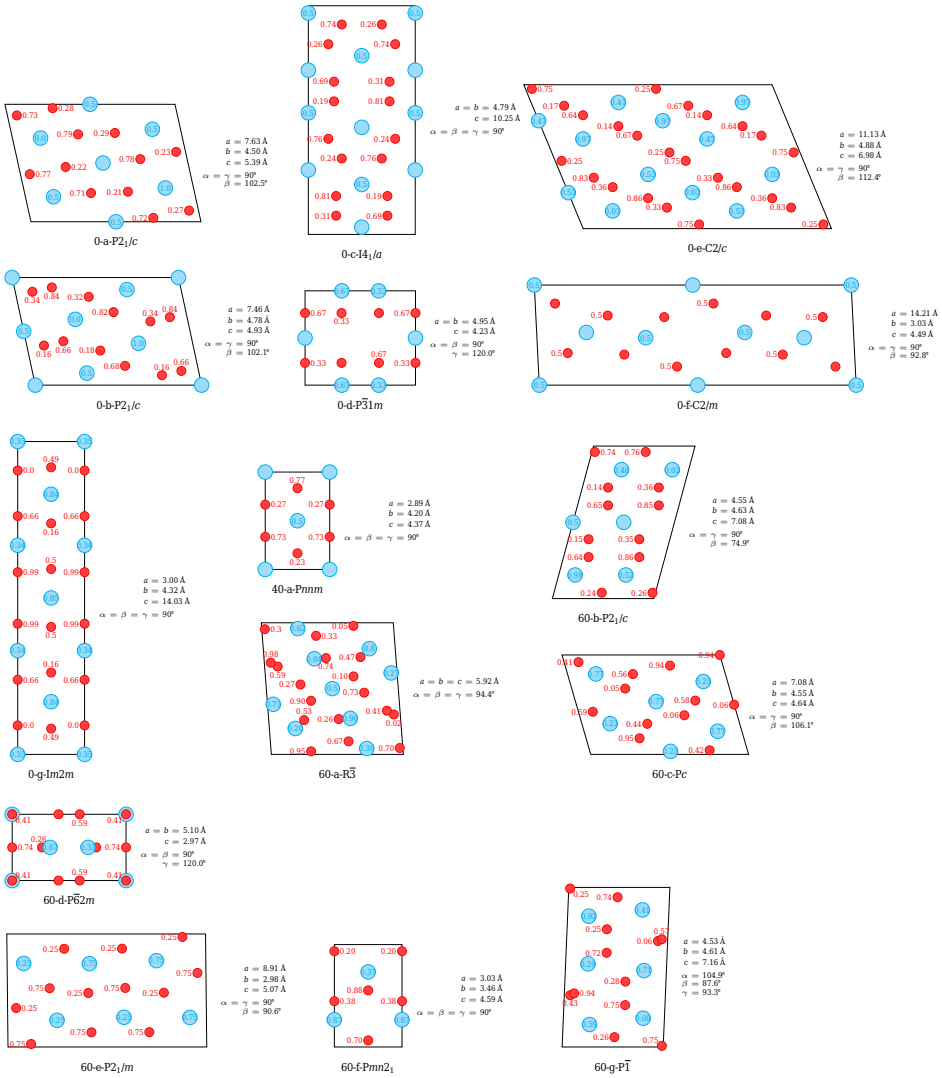
Recap

- The basic concept in machine learning
- Linear regression, nonlinear regression, kernel regression
- Neural networks, deep learning
- Representations of molecules
- Dimensionality reduction

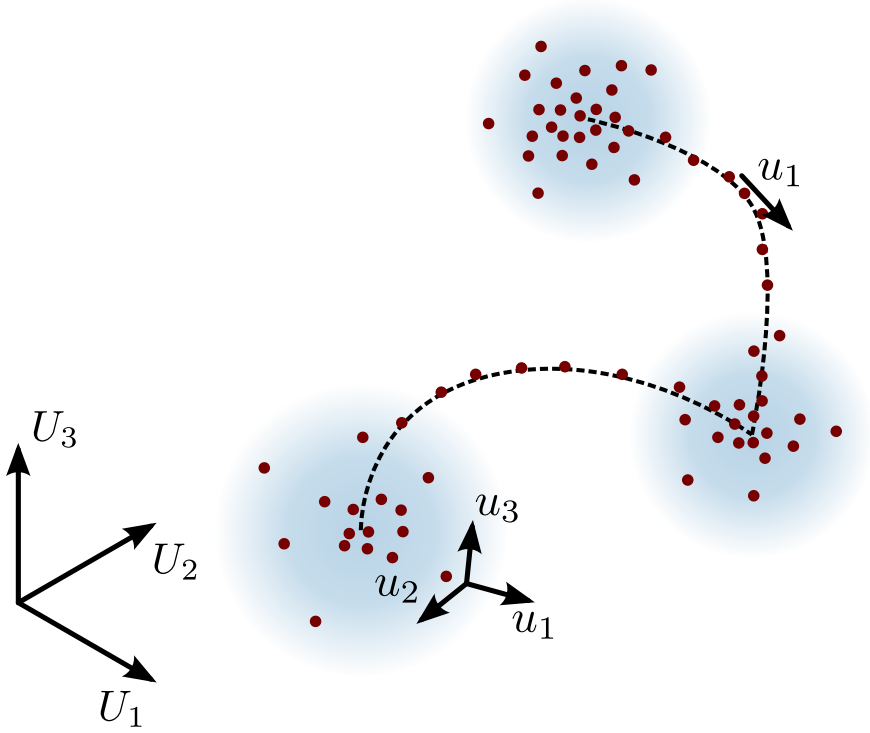
Dimensionality reduction for chemical data

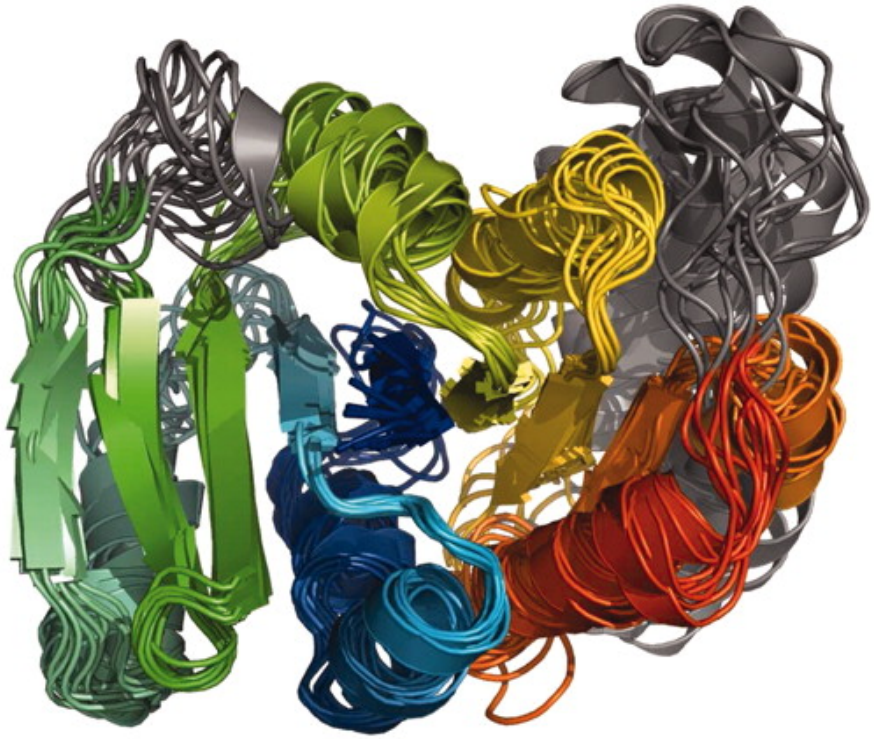


Dimensionality reduction for chemical data



Dimensionality reduction for chemical data





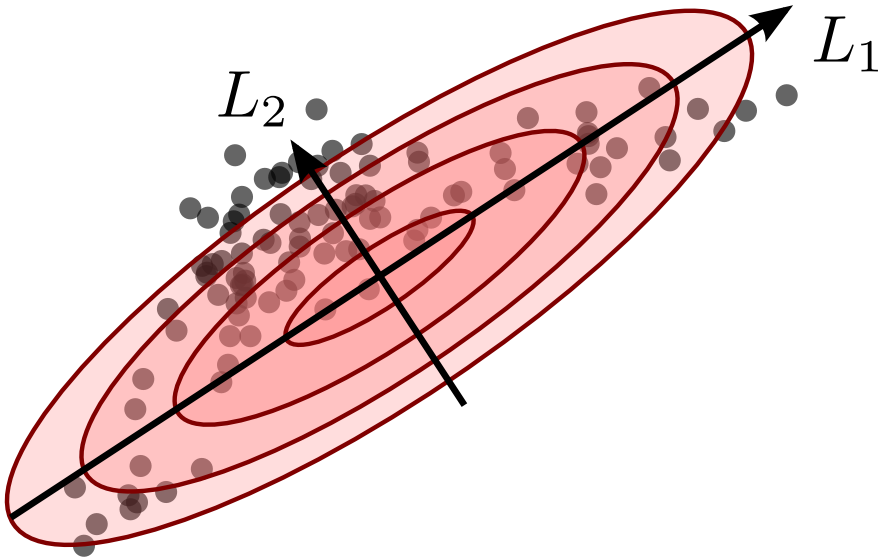
Outline

- More tricks of the trade
  - Sparsification
    - Single value decomposition (SVD)
    - CUR decomposition
    - Farthest Point Sampling (FPS)
  - Clustering
    - Gaussian mixture model (GMM)
    - K-means
    - DBSCAN
- Representing structual data

Principal component analysis

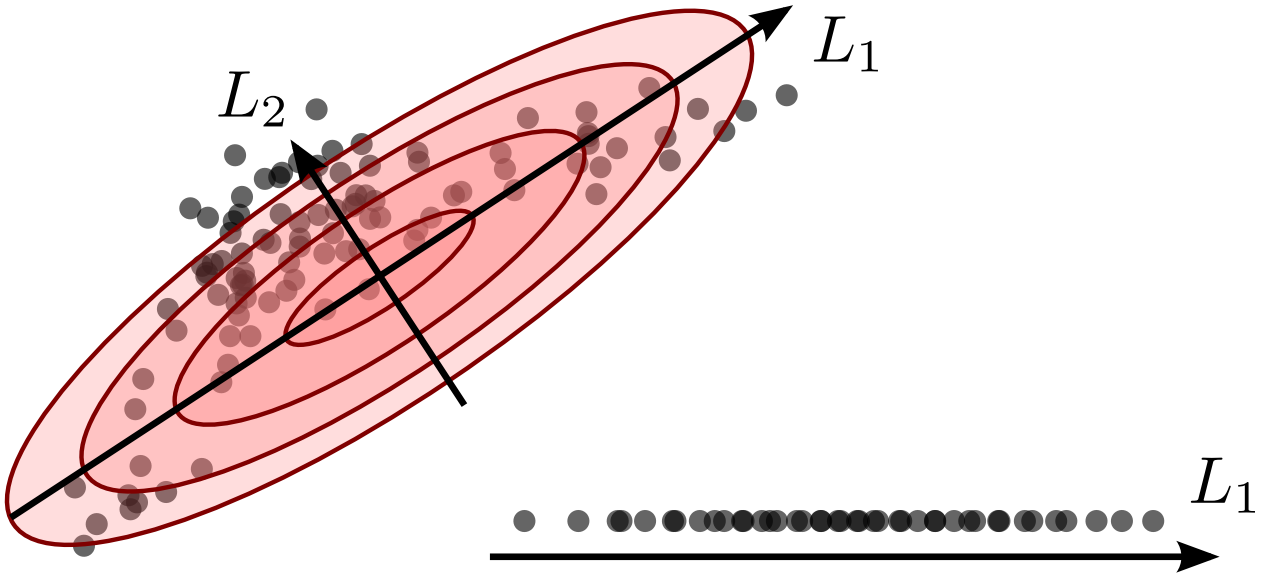
Question:
What is preserved during PCA?

Principal component analysis



Question:
What is preserved during PCA?

Principal component analysis



Question:
What is preserved during PCA?

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i \mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T \mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T \mathbf{x}$

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i\mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T\mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T\mathbf{x}$

Given d , how to reserve the largest amount of variance?

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i\mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T\mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T\mathbf{x}$

Keep the first d eigenvectors of the covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$

Eigenvalues and eigenvectors

- The covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$: $D \times D$ form.
- Eigenvalues $\{\lambda^j\}$
- Corresponding eigenvectors $\{\mathbf{v}^j\}$ of the matrix

Eigenvalues and eigenvectors fulfills

$$\mathbf{C}\mathbf{v}^j = \lambda^j\mathbf{v}^j$$

for $j = 1 \ldots D$.
One can find the eigenvalues $\{\lambda^j\}$ by solving

$$det(\mathbf{C} - \lambda\mathbf{I}) = 0$$

Eigenvalues and eigenvectors

- The covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$: $D \times D$ form.
- Eigenvalues $\{\lambda^j\}$.Diagonal matrix $\Sigma = \mathbf{diag}(\lambda^1, \lambda^2, \ldots, \lambda^d)$
- Corresponding eigenvectors $\mathbf{V} = [\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^d]$ of the matrix.
- Normalization condition $\mathbf{V}\mathbf{V}^T = \mathbf{I}$

The covariance matrix can be expressed as

$$\mathbf{C} = \mathbf{V}\Sigma\mathbf{V}^T$$

equalvalently

$$\Sigma = \mathbf{V}^T\mathbf{C}\mathbf{V}$$

Singular value decomposition (SVD)

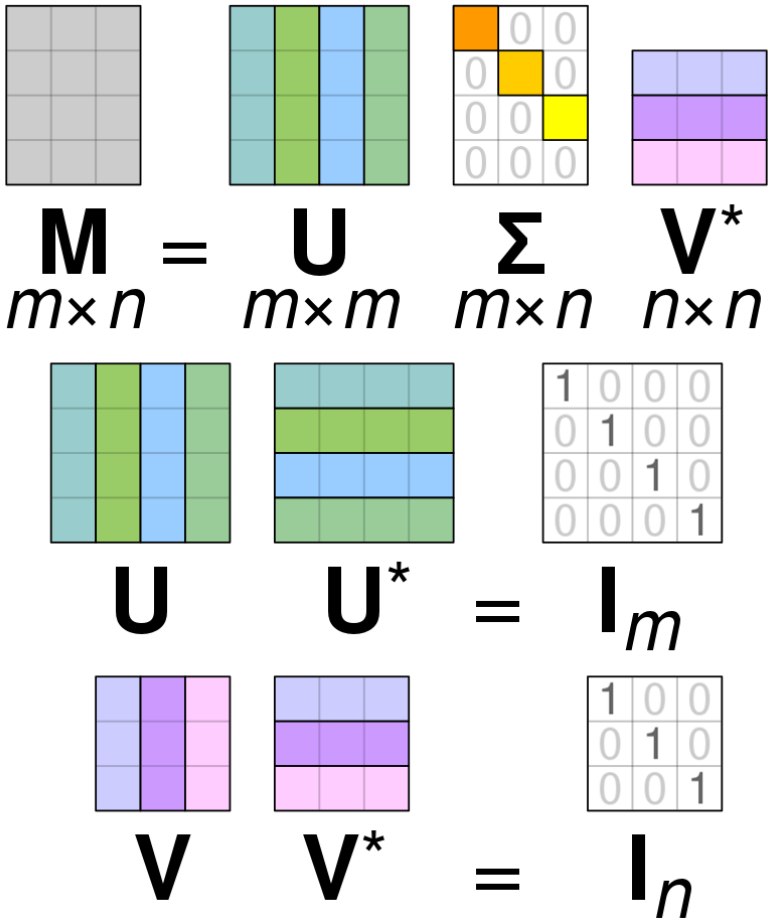
a general matrix $\mathbf{M} \in \mathcal{R}^{m \times n}$ can be represented in the SVD form:

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$$

- $\mathbf{U} = [\mathbf{u}_2, \mathbf{u}_2, \ldots, \mathbf{u}_m]$
- $\mathbf{V} = [\mathbf{v}_2, \mathbf{v}_2, \ldots, \mathbf{v}_n]$
- $\Sigma = \mathbf{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$

Singular value decomposition (SVD)

a general matrix $\mathbf{M} \in \mathcal{R}^{m \times n}$ can be represented in the SVD form:



Singular value decomposition (SVD)

- $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$
- $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$
- $\Sigma = \mathbf{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)$

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$$

Problem: each vector $\mathbf{v}_i$ and $\mathbf{u}_i$ a linear combination of the original rows and columns of the design matrix, so they may not be particularly informative or meaningful.

Example: gene expression data, the eigenvector of images, and a linear combination of atomic environments.

CUR Decomposition

Similar with SVD, but using the actual rows and columns for the design matrix for the decomposition.

$$\mathbf{X} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{R}$$

- $\mathbf{C} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$
- $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k]^T$
- $\mathbf{U}$: a $k \times k$ matrix

Many ways of selecting $\mathbf{C}$ and $\mathbf{R}$:

- deterministic
- probabilistic

"CUR matrix decompositions for improved data analysis" PNAS 2009

CUR Decomposition

Similar with SVD, but using the actual rows and columns for the design matrix for the decomposition.

$$\mathbf{X} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{R}$$

- $\mathbf{C} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$
- $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k]^T$
- $\mathbf{U}$: a $k \times k$ matrix

Q: What is the physical significance of CUR decomposition?

"CUR matrix decompositions for improved data analysis" PNAS 2009

CUR Decomposition

Similar with SVD, but using the actual rows and columns for the design matrix for the decomposition.

$$\mathbf{X} \approx \mathbf{C}\,\mathbf{U}\,\mathbf{R}$$

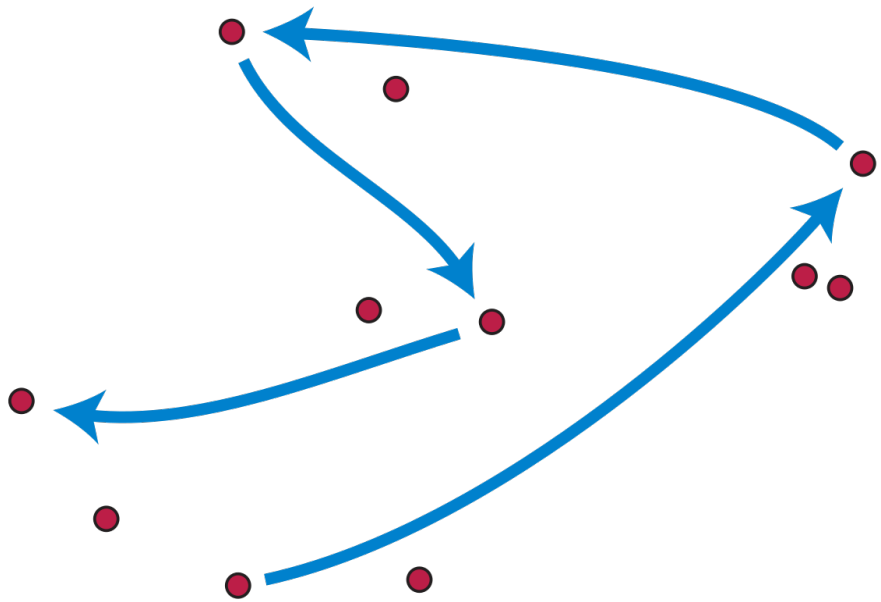
- $\mathbf{C} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$
- $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k]^T$
- $\mathbf{U}$: a $k \times k$ matrix

A: feature selection and data sparsification

"CUR matrix decompositions for improved data analysis" PNAS 2009

Farthest point sampling (FPS)

- Aim to select uniformly-spaced reference points
- Successive points are chosen so as to maximize the Euclidean distance between them.
  - Arbitrarily selecting the first fingerprint,
  - each subsequent one is chosen as
    $k = \text{argmax}(\min_j |\mathbf{x}_k - \mathbf{x}_j|)$



Outline

- More tricks of the trade
  - Sparsification
    - Single value decomposition (SVD)
    - CUR decomposition
    - Farthest Point Sampling (FPS)
  - Clustering
    - Gaussian mixture model (GMM)
    - K-means
    - DBSCAN
- Representing structual data

The IRIS dataset

- Measures of four morphological features (petal/sepal width/length) of a total of 150 samples of three species of iris
- 4D dataset, strong correlation between the indicators, but also spread within one species

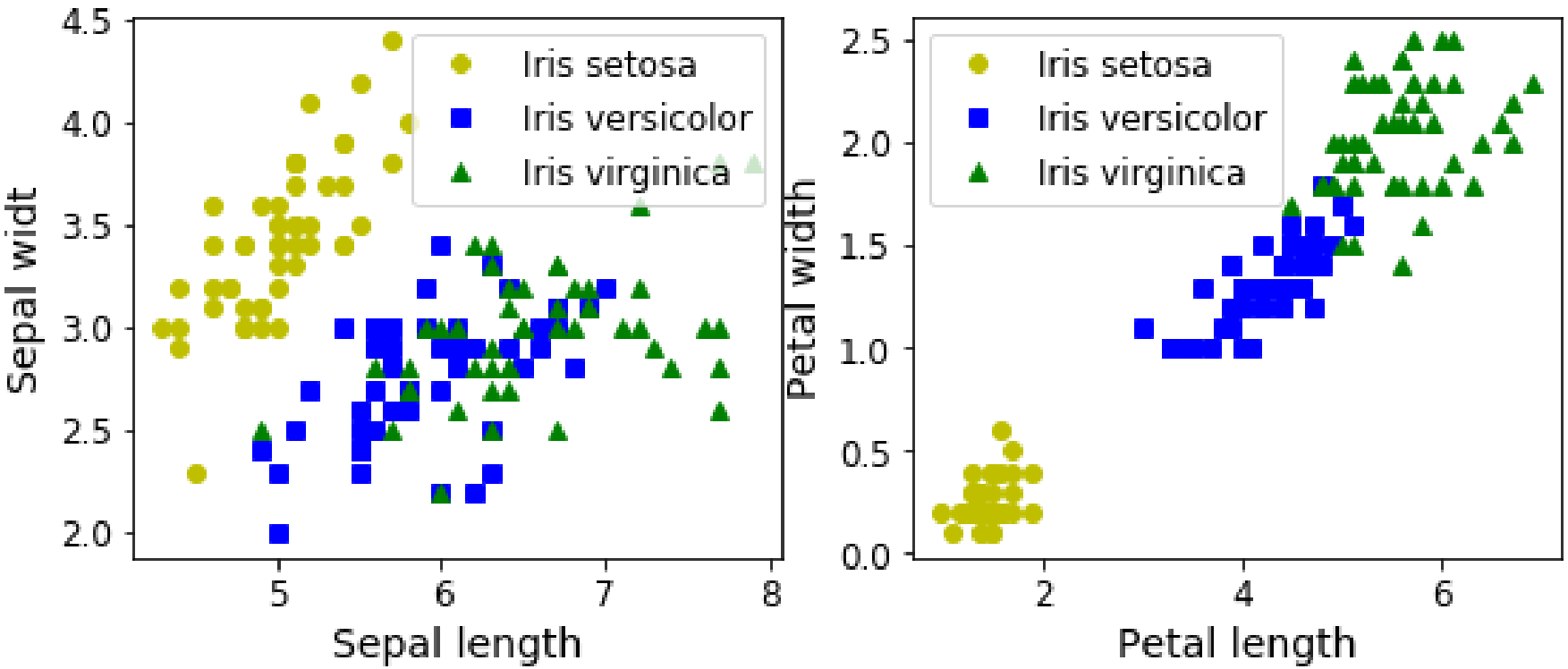
Iris Setosa

Iris Versicolor

Iris Virginica

The IRIS dataset

- Measures of four morphological features (petal/sepal width/length) of a total of 150 samples of three species of iris
- 4D dataset, strong correlation between the indicators, but also spread within one species



The IRIS dataset

- Clearly clustered in 2D.
- Versicolor and Virginica are pretty close... and they look quite similar!

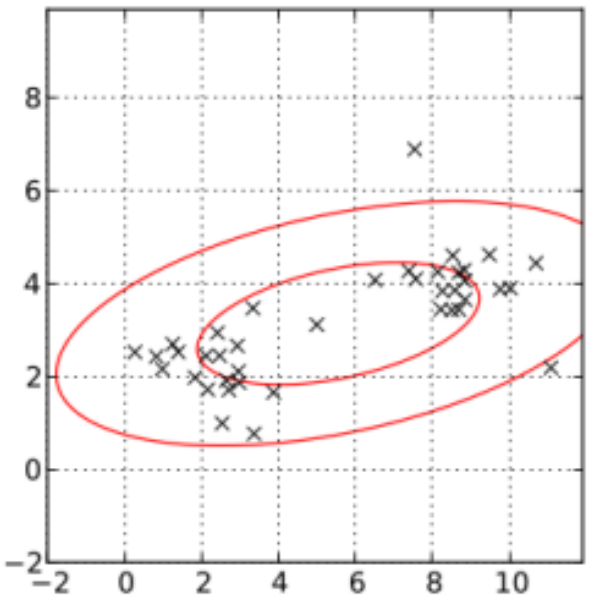
Iris Setosa

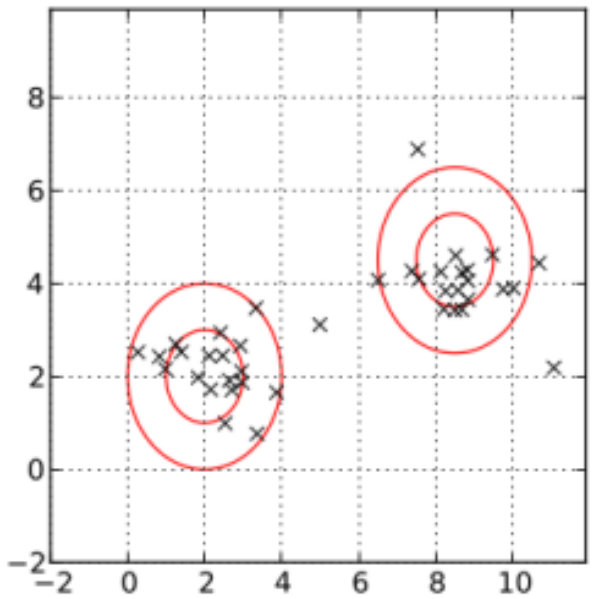
Iris Versicolor

Iris Virginica

Gaussian mixture model (GMM)

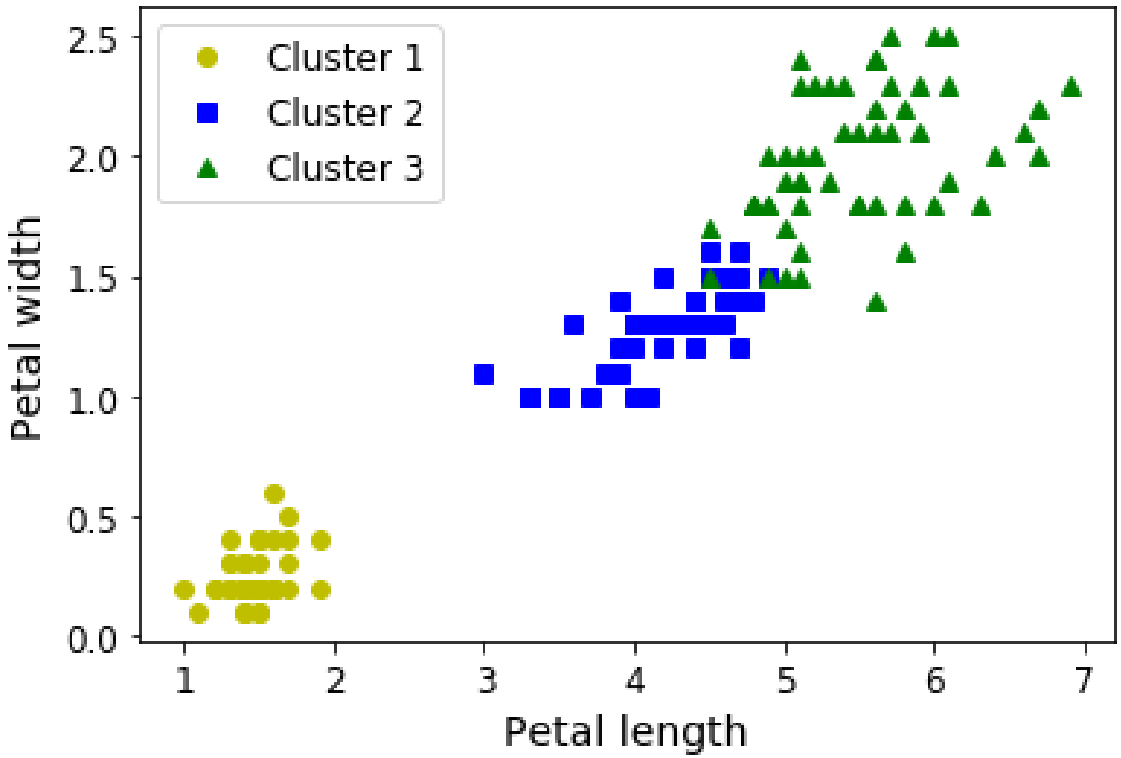
- good for multimodal distribution, and each blob is Gaussian-like
- K components:
  - a mean of μ_k and a covariance matrix of $\mathbf{C}_k$





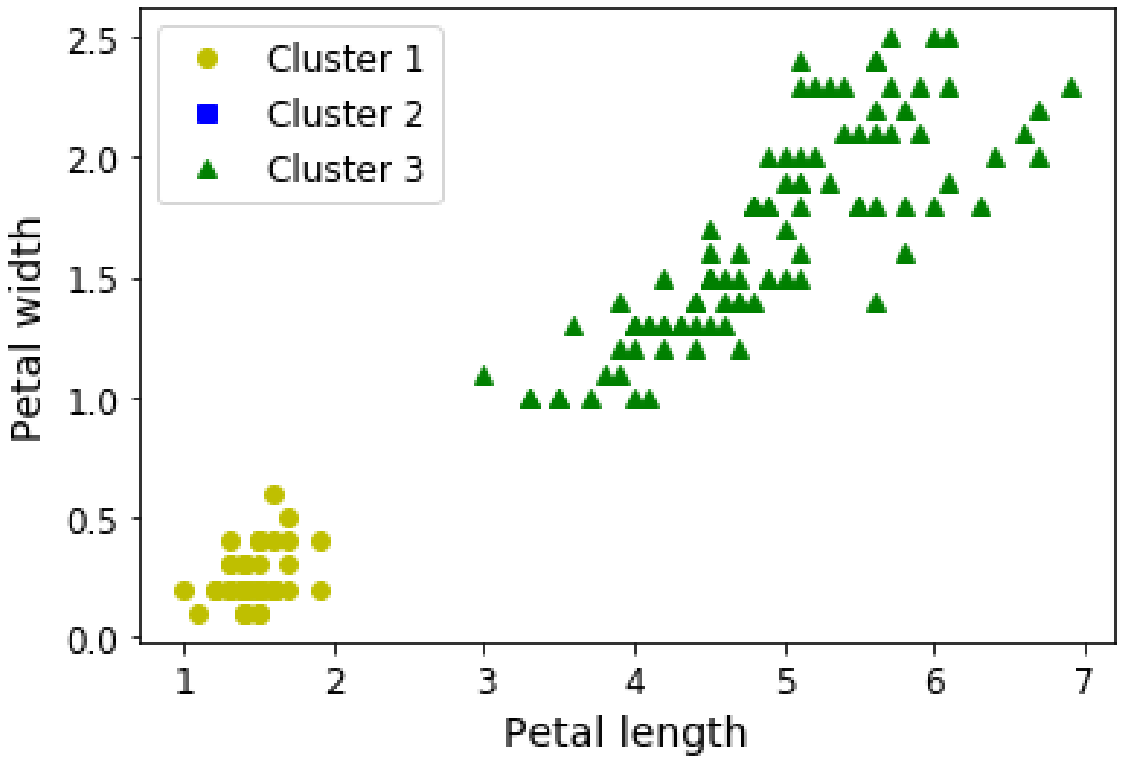
Gaussian mixture model (GMM)

- good for multimodal distribution, and each blob is Gaussian-like
- K components:
  - a mean of μ_k and a covariance matrix of $\mathbf{C}_k$



Gaussian mixture model (GMM)

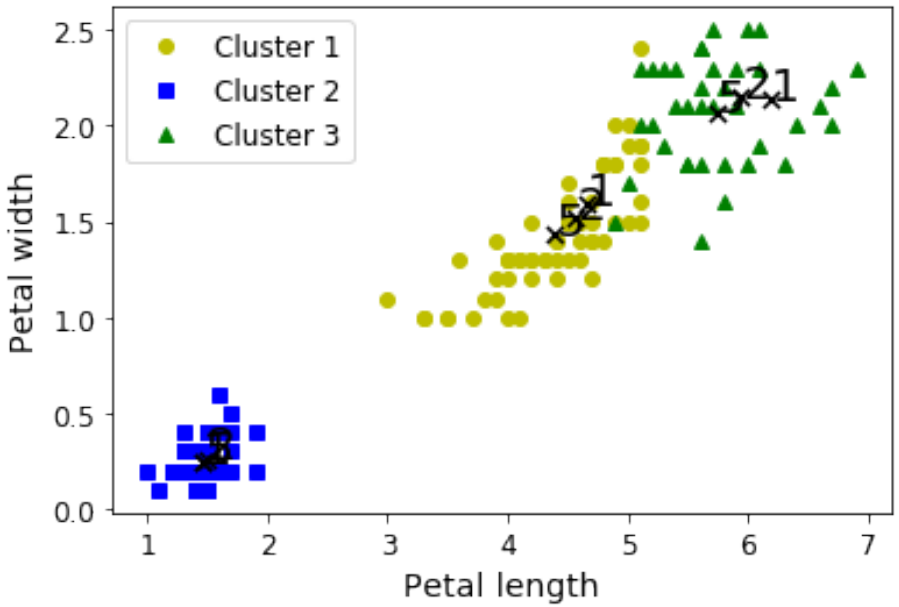
- good for multimodal distribution, and each blob is Gaussian-like
- K components:
  - a mean of μ_k and a covariance matrix of $\mathbf{C}_k$



k-means clustering

A special case of GMM. Use an iterative procedure

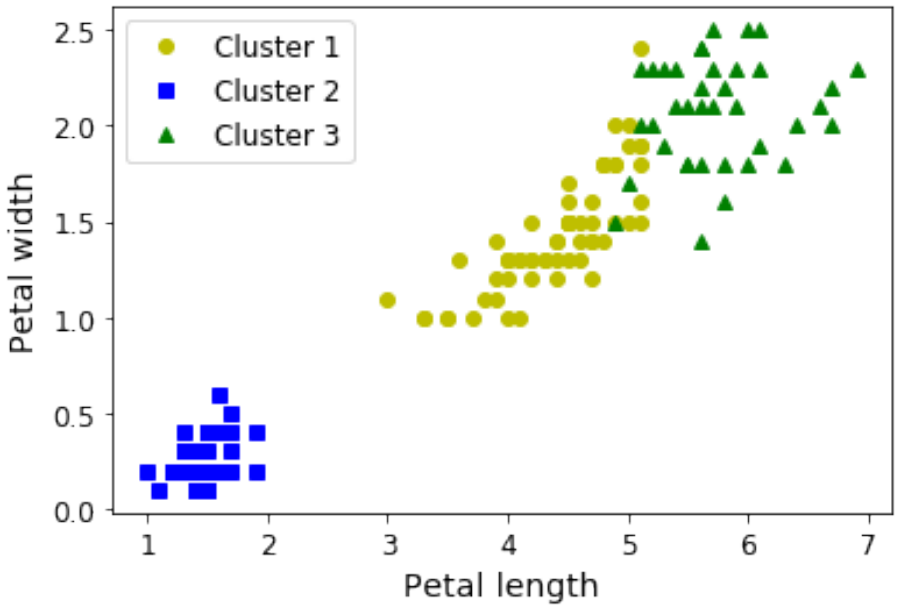
- first placing the centroids randomly
- data points are assigned to their nearest centroid
- Then label the instances, update the centroids, and repeat



k-means clustering

A special case of GMM. Use an iterative procedure

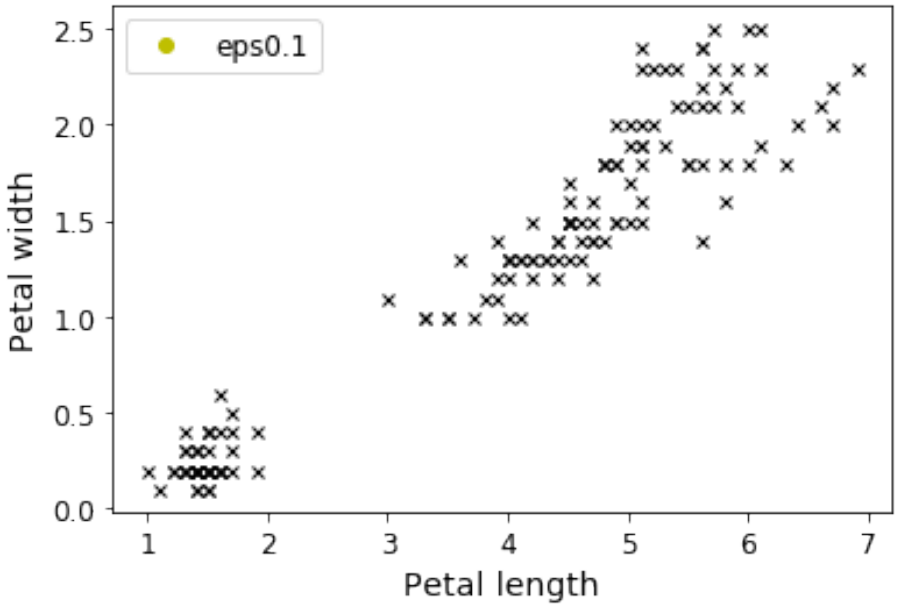
- first placing the centroids randomly
- data points are assigned to their nearest centroid
- Then label the instances, update the centroids, and repeat



DBSCAN

Density based algorithm relying on two parameters (eps and min_samples)

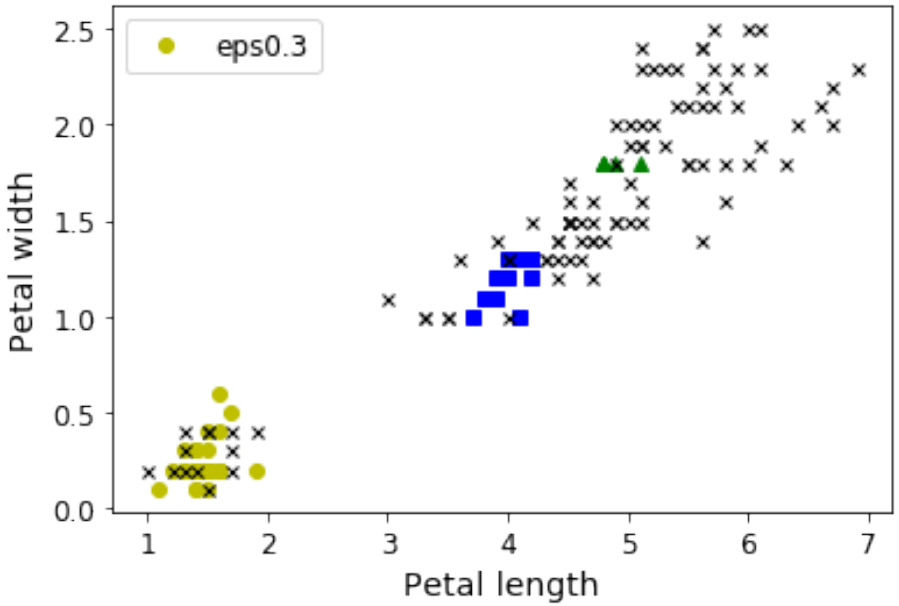
- Counts number of points with a small distance ϵ
- If the number of neighbors > min_samples, it is a core instance.
- All points in the neighborhood of a core instance belong to the same cluster.
- Anything that is not a core instance and does not have one in its neighborhood is noise.



DBSCAN

Density based algorithm relying on two parameters (eps and min_samples)

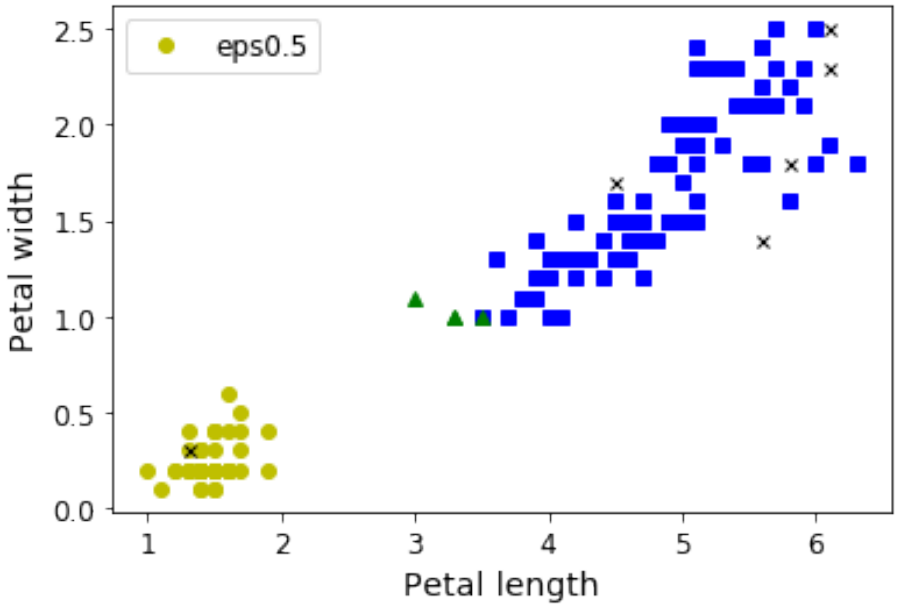
- Counts number of points with a small distance ϵ
- If the number of neighbors > min_samples, it is a core instance.
- All points in the neighborhood of a core instance belong to the same cluster.
- Anything that is not a core instance and does not have one in its neighborhood is noise.



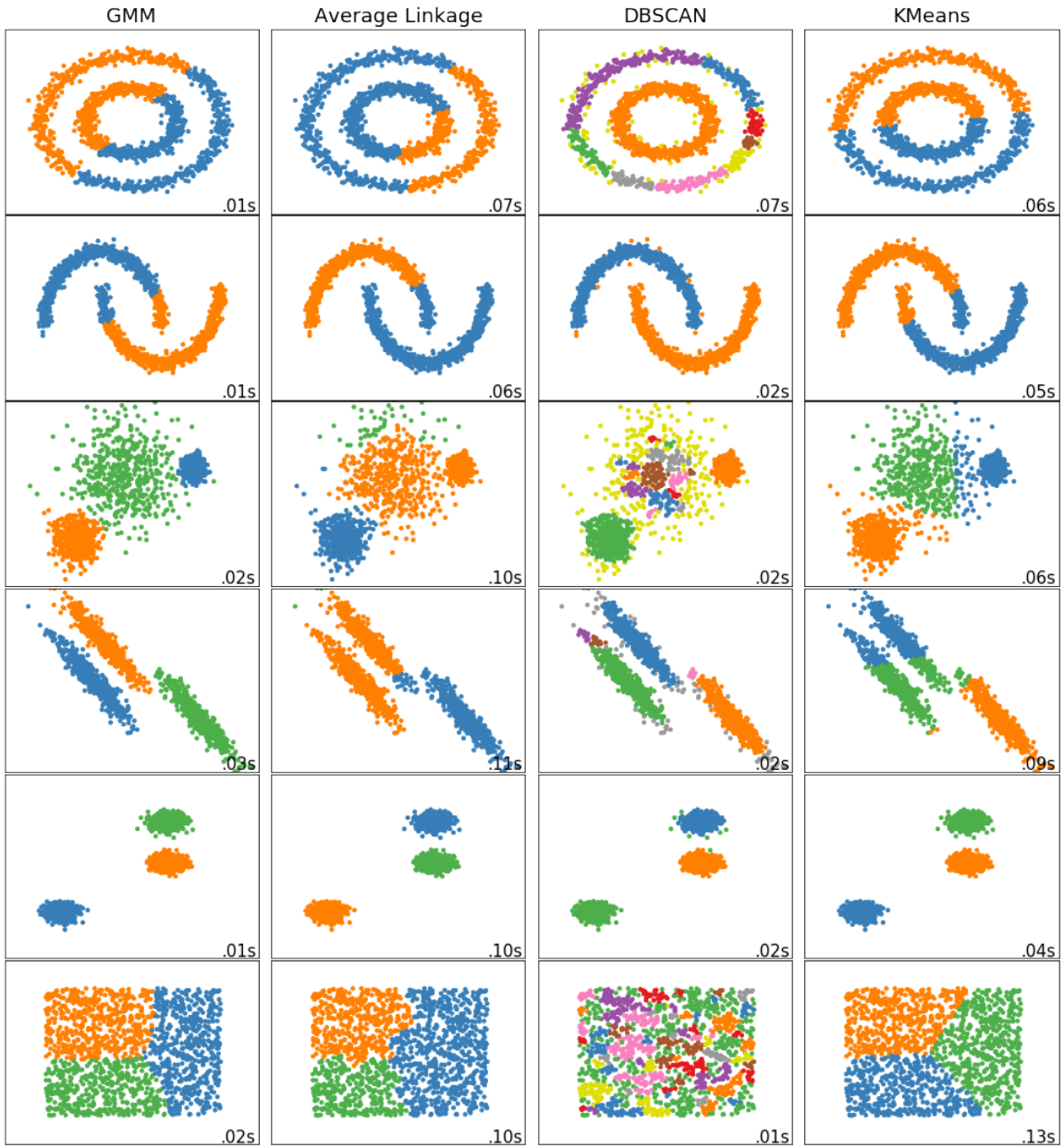
DBSCAN

Density based algorithm relying on two parameters (eps and min_samples)

- Counts number of points with a small distance ϵ
- If the number of neighbors > min_samples, it is a core instance.
- All points in the neighborhood of a core instance belong to the same cluster.
- Anything that is not a core instance and does not have one in its neighborhood is noise.



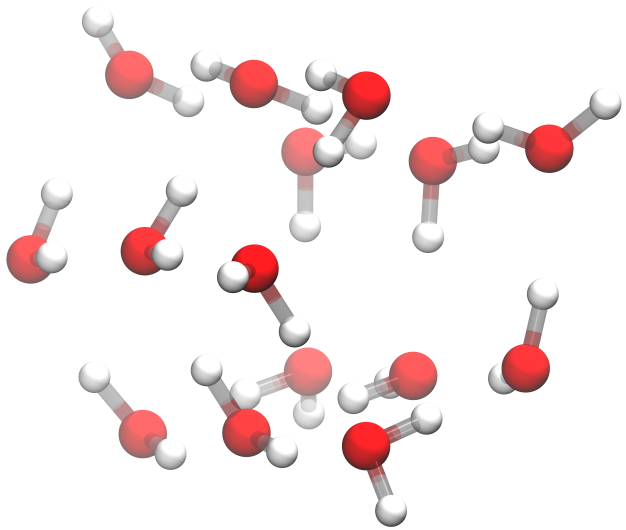
A comparison between clustering algorithms



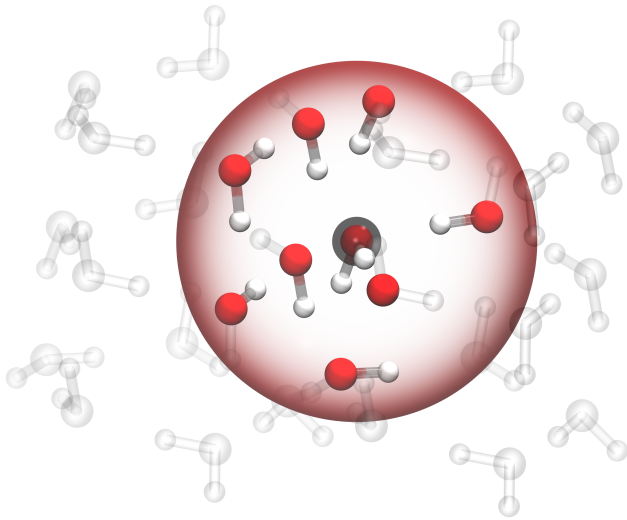
Outline

- More tricks of the trade
  - Sparsification
    - Single value decomposition (SVD)
    - CUR decomposition
    - Farthest Point Sampling (FPS)
  - Clustering
    - Gaussian mixture model (GMM)
    - K-means
    - DBSCAN
- Representing structual data

Representing atomistic environments



A



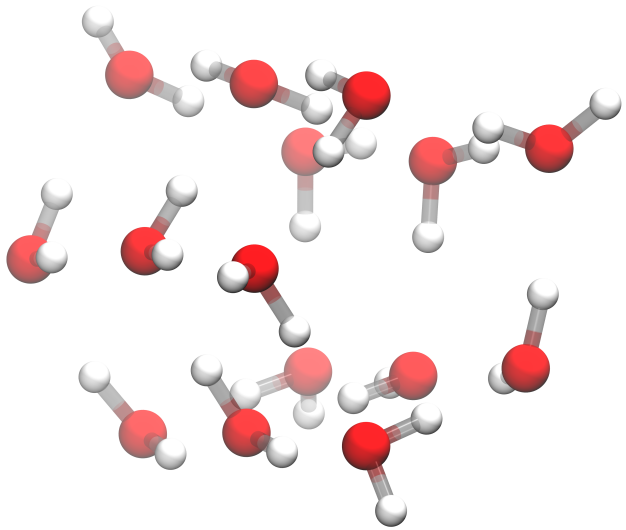
$\mathcal{X}_i$

The first step is to divide the system into a set of atomic environments.
So the task becomes representing atomic environments.

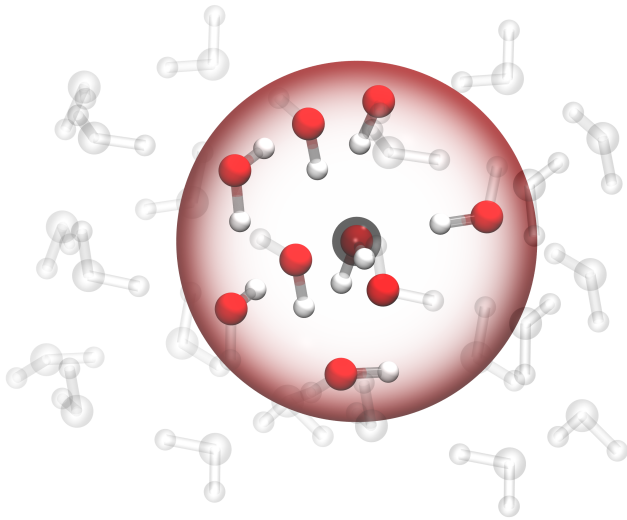
Popular representations (Invariant with respect to translation, rotation and permutation.):

- Smooth overlap of atomic positions (SOAP) [Bartók, Kondor & Csányi PRB 2013]
- Behler-Parrinello symmetry functions [Behler & Parrinello PRL 2008]

Representing atomistic environments



$$\Phi \leftarrow \{\Psi(\mathcal{X}_i)\}$$



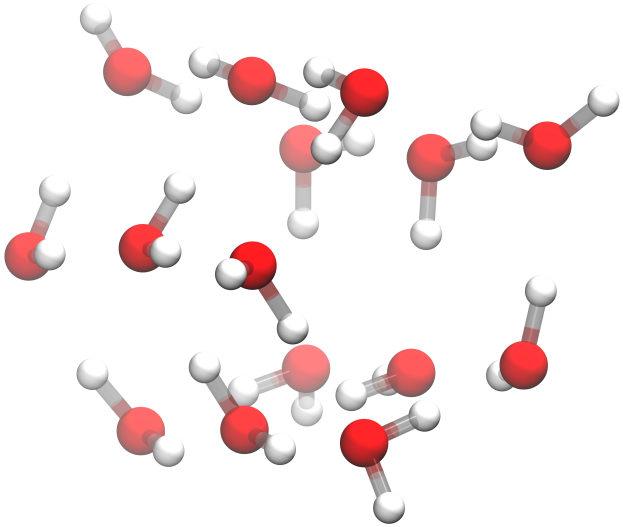
$$\Psi(\mathcal{X}_i)$$

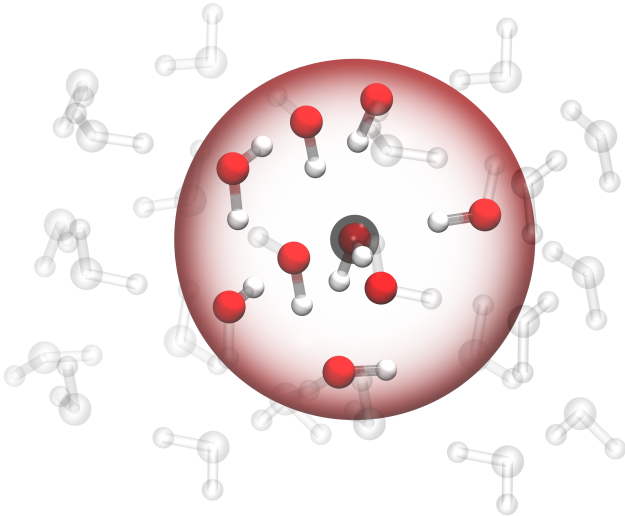
The first step is to divide the system into a set of atomic environments.
So the task becomes representing atomic environments.

Popular representations (Invariant with respect to translation, rotation and permutation.):

- Smooth overlap of atomic positions (SOAP) [Bartók, Kondor & Csányi PRB 2013]
- Behler-Parrinello symmetry functions [Behler & Parrinello PRL 2008]

Representing atomistic environments





$$O(A) = F(\Phi(A))$$

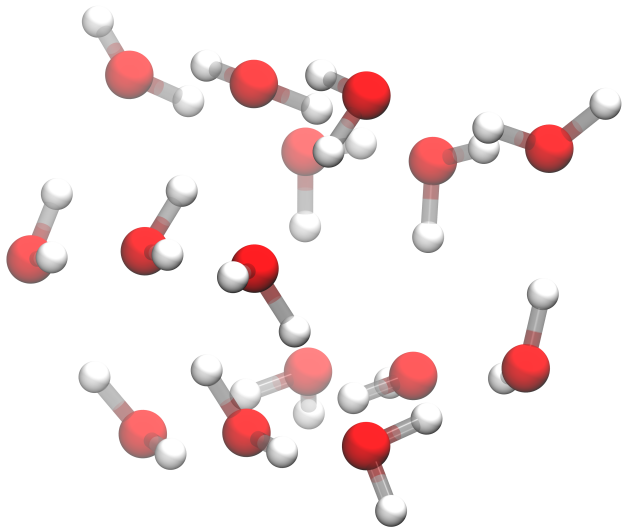
$$O_i = f(\Psi(\mathcal{X}_i))$$

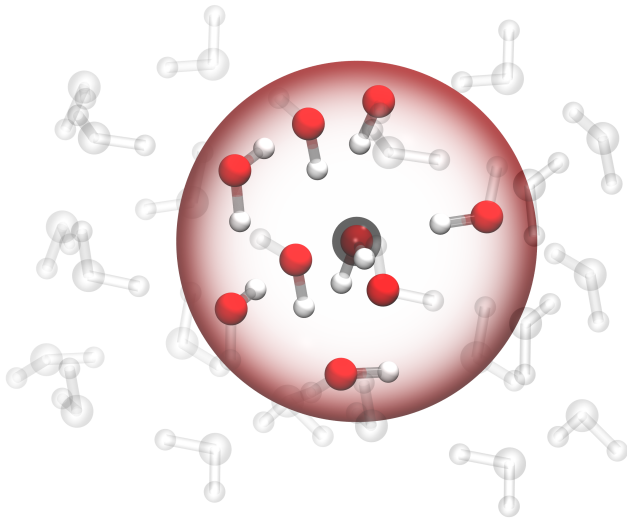
The first step is to divide the system into a set of atomic environments.
So the task becomes representing atomic environments.

Popular representations (Invariant with respect to translation, rotation and permutation.):

- Smooth overlap of atomic positions (SOAP) [Bartók, Kondor & Csányi PRB 2013]
- Behler-Parrinello symmetry functions [Behler & Parrinello PRL 2008]

Representing atomistic environments





$$E(A) = \sum E_i$$

$$E_i = e(\Psi(\mathcal{X}_i))$$

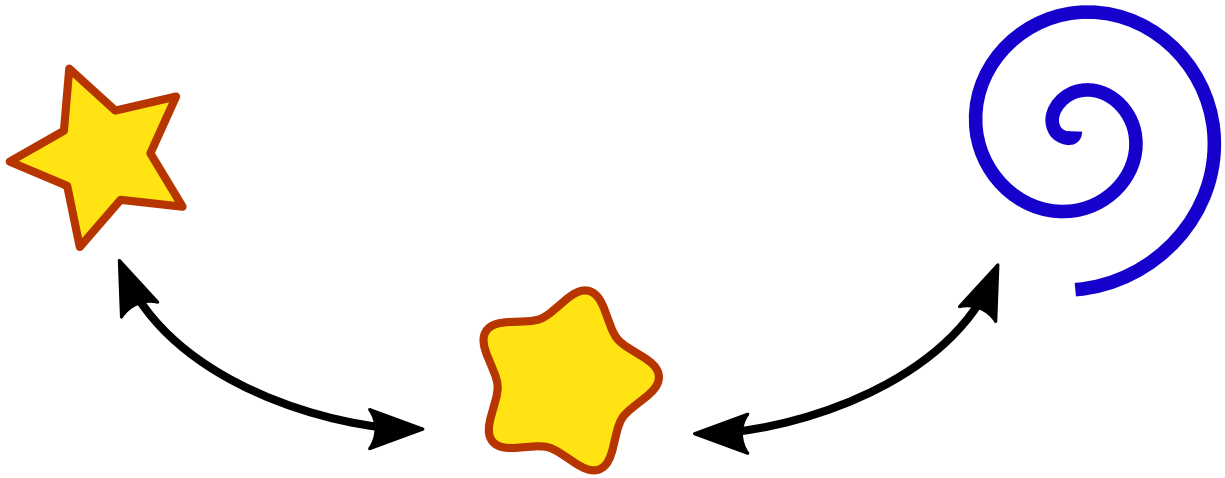
The first step is to divide the system into a set of atomic environments.
So the task becomes representing atomic environments.

Popular representations (Invariant with respect to translation, rotation and permutation.):

- Smooth overlap of atomic positions (SOAP) [Bartók, Kondor & Csányi PRB 2013]
- Behler-Parrinello symmetry functions [Behler & Parrinello PRL 2008]

Representing atomic environments

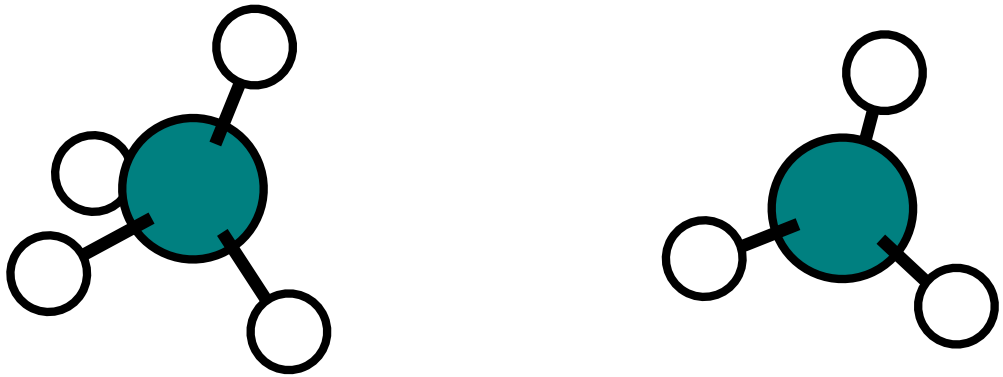
[Bartók, Kondor & Csányi PRB 2013]



$$K(\mathcal{X}, \mathcal{X}') \approx 1 \qquad K(\mathcal{X}, \mathcal{X}') \approx 0$$
$$D(\mathcal{X}, \mathcal{X}') \approx 0 \qquad D(\mathcal{X}, \mathcal{X}') \gg 0$$

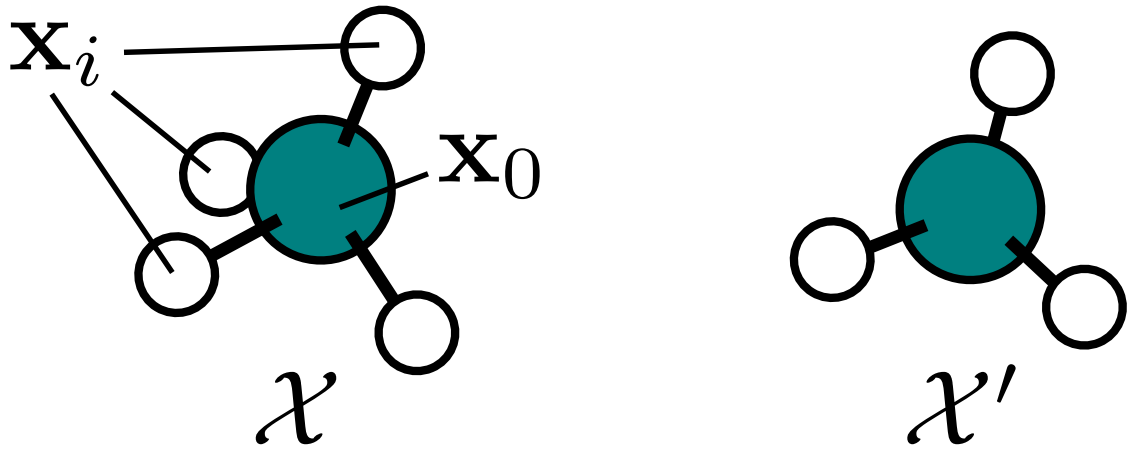
Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]



Representing atomic environments

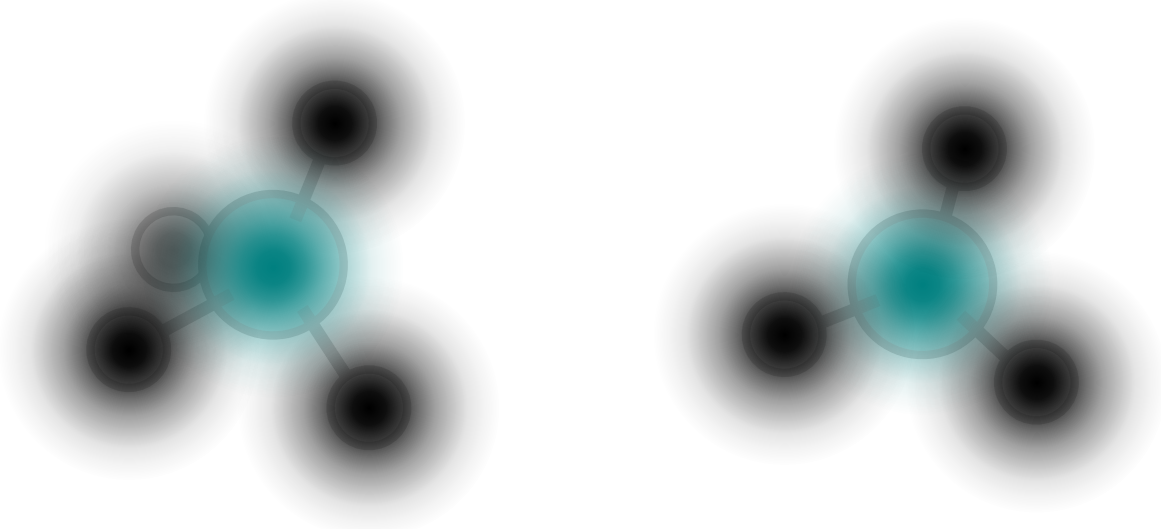
[Bartók, Kondor & Csányi PRB 2013]



$$\{\mathbf{x}_i - \mathbf{x}_0\} \leftrightarrow \mathcal{X}$$

Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]



$$\rho_\alpha(\mathbf{x}) = \sum_{i \in \alpha} g(\mathbf{x} - \mathbf{x}_i)$$

Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]

$$k(\mathcal{X}, \mathcal{X}') = \int \rho(\mathbf{x})\rho'(\mathbf{x})$$

Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]

$$k(\mathcal{X}, \mathcal{X}') = \int \mathrm{d}\hat{R} \left| \int \rho(\mathbf{x}) \rho'(\hat{R}\mathbf{x}) \right|^2$$

Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]

$$k(\mathcal{X}, \mathcal{X}') = \int \mathrm{d}\hat{R} \left| \int \rho(\mathbf{x}) \rho'(\hat{R}\mathbf{x}) \right|^2$$

Representing atomic environments

[Bartók, Kondor & Csányi PRB 2013]

$$\kappa(\mathcal{X}, \mathcal{X}') = \int \mathrm{d}\hat{R} |\rho(\mathbf{x})\rho'(\hat{R}\mathbf{x})|^2$$

$$\rho(\mathbf{x}) = \sum_{nlm} c_{nlm} g_n(|r|) Y_{lm}(\hat{r})$$

$$k_{nn'l}(\mathcal{X}) = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{nlm})^* c_{n'lm}$$

The list of vector $\{k_{nn'l}(\mathcal{X})\}$ is the descriptor of the atomic environment $\mathcal{X}$.

Similarity measurement between structures

- The kernel matrix $\{K\}$ records the similarity measurement for each pair of structures in the data set.
- The kernel function $K(A, B)$ for structure A and B is

$$K(A, B) = \Phi(A)^{\mathsf{T}} \Phi(B) = \sum_{i=1}^{M} \phi_i(A) \phi_i(B)$$

- Global features are constructed from local features by taking the average:

$$\Phi(A) = \frac{1}{N_A} \sum_{n=1}^{N_A} \Psi(\mathcal{X}_n^A)$$

- We use SOAP vectors as local features. [Bartók et al. PRB 2013]
  We use DScribe Python libray [Himanen et al. arxiv 2019]

$$k_{nn'l}(\mathcal{X}) = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{nlm})^* c_{n'lm}$$

Visualizing and processing the data set

- Build two-dimensional map using dimensionality reduction
  - (kernel) PCA, Multidimensional scaling (MDS), Isometric mapping (ISOMAP), locally linear embedding (LLE)
- Sparsity the train set
  - farthest point sampling, CUR decomposition
- Clustering
  - Gaussian mixture model (GMM), K-means, DBSCAN....
- Regression
  - ridge regression, neural networks

Questions?

Thank you for your attention!