Psych 253
Advanced Statistical Modeling

Clustering and Mixture Models

Daniel Yamins
Wu Tsai Neurosciences Institute
Departments of Psychology and Computer Science
Stanford Artificial Intelligence Laboratory
Stanford University

Stanford Neuro AI Lab

Unsupervised data analysis

Supervised models:

$$data\_slice\_2 = F_{[params]}(data\_slice\_1)$$

Unsupervised data analysis

Supervised models:

data_slice_2 = F[params](data_slice_1)

examples:

regression

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \sum_i w_i x_i + b$$

categorization

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \text{sign}\left[\sum_i w_i x_i + b\right]$$

Unsupervised data analysis

Supervised models:

data_slice_2 = F[params](data_slice_1)

*determine **params** (like **w**, **b**) from pair of linked data tensors*

examples:

regression

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \sum_i w_i x_i + b$$

categorization

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \text{sign}\left[\sum_i w_i x_i + b\right]$$

Unsupervised data analysis

Supervised models:

data_slice_2 = F[params](data_slice_1)

*determine **params** (like **w**, **b**) from pair of linked data tensors*

Parameters found which minimize some pre-specified **loss function**

examples:

regression:

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \sum_i w_i x_i + b$$

categorization:

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \text{sign}\left[\sum_i w_i x_i + b\right]$$

regression: e.g. mean squared error

categorization: e.g. hinge loss

Unsupervised data analysis

Supervised models:

data_slice_2 = $F_{[params]}$(data_slice_1)

*determine **params** (like **w**, **b**) from pair of linked data tensors*

Parameters found which minimize some pre-specified **loss function**

examples:

regression:

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \sum_i w_i x_i + b$$

categorization:

$$F_{\mathbf{w},\mathbf{b}}(\vec{x}) = \text{sign}\left[\sum_i w_i x_i + b\right]$$

regression: e.g. mean squared error

categorization: e.g. hinge loss

NB: since parameter-finding procedure involves some examples of data_slice_2, evaluation has to be done via **cross-validation on held-out data**

Unsupervised data analysis

Unsupervised models:

$$\text{exploratory idea} = F_{[\text{params}]}(\text{data\_slice\_1})$$

doesn't involve pre-determined output

Unsupervised data analysis

Unsupervised models:

$$\text{exploratory idea} = F_{[\text{params}]}(\text{data\_slice\_1})$$

doesn't involve pre-determined output

Parameters found which minimize some pre-specified **loss function??**

Unsupervised data analysis

Unsupervised models:

$$\text{exploratory idea} = F_{[\text{params}]}(\text{data\_slice\_1})$$

doesn't involve pre-determined output

Parameters found which minimize some pre-specified **loss function??**

How to validate given there are no labels??

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

k-means clustering

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

k-means clustering

Hierarchical k-means

Affinity Propagation

DBScan

Mixture methods:

Gaussian Mixture Models (GMMs)

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.mixture

Mixture models: **`sklearn.mixture`**

Gaussian Mixture Models

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

Unsupervised data analysis

No "labels", just "data".

unsupervised equivalent of classification

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

~unsupervised equivalent of regression (kinda)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Unsupervised data analysis

No "labels", just "data".

today

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

future class (but after Linear Algebra review)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

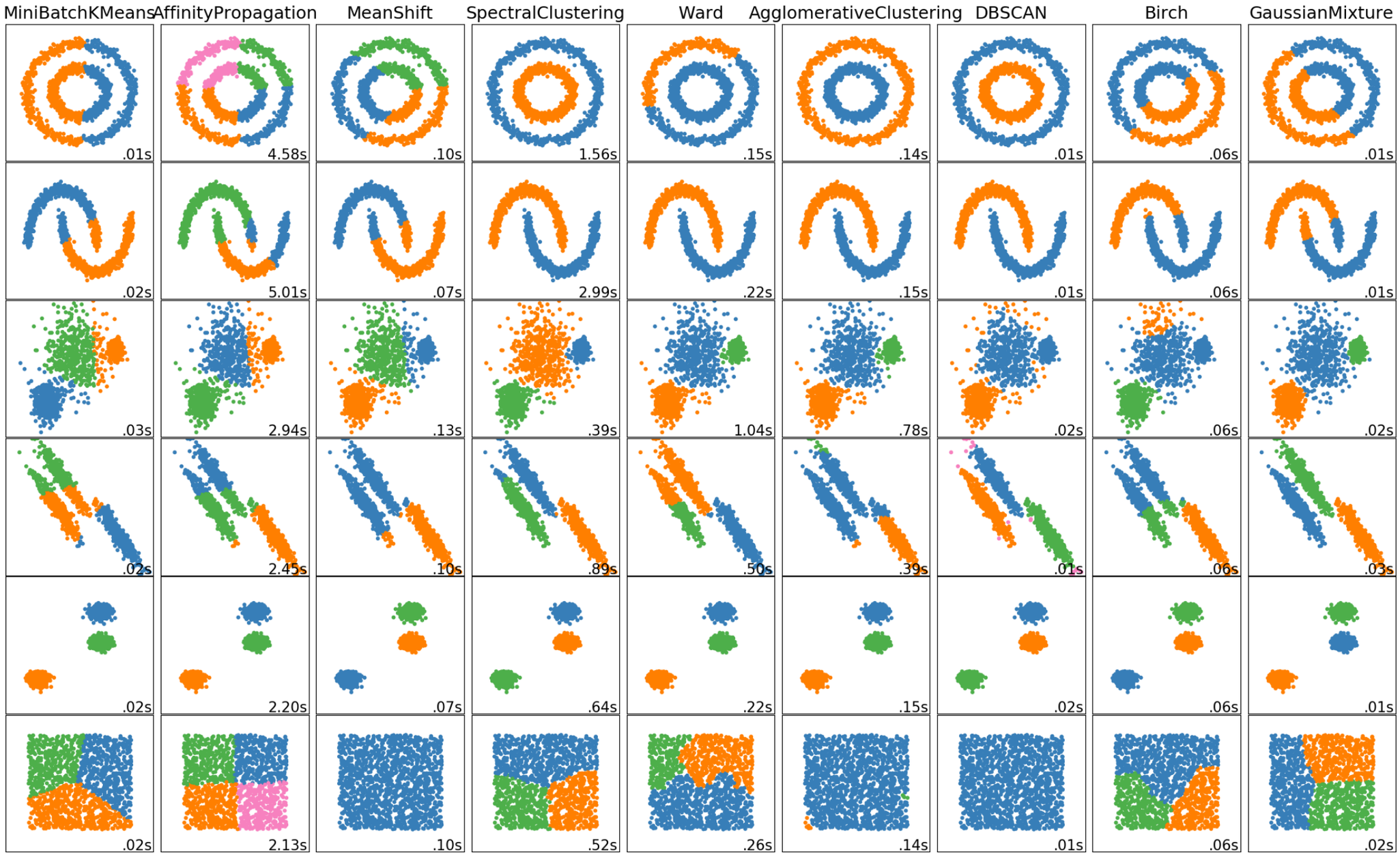
Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Clustering

Clustering: assign datapoint to natural groups based on the way the data is laid out.

Some of the many methods of clustering



K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised

loss function

K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

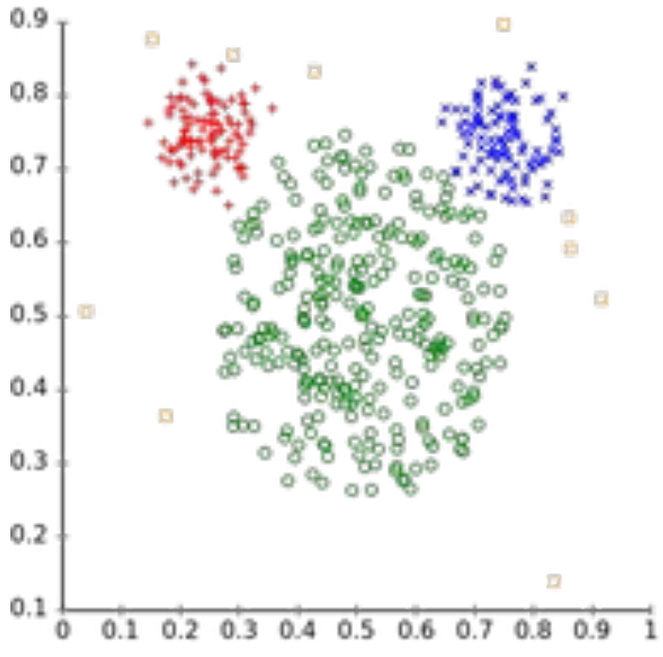
C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

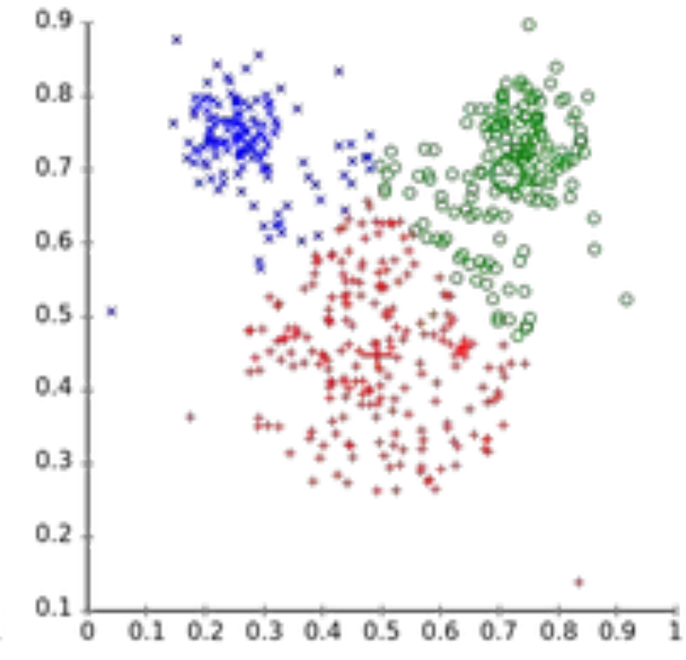
$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised
loss function

true data generated from three gaussian blobs



k-means clustering result



K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised loss function

```
def kmeans(data, k):
    choose initial means
```

K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised loss function

```
def kmeans(data, k):
    choose initial means
    repeat until stable:
        assign points to clusters by distance
        update means
```

K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised loss function

```
def kmeans(data, k):
    choose initial means
    repeat until stable:
        assign points to clusters by distance
        update means
    return assignment
```

K-means Clustering

k = number of clusters

C = partition of the data into clusters

assignment of each point to a cluster

e.g. vector of values in {1, …, k}, one for each datapoint

C assignment chosen ('learned') to minimize within-cluster sum-of-squares:

= maximize between-cluster sum-of-squares:

$$\text{argmin}_C \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

unsupervised loss function

```
def kmeans(data, k):
    choose initial means
    repeat until stable:
        assign points to clusters by distance
        update means
    return assignment
```

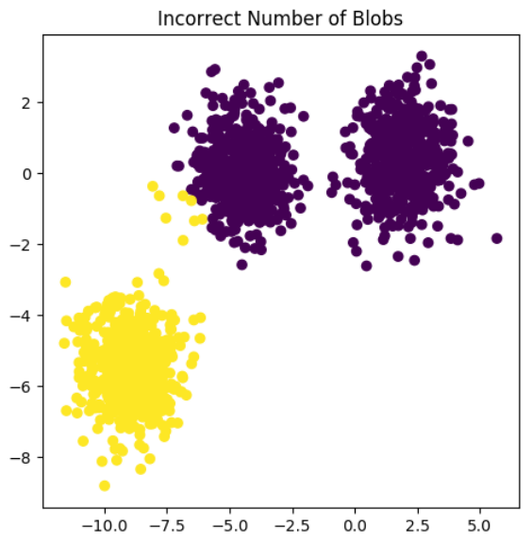
"Lloyds algorithm"

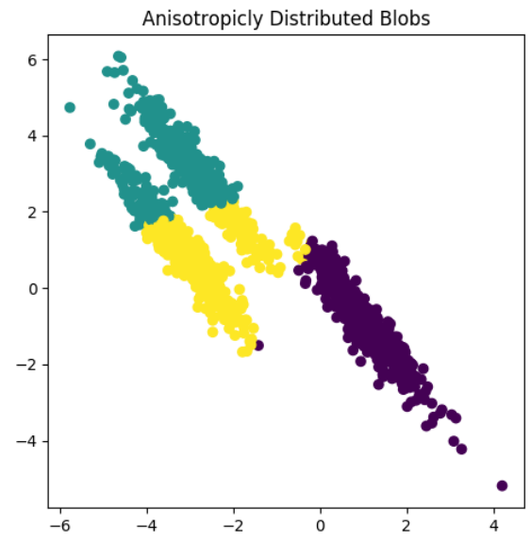
Clustering

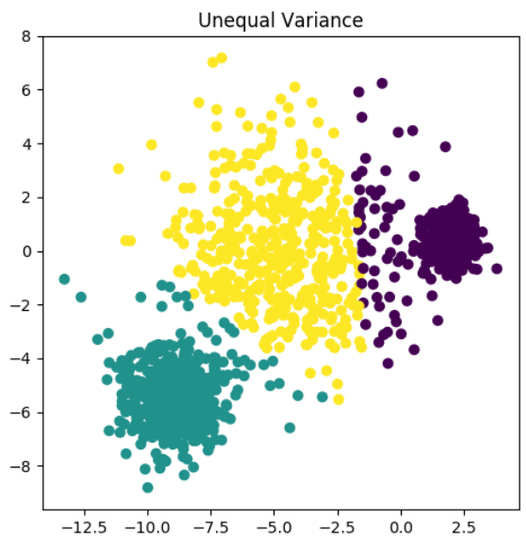
[IPYNB: K-means clustering]

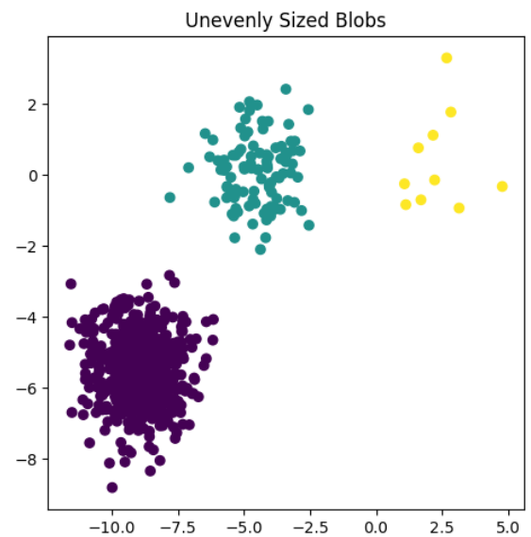
Clustering

many failure cases:



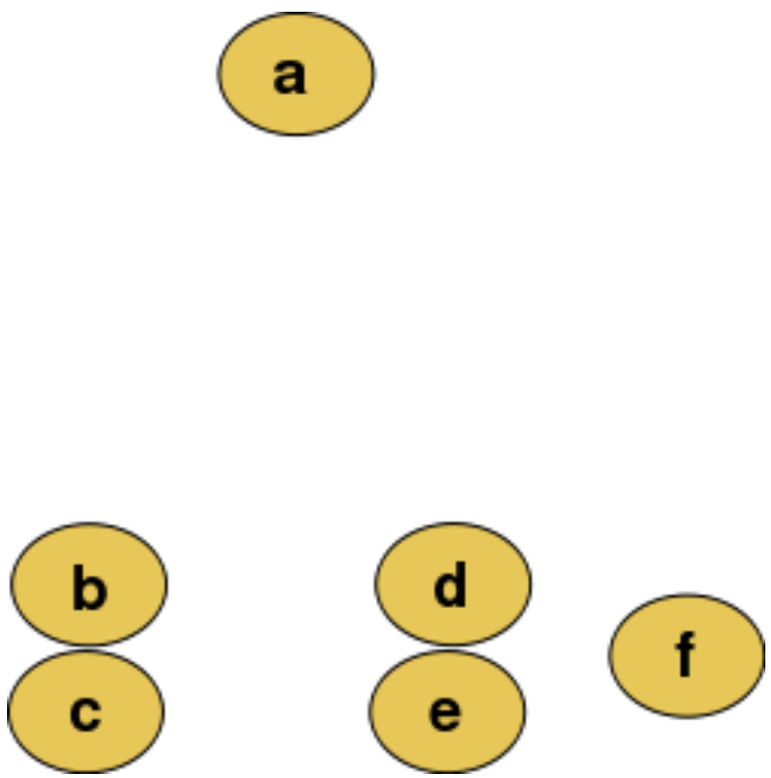






Clustering

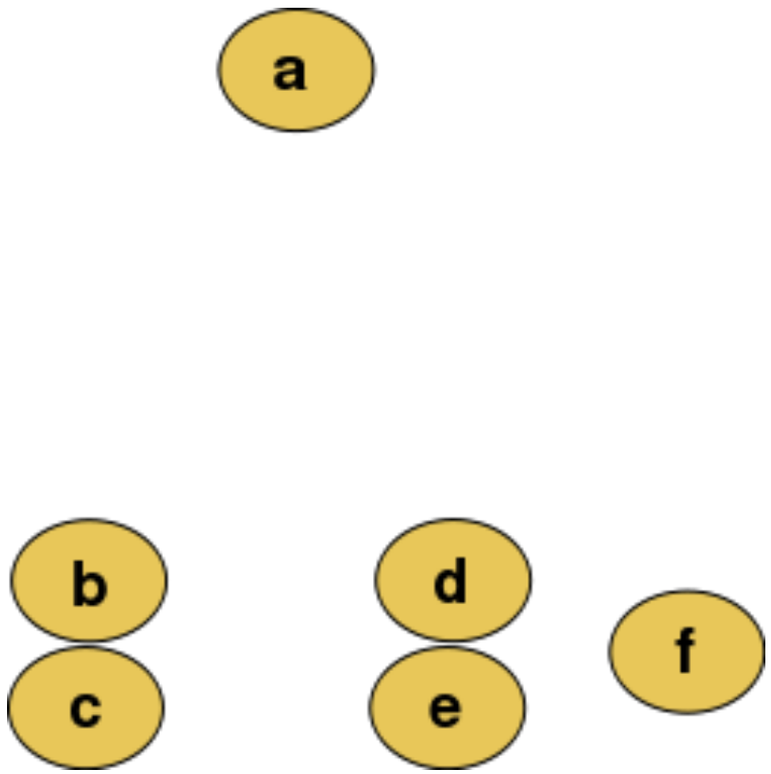
agglomerative clustering:



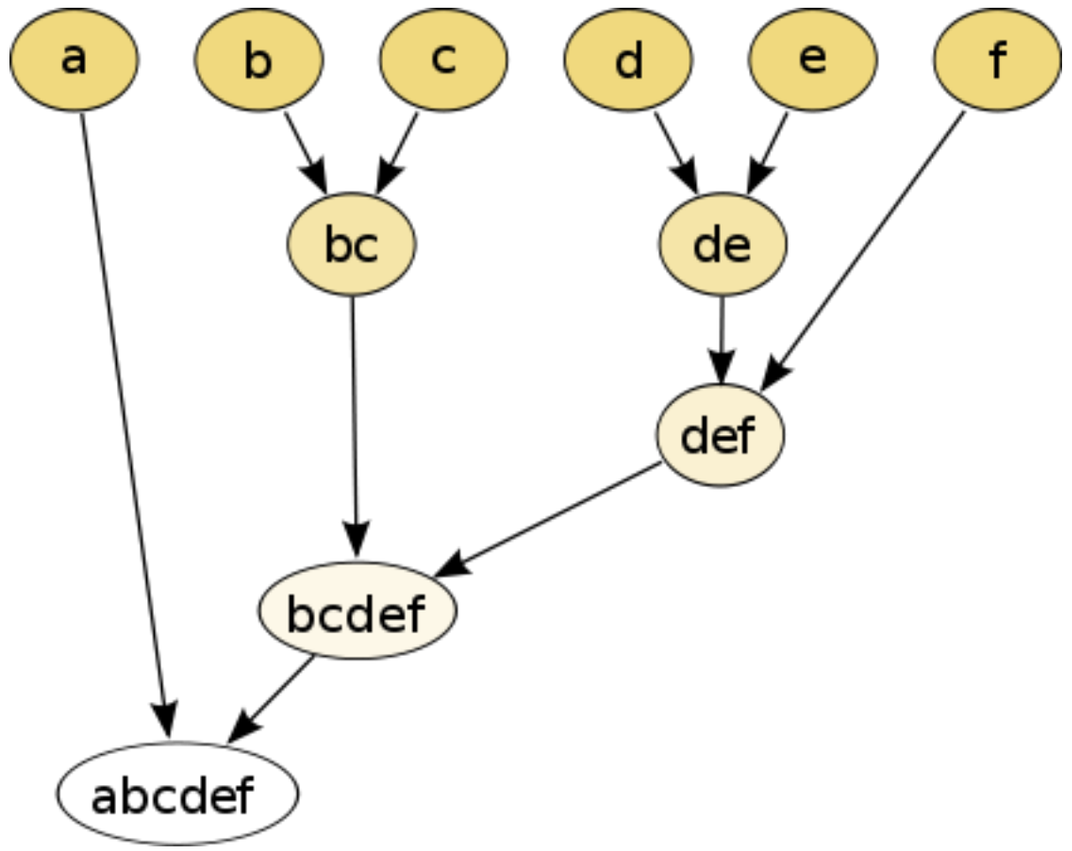
data

Clustering

agglomerative clustering:



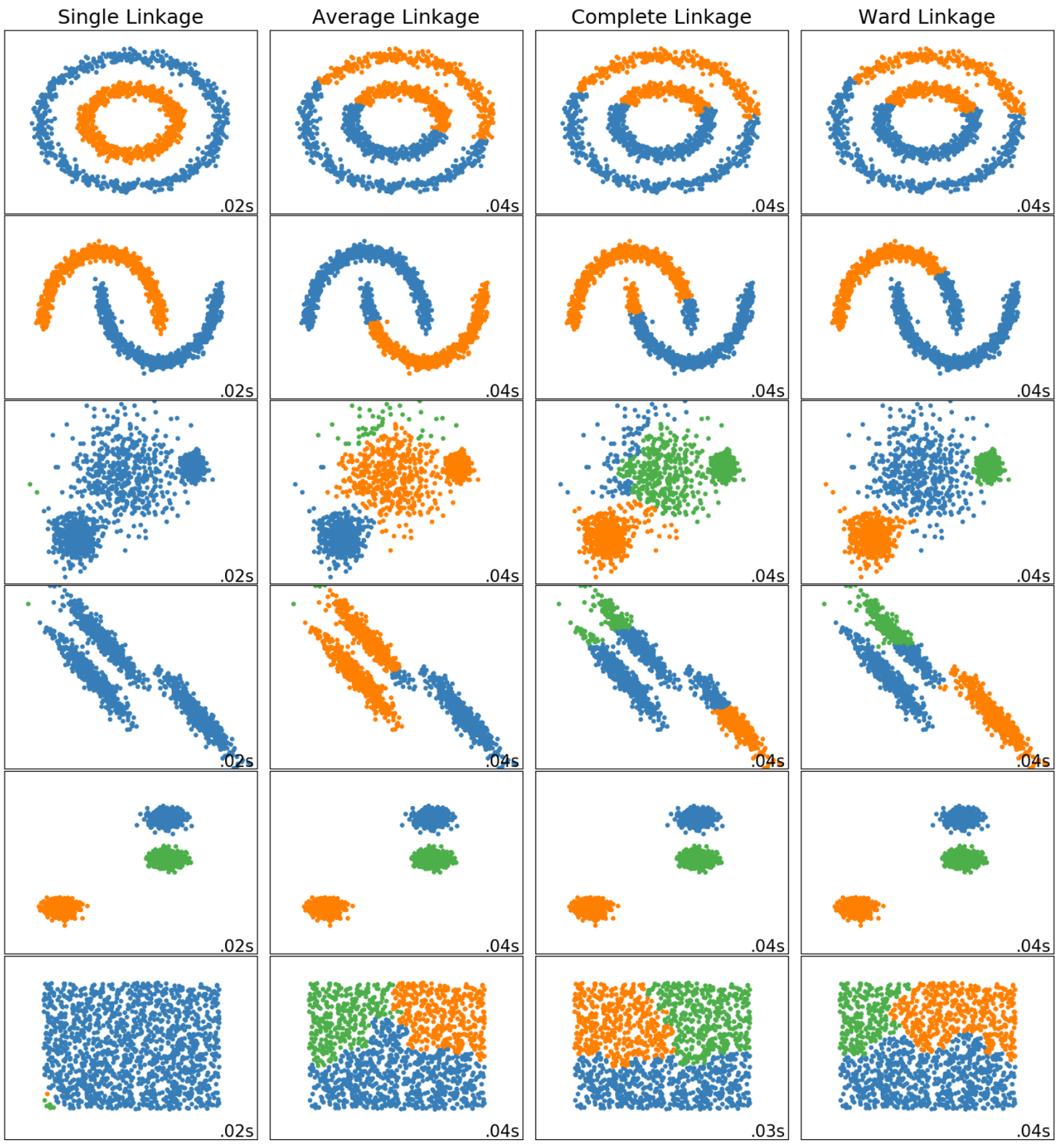
data



cluster hierarchical

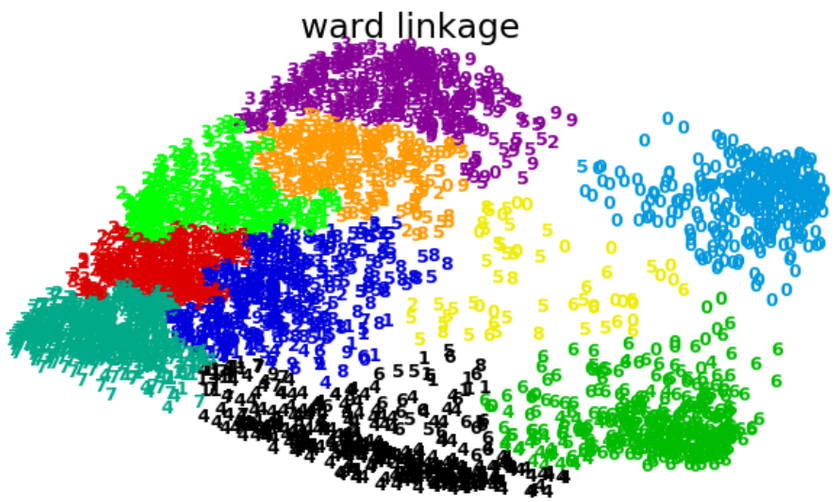
Clustering

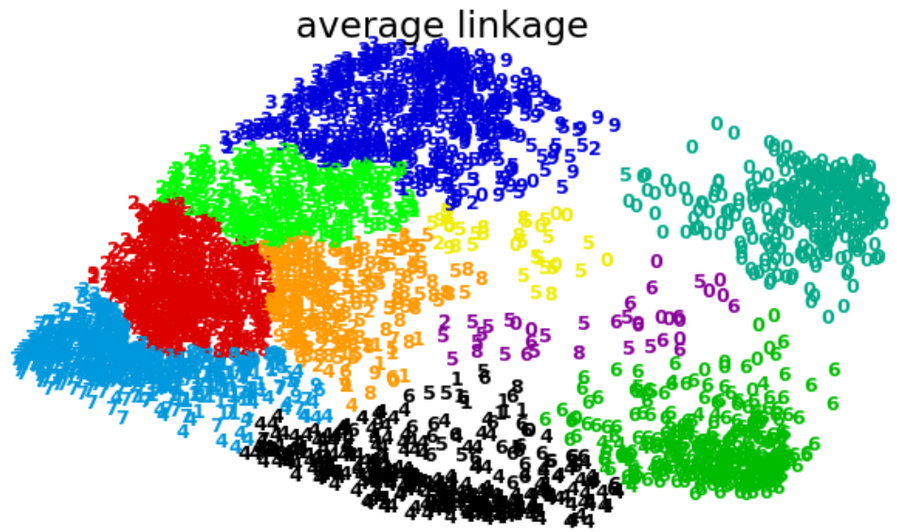
agglomerative clustering:

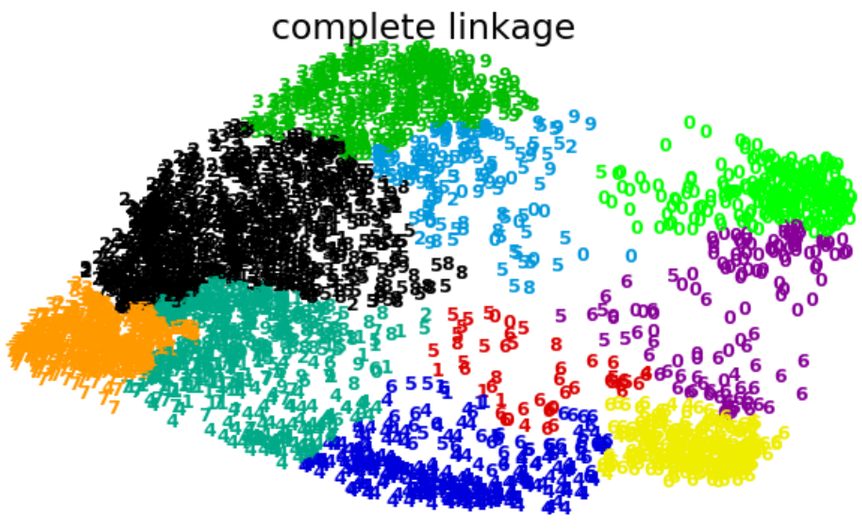


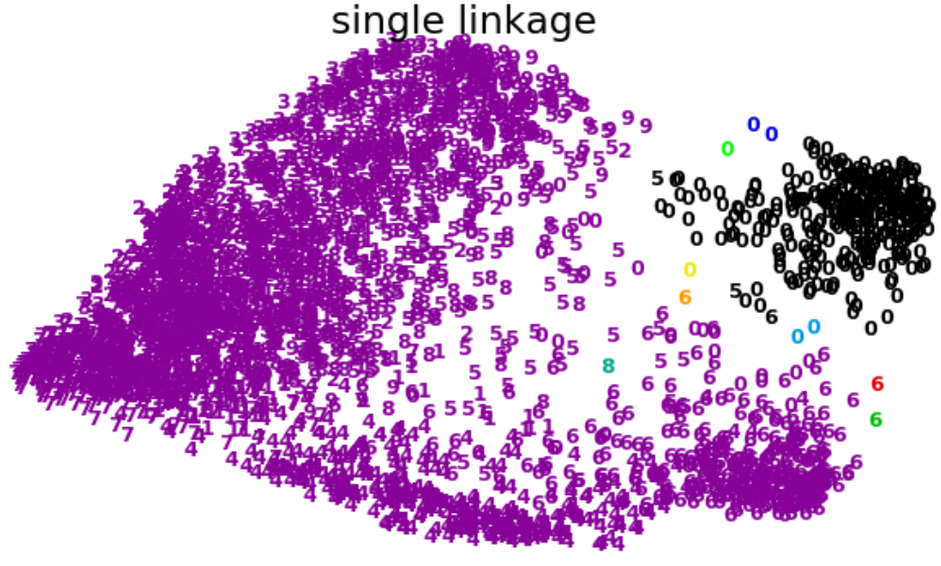
Clustering

agglomerative clustering:



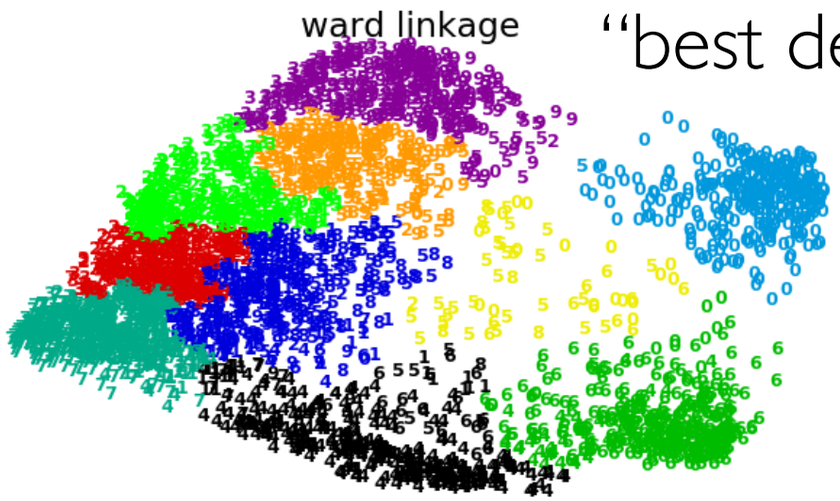




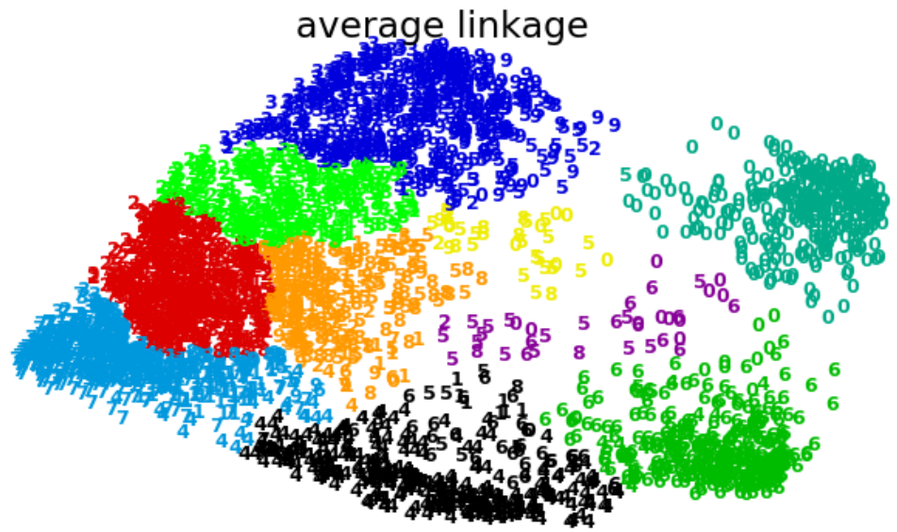


Clustering

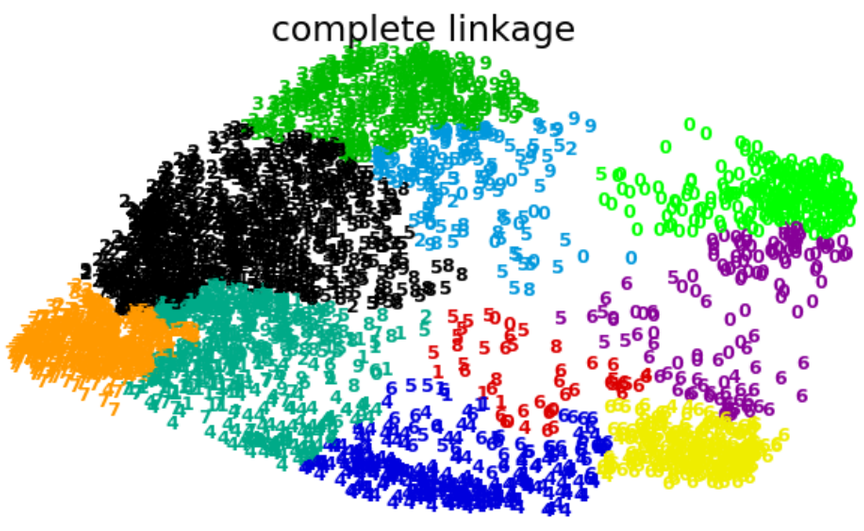
agglomerative clustering:

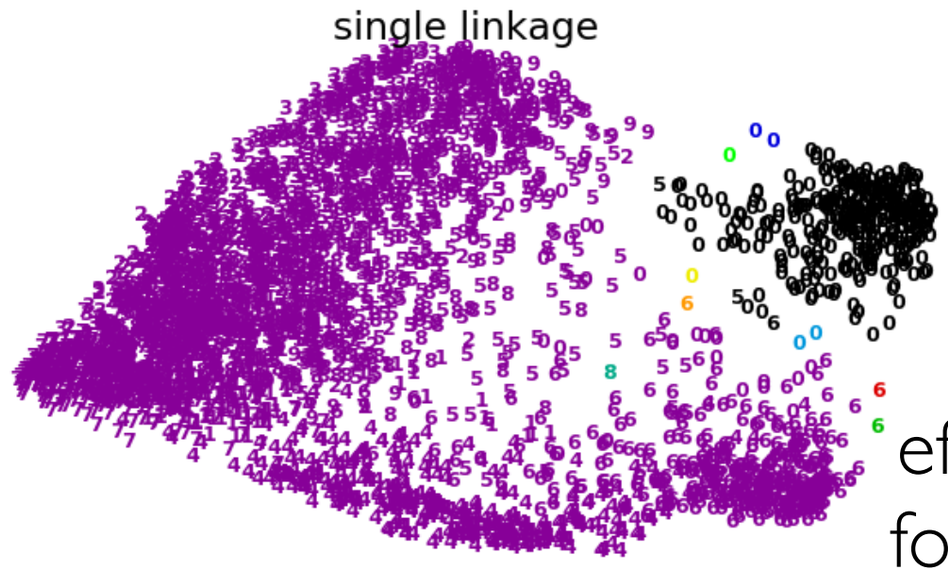


"best default"



good for non-euclidean distances

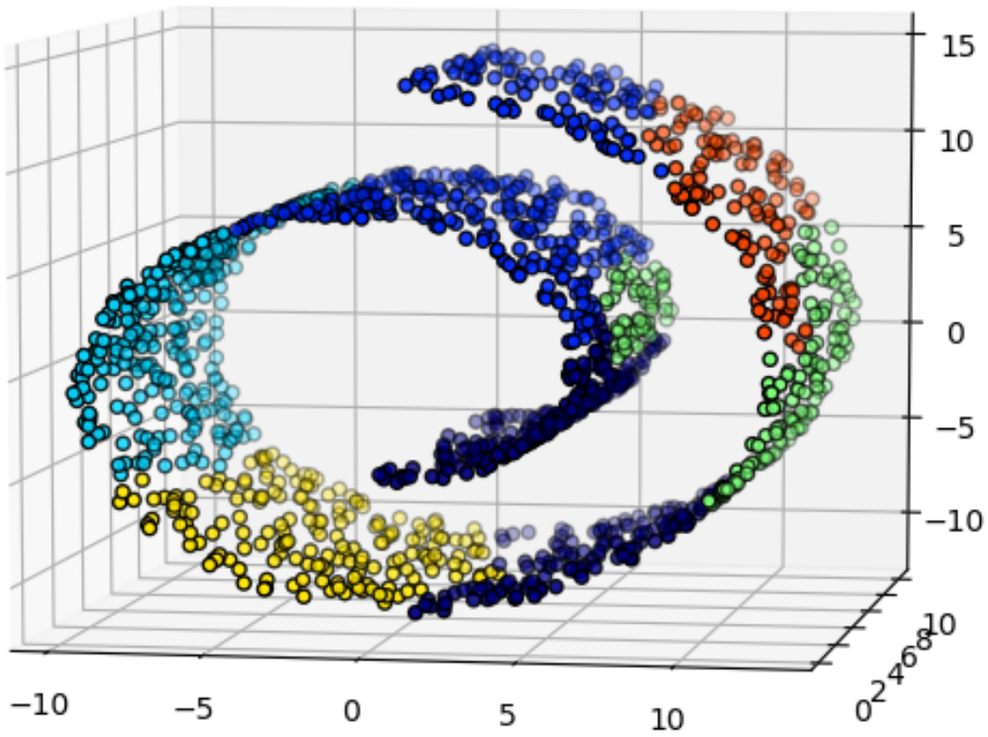


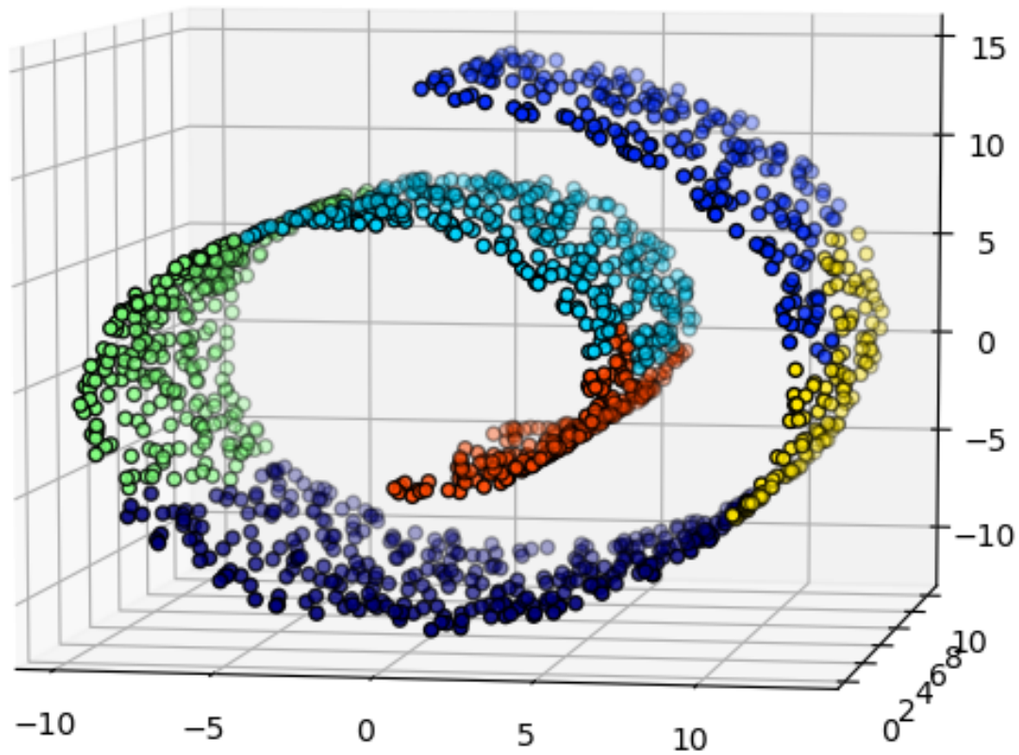


efficient for large datasets

Clustering

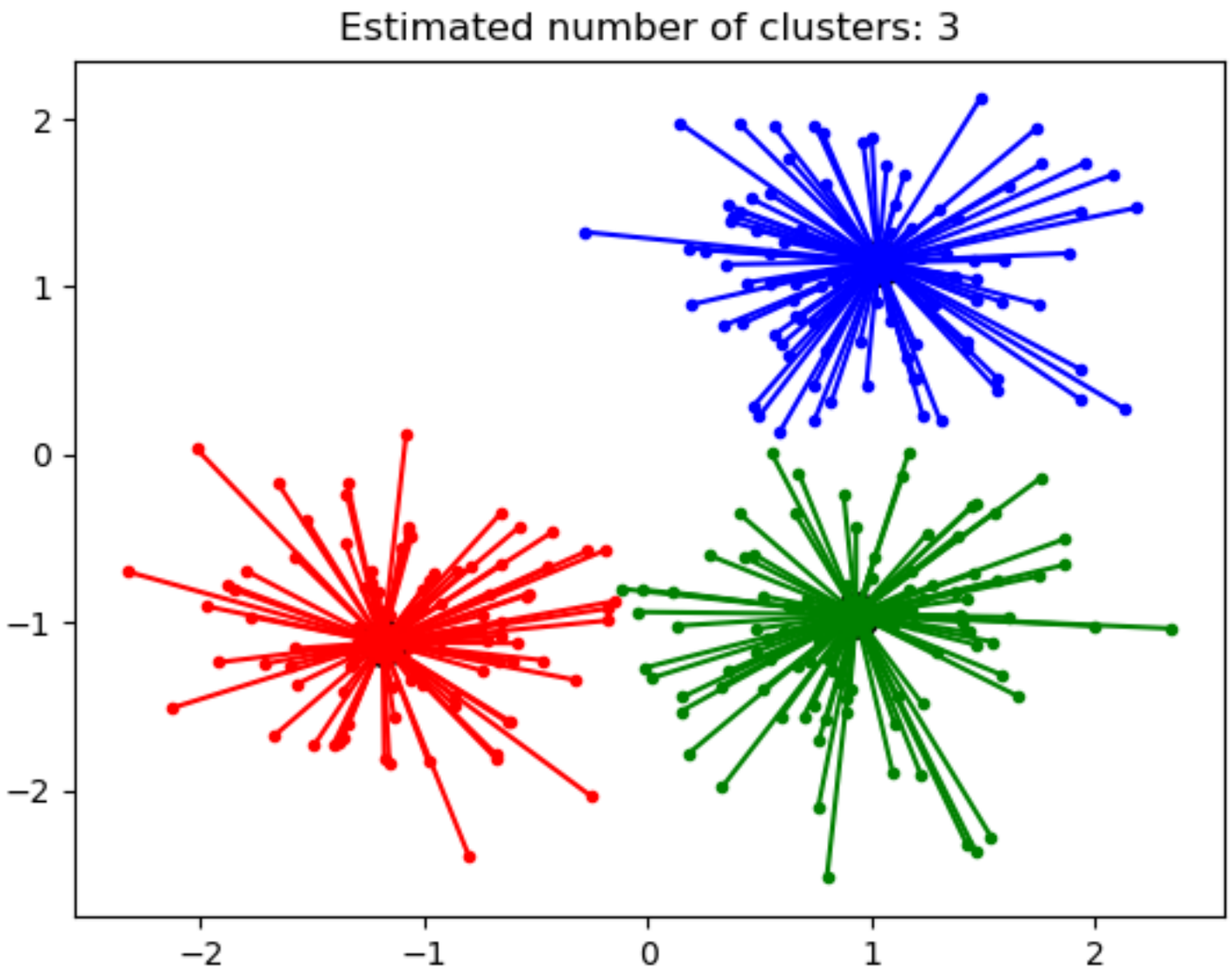
agglomerative clustering:



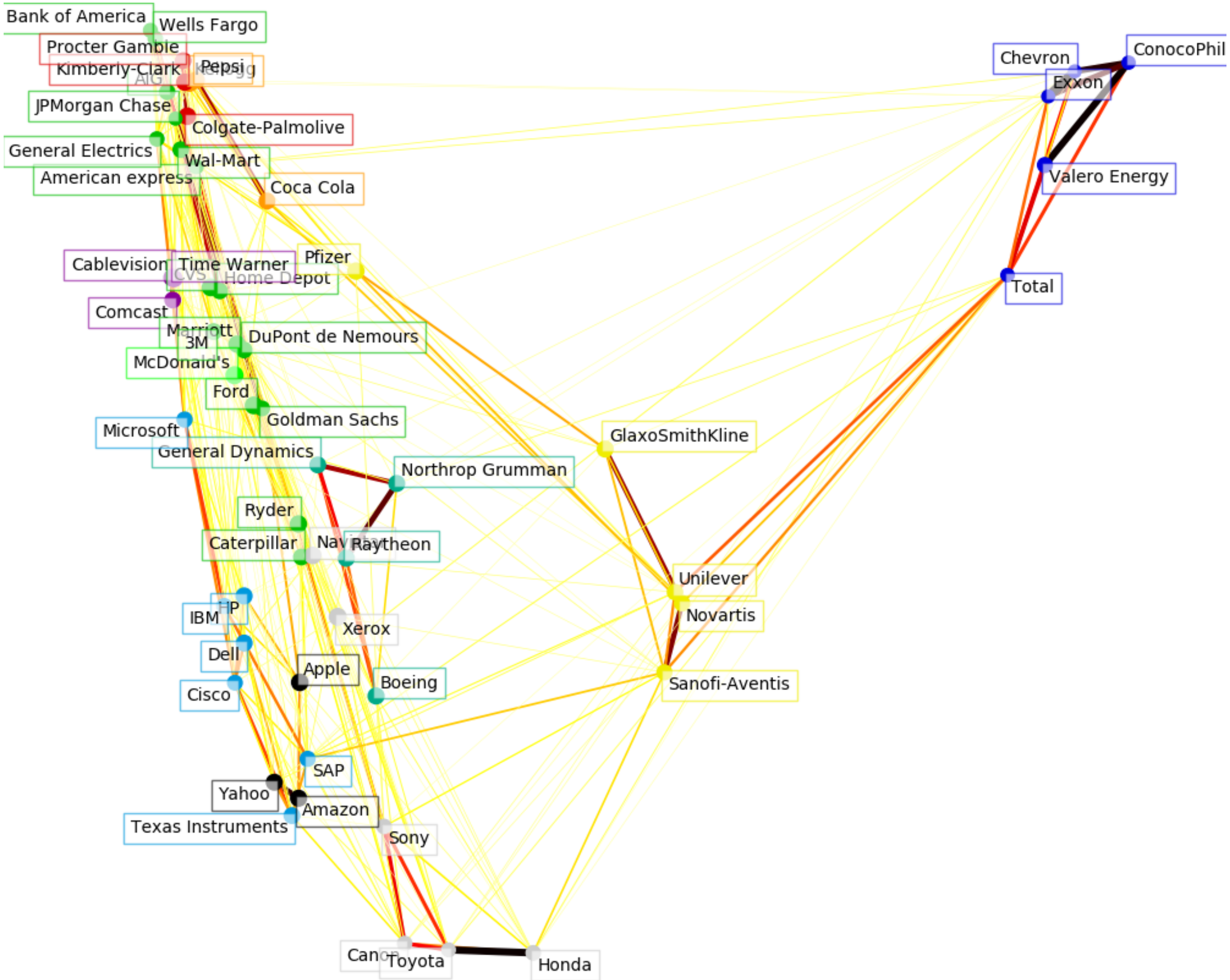


Clustering

affinity propagation:



self-selects numbers of clusters



https://scikit-learn.org/stable/auto_examples/applications/plot_stock_market.html

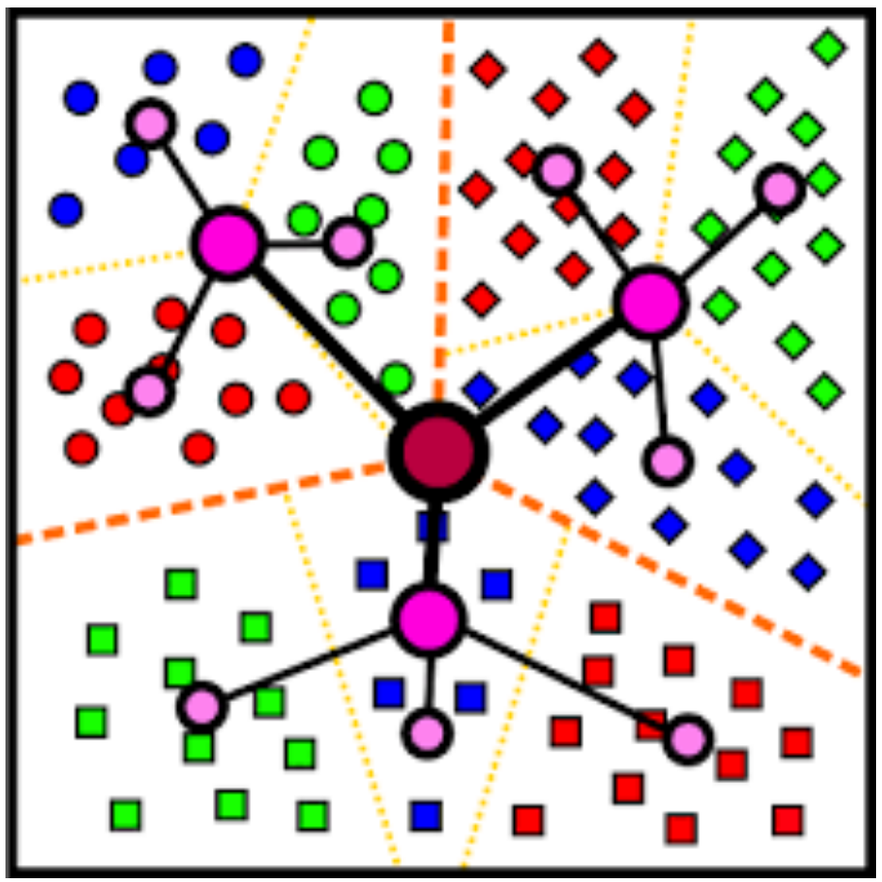
K-means Clustering

[IPYNB: Other clustering methods]

Clustering

Hierarchical K-means

1. Run K-means on original data

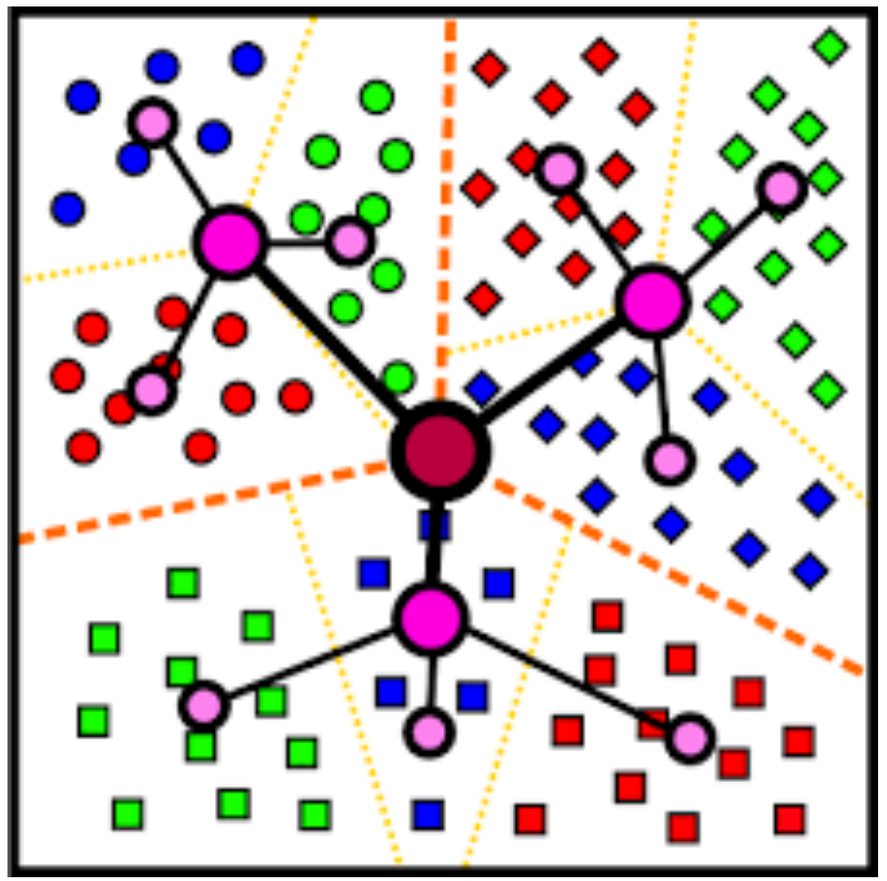


Clustering

Hierarchical K-means

1. Run K-means on original data

2. for each resulting cluster, recursively run K-means on points in each cluster

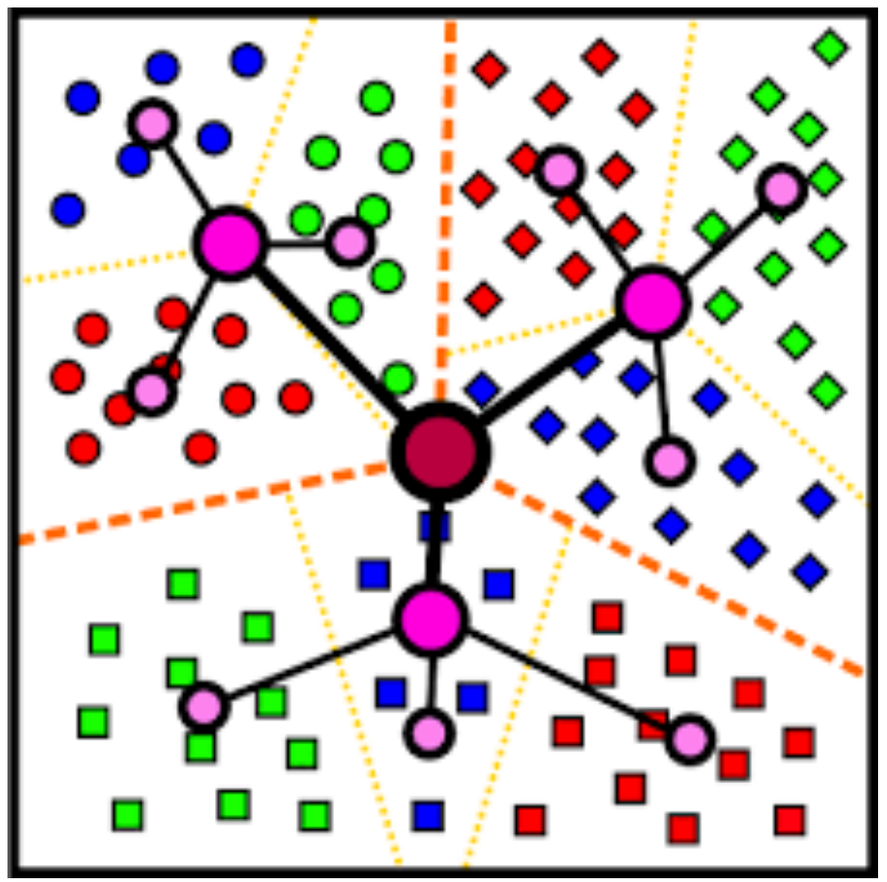


Clustering

Hierarchical K-means

1. Run K-means on original data

2. for each resulting cluster, recursively run K-means on points in each cluster



$$\sum_{i=0}^{L} k^i$$ parameters to learn with supervision

L = number of layers
k = average branching factor

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

Unsupervised data analysis

No "labels", just "data".

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

Clustering: **`sklearn.cluster`**

- k-means clustering
- Hierarchical k-means
- Affinity Propagation
- DBScan

Mixture methods:

- Gaussian Mixture Models (GMMs)

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

Dimension reduction: **`sklearn.decomposition`**

- Principal Components Analysis (PCA)
- Factor Analysis
- Multi-Dimensional Scaling (MDS)
- Independent Components Analysis (ICA)
- Non-negative Matrix Factorization (NMF)

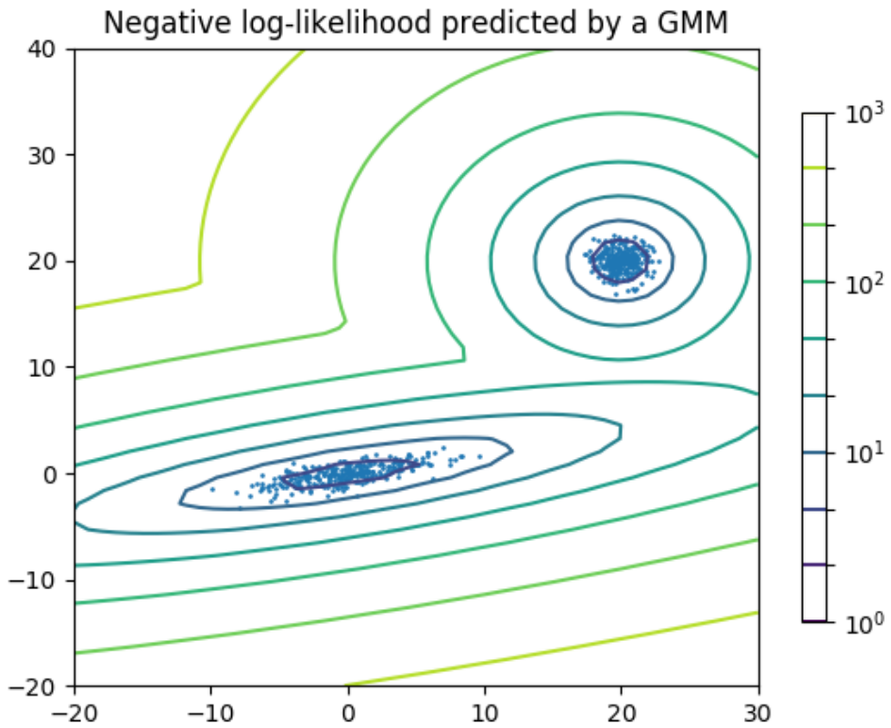
Mixture Models

Mixture Models are a statistical relaxation of clustering

$$x_i = \sum_{i=1}^{k} p_i g_i(x)$$

$$\sum_i p_i = 1$$

$$g_i \sim N(\mu_i, \sigma_i^2)$$



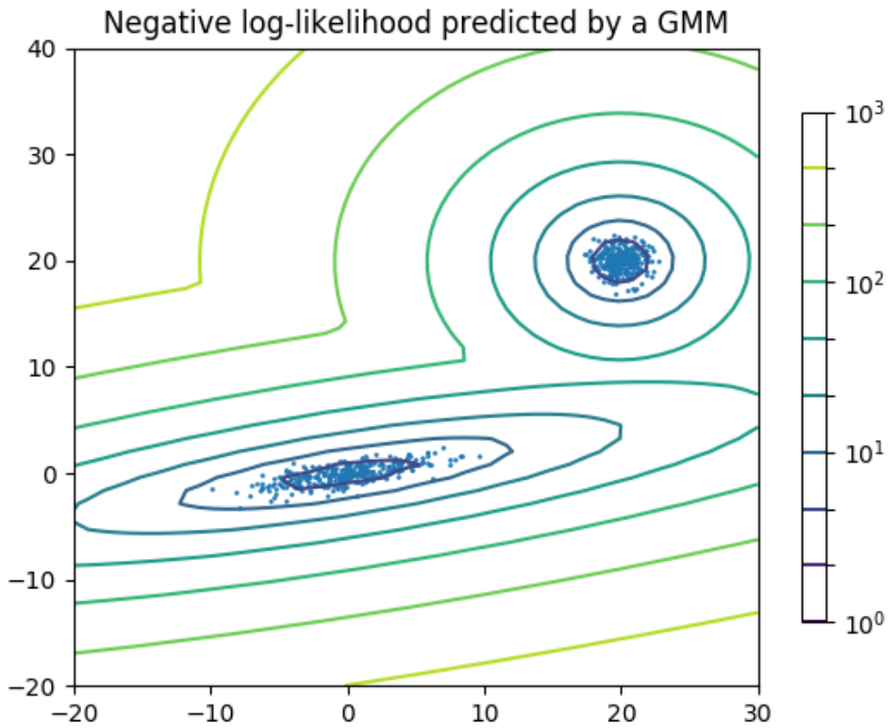
Mixture Models

Mixture Models are a statistical relaxation of clustering

$$x_i = \sum_{i=1}^{k} p_i g_i(x)$$

$$\sum_i p_i = 1$$

$$g_i \sim N(\mu_i, \sigma_i^2)$$



Q: How to estimate the means and the variances?

A: Beyond the scope of this lecture ("Expectation Maximization")

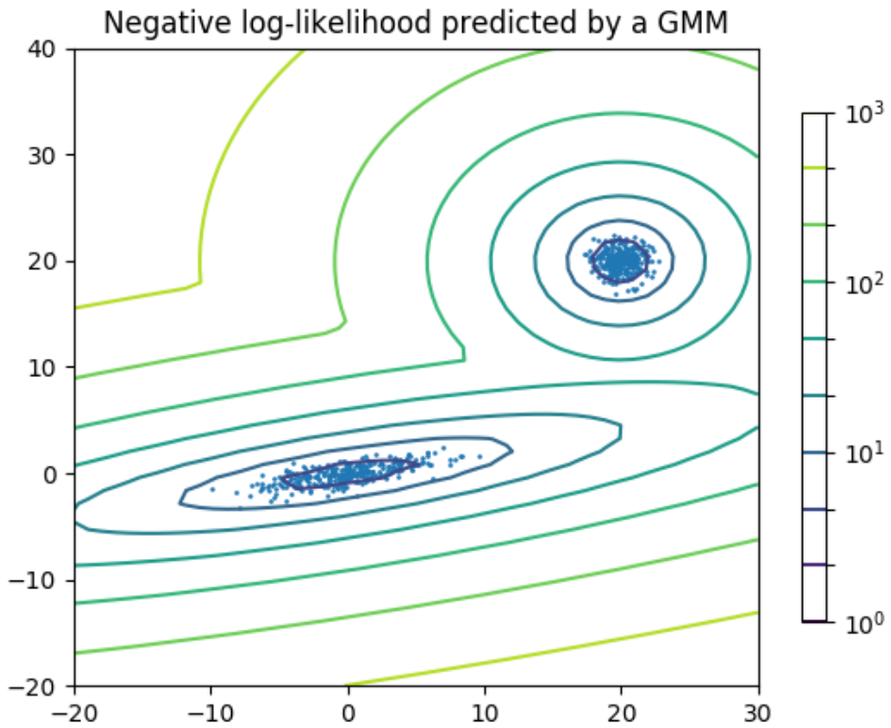
Mixture Models

Mixture Models are a statistical relaxation of clustering

$$x_i = \sum_{i=1}^{k} p_i g_i(x)$$

$$\sum_i p_i = 1$$

$$g_i \sim N(\mu_i, \sigma_i^2)$$



Q: How to estimate the means and the variances?

A: Beyond the scope of this course ("Expectation Maximization")

—> Or you can do stochastic gradient descent, of course

https://www.tensorflow.org/api_docs/python/tf/contrib/factorization

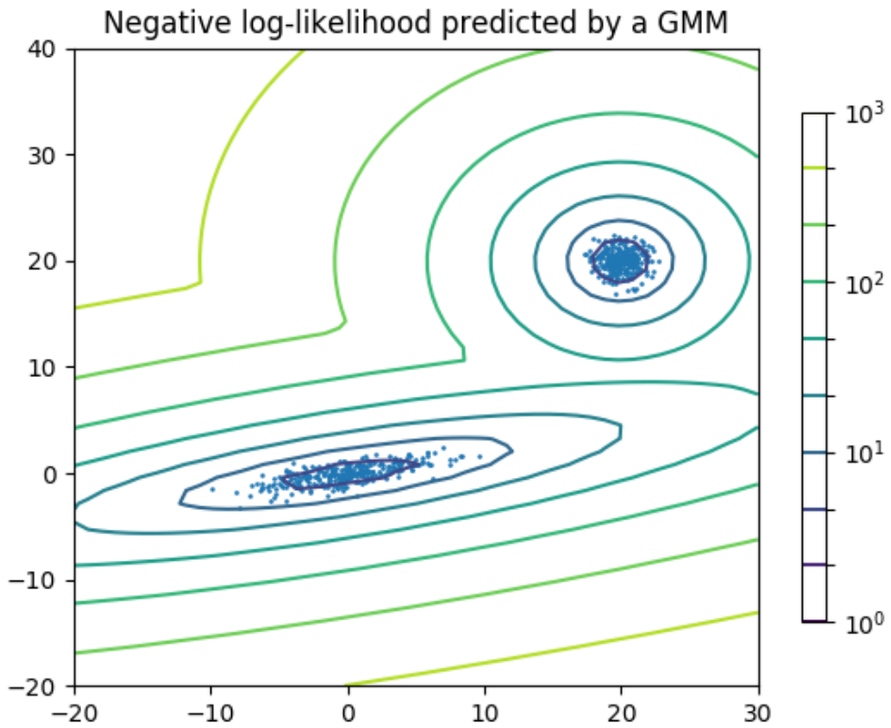
Mixture Models

Mixture Models are a statistical relaxation of clustering

$$x_i = \sum_{i=1}^{k} p_i g_i(x)$$

$$\sum_i p_i = 1$$

$$g_i \sim N(\mu_i, \sigma_i^2)$$



Can say: initialize parameters via K-means

Q: How to estimate the means and the variances?

A: Beyond the scope of this course ("Expectation Maximization")

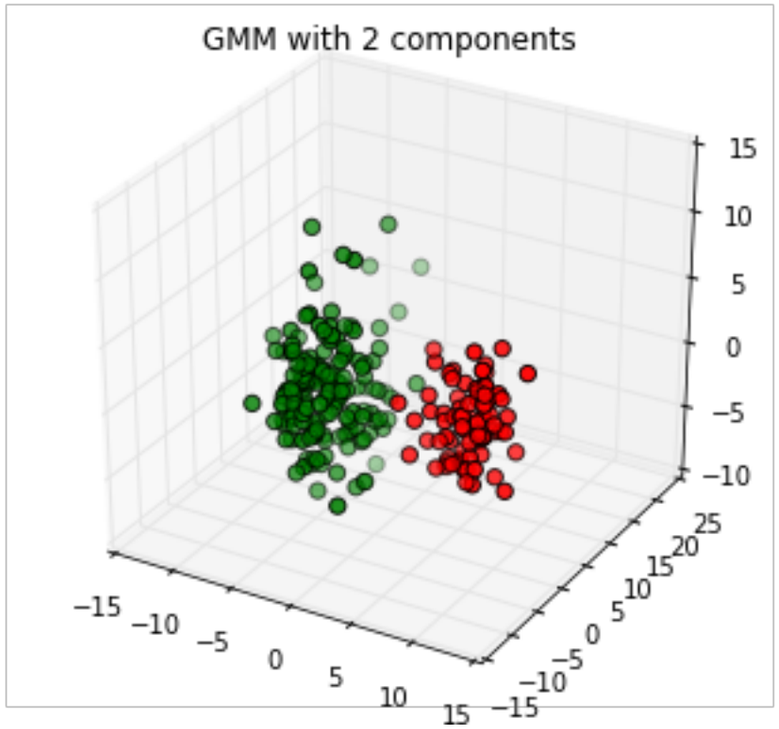
—> Or you can do stochastic gradient descent, of course

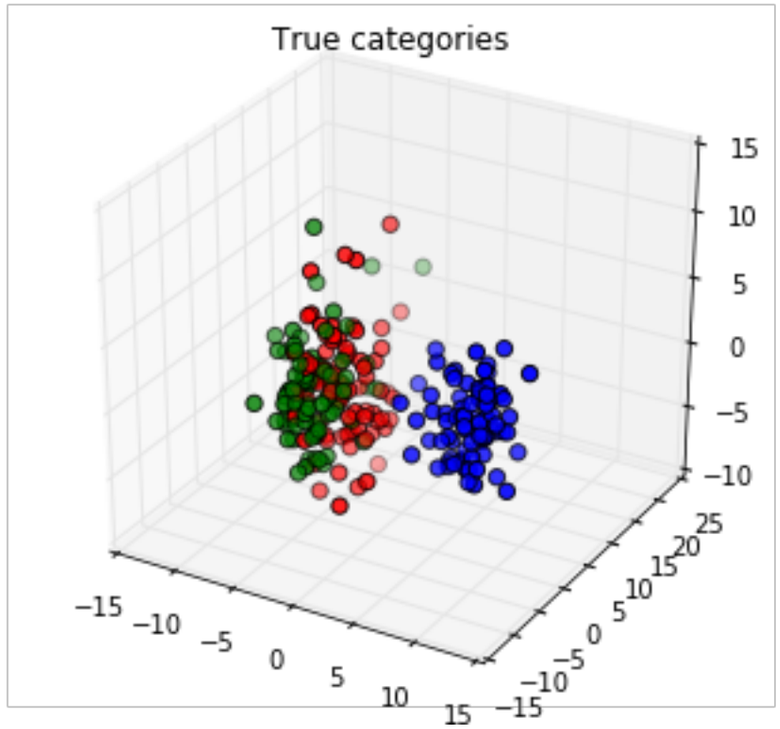
https://www.tensorflow.org/api_docs/python/tf/contrib/factorization

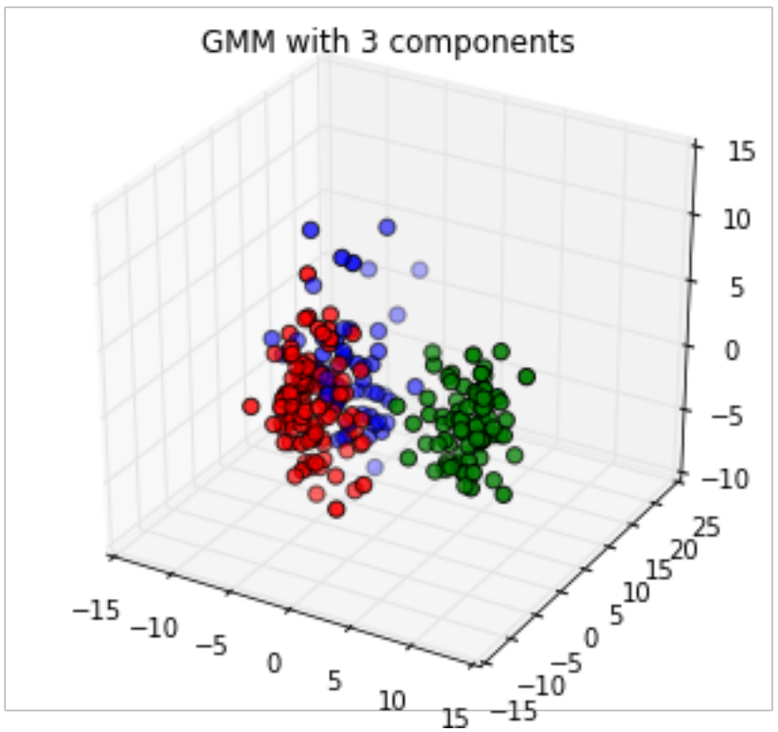
Mixture Models

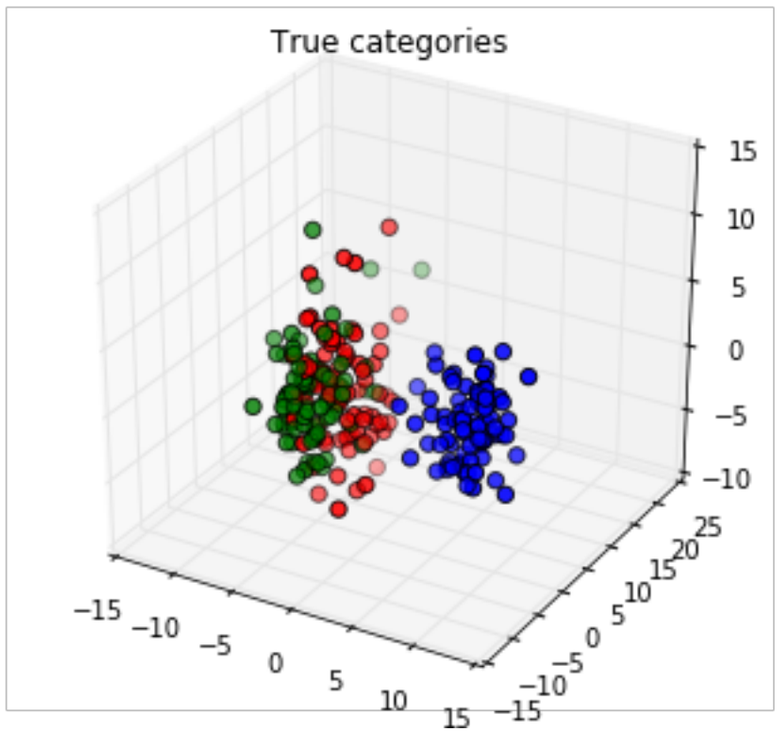
[IPYNB: Gaussian Mixture Models]

Validating Unsupervised Models?









Validating Unsupervised Models?

Akaike Information Criterion (AIC)

$$\mathrm{AIC} = 2 \cdot k - 2 \log(\hat{L})$$

k number of parameters of model

$\hat{L}$ likelihood of the data (according to the model)

Validating Unsupervised Models?

Akaike Information Criterion (AIC)

penalty for too many parameters

$$\mathrm{AIC} = \boxed{2 \cdot k} - \boxed{2 \log(\hat{L})}$$

measure of accuracy

k number of parameters of model

$\hat{L}$ likelihood of the data (according to the model)

Validating Unsupervised Models?

Akaike Information Criterion (AIC)

penalty for too many parameters

$$\mathrm{AIC} = \boxed{2 \cdot k} - \boxed{2 \log(\hat{L})}$$

measure of accuracy

k number of parameters of model

$\hat{L}$ likelihood of the data (according to the model)

$$k = 2 \cdot n_c \cdot d$$

n_c number of components

d number of data dimensions

Validating Unsupervised Models?

Bayesian Information Criterion (BIC)

bigger penalty than AIC

$$\mathrm{BIC} = \log(n) \cdot k - 2\log(\hat{L})$$

k number of parameters of model

$\hat{L}$ likelihood of the data (according to the model)

n number of data points

Validating Unsupervised Models?

Bayesian Information Criterion (BIC)

bigger penalty than AIC

$$\mathrm{BIC} = \log(n) \cdot k - 2\log(\hat{L})$$

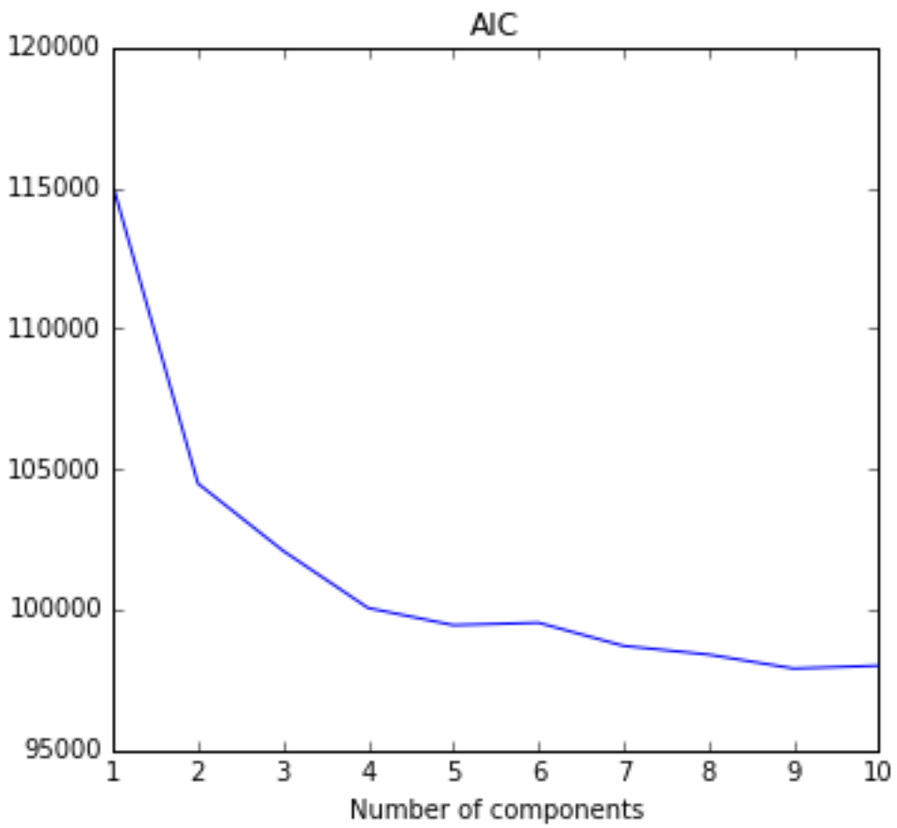
k number of parameters of model

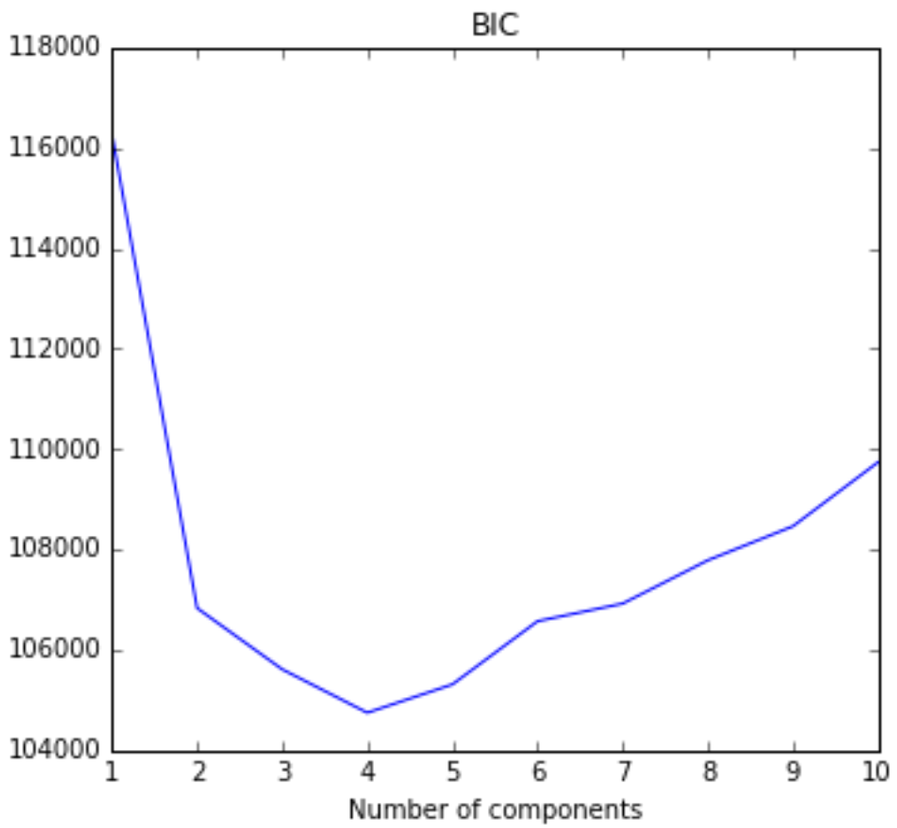
$\hat{L}$ likelihood of the data (according to the model)

n number of data points

$k = 2 \cdot n_c \cdot d$ as before

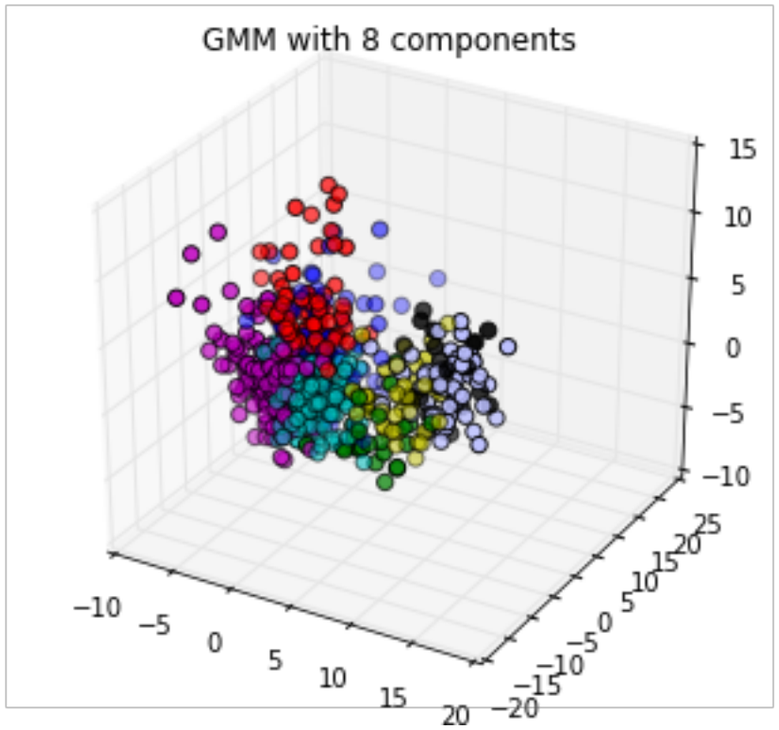
Validating Unsupervised Models?

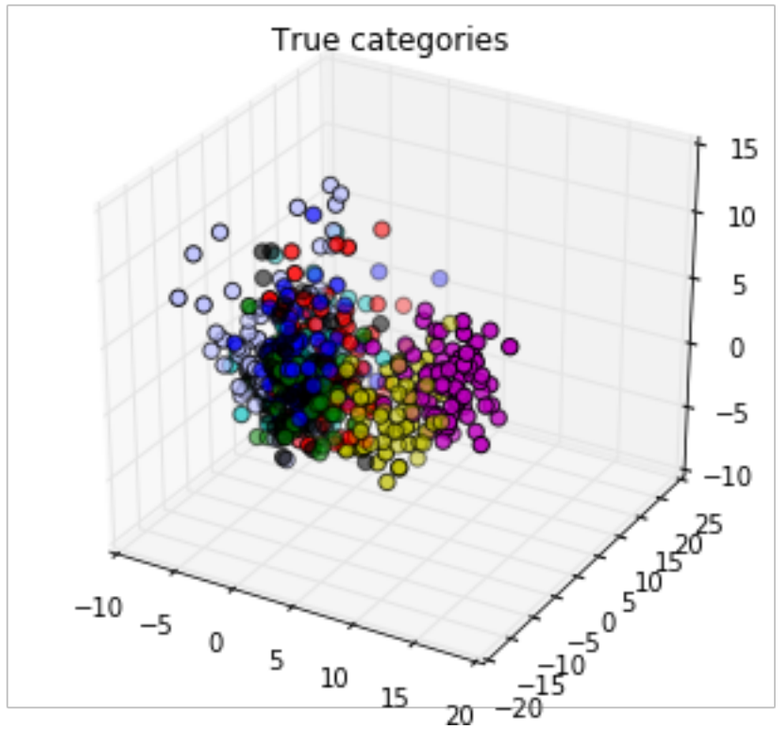


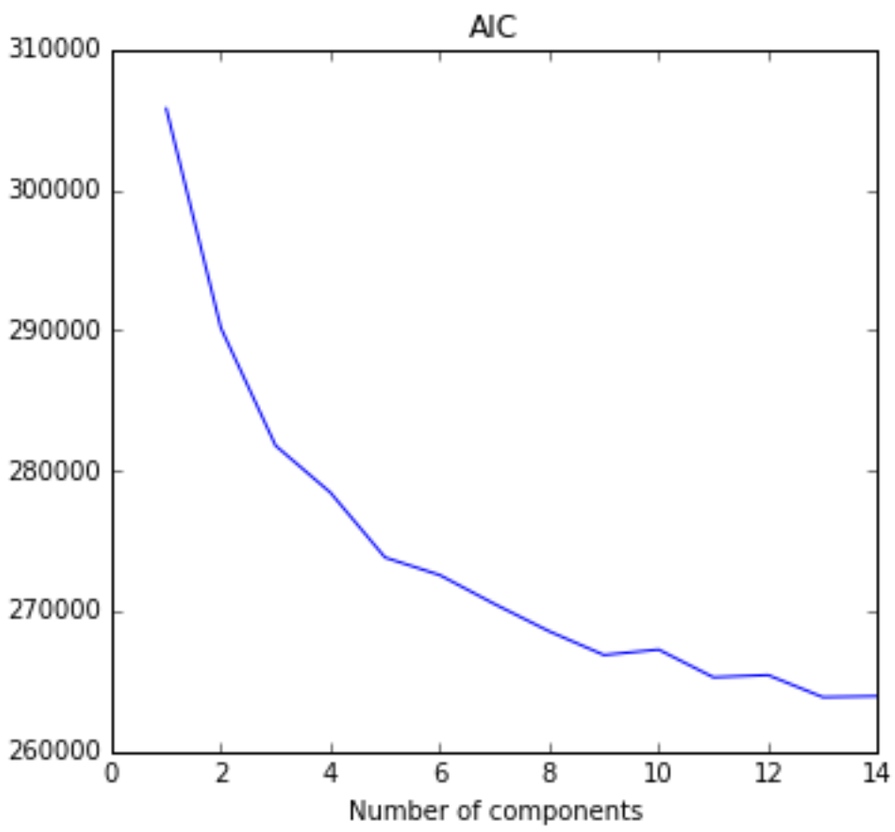


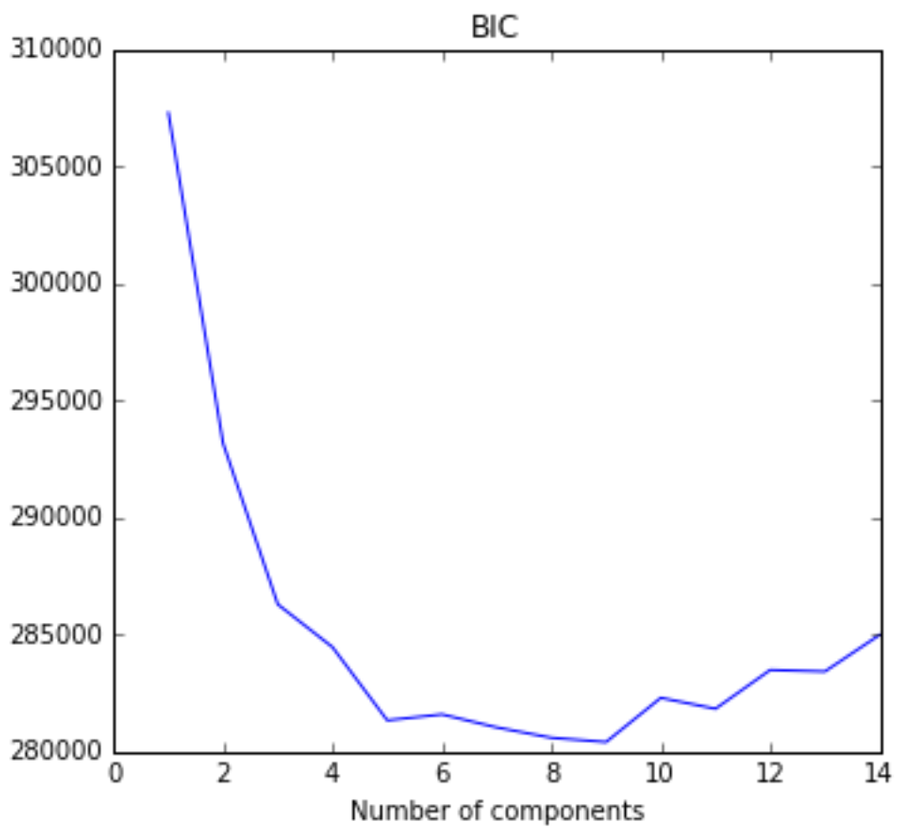
Validating Unsupervised Models?

Variation 0 data



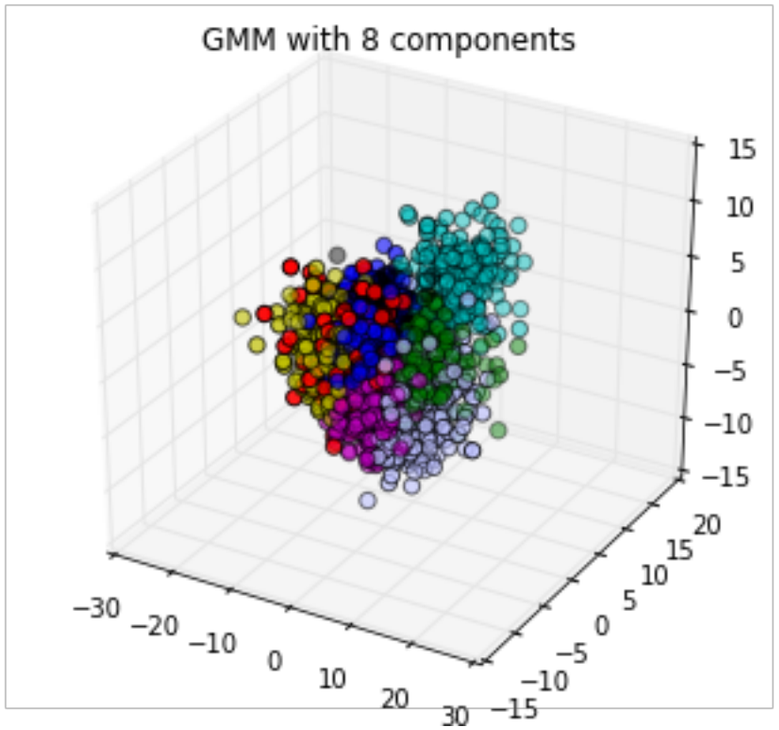


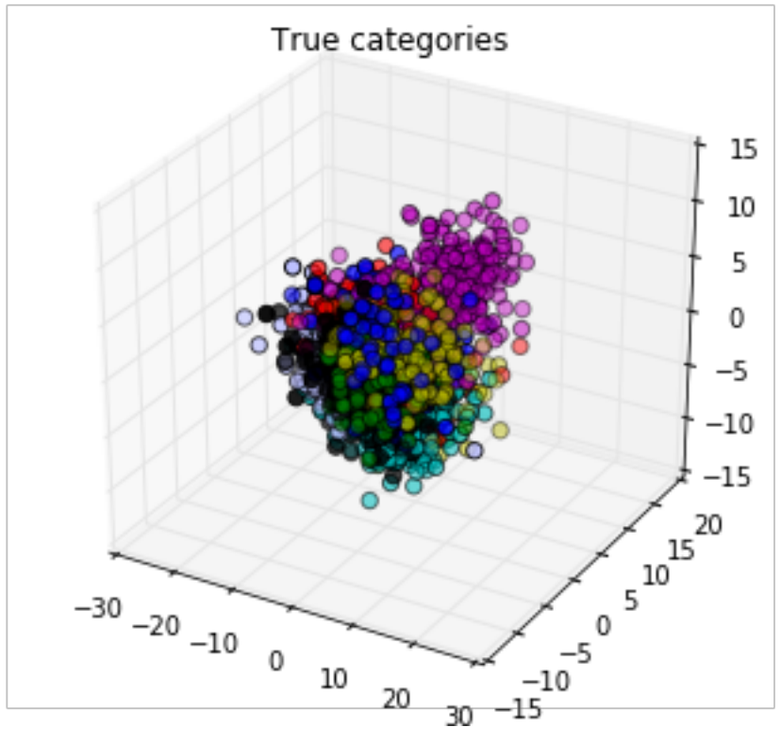


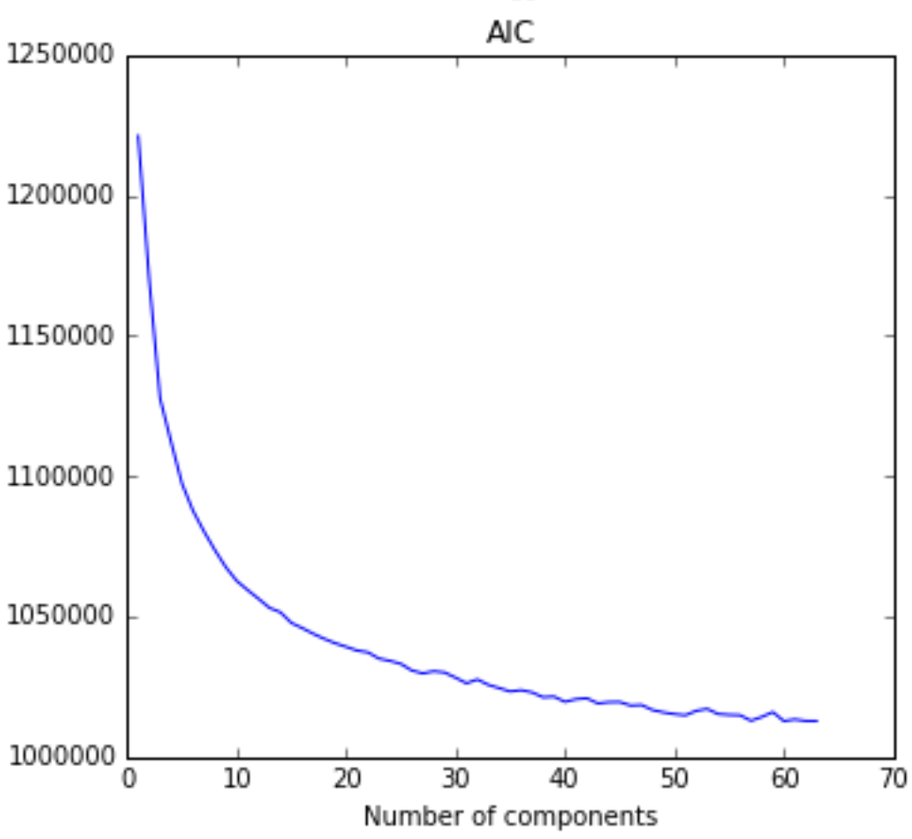


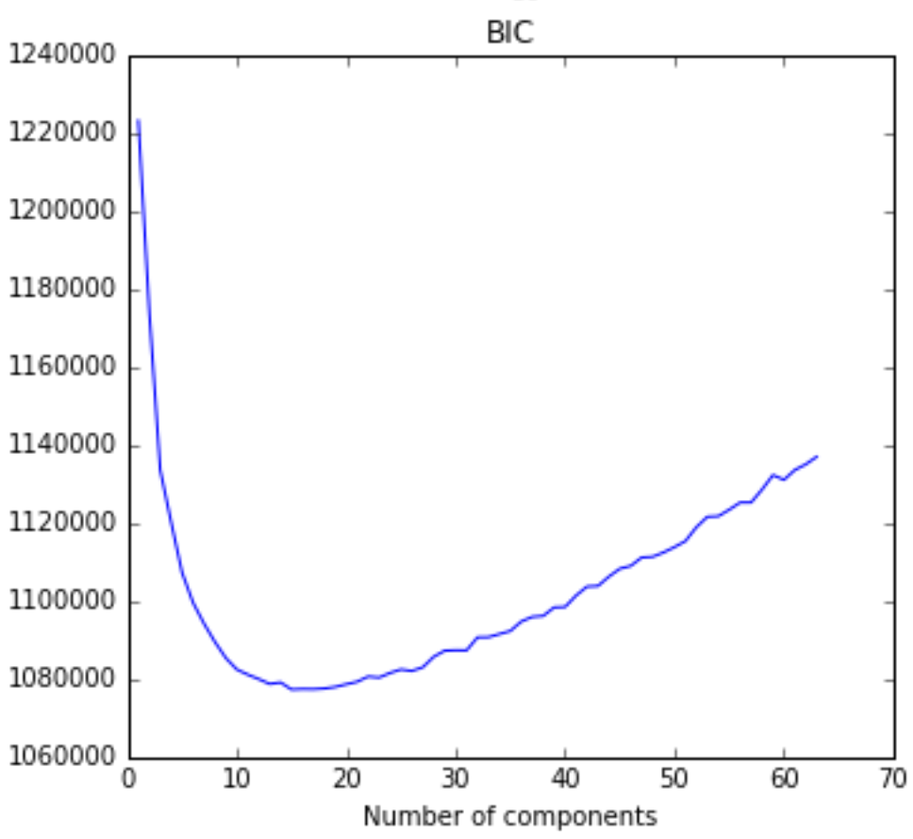
Validating Unsupervised Models?

Variation 3 data



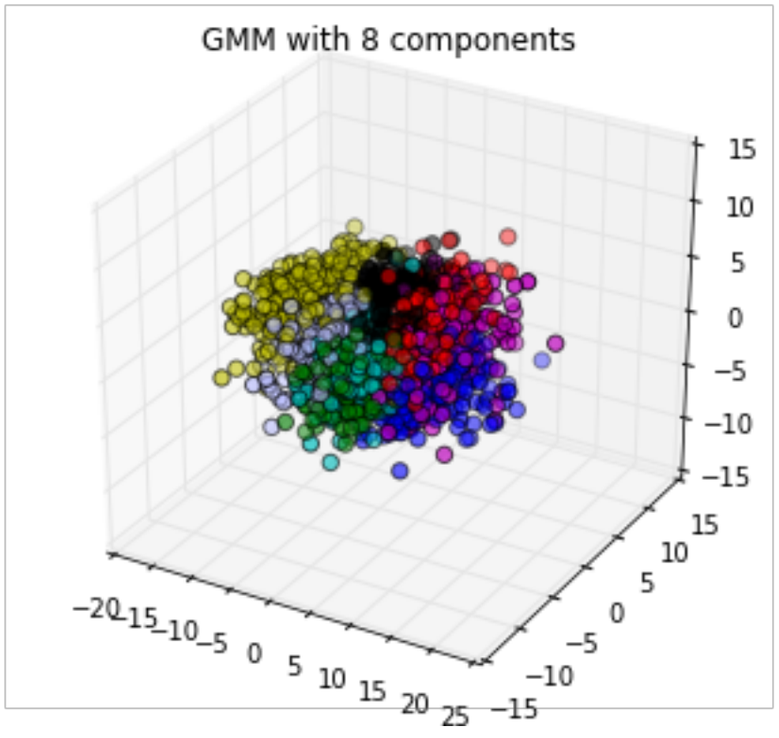


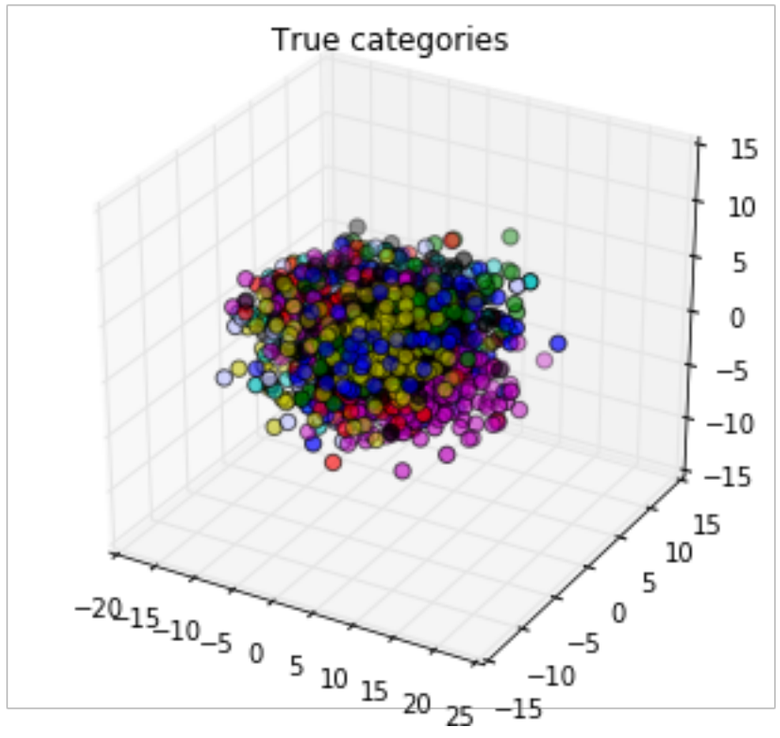




Validating Unsupervised Models?

From V4 data





Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Why is this a good idea?

$$\mathrm{BIC} = \log(n) \cdot k - 2\log(\hat{L})$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Why is this a good idea?

$$\mathrm{BIC} = \log(n) \cdot k - 2\log(\hat{L})$$

$f(x|\theta)$ = parametric distribution describing models as a function of parameters $\boldsymbol{\theta}$ and data $\boldsymbol{x}$

$\pi(\theta)$ = prior distribution of parameters $\boldsymbol{\theta}$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Why is this a good idea?

$$\mathrm{BIC} = \log(n) \cdot k - 2\log(\hat{L})$$

$f(x|\theta)$ = parametric distribution describing models as a function of parameters $\boldsymbol{\theta}$ and data $\boldsymbol{x}$

$\pi(\theta)$ = prior distribution of parameters $\boldsymbol{\theta}$

Probability of observations $\boldsymbol{x}$ is:

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

$$= \int e^{\log[f(x|\theta)]}\pi(\theta)d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

$$= \int e^{\log[f(x|\theta)]}\pi(\theta)d\theta$$

$$= \int e^{\mathcal{L}(x,\theta)}\pi(\theta)d\theta$$

$\mathcal{L}(x,\theta)$ = log-likelihood function

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

$$= \int e^{\log[f(x|\theta)]}\pi(\theta)d\theta$$

$$= \int e^{\mathcal{L}(x,\theta)}\pi(\theta)d\theta \qquad \mathcal{L}(x,\theta) = \text{log-likelihood function}$$

Key idea: approximate $\mathcal{L}(x,\theta)$ near maximum-likelihood estimate

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$p(x) = \int f(x|\theta)\pi(\theta)d\theta$$

$$= \int e^{\log[f(x|\theta)]}\pi(\theta)d\theta$$

$$= \int e^{\mathcal{L}(x,\theta)}\pi(\theta)d\theta \qquad \mathcal{L}(x,\theta) = \text{log-likelihood function}$$

Key idea: approximate $\mathcal{L}(x,\theta)$ near maximum-likelihood estimate

$\hat{\theta}$ = maximum-likelihood estimate

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$\mathcal{L}(\theta) =$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) +$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + (\theta - \hat{\theta}) \left. \frac{\partial \mathcal{L}}{\partial \theta} \right|_{\theta = \hat{\theta}}$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + (\theta - \hat{\theta}) \left.\frac{\partial \mathcal{L}}{\partial \theta}\right|_{\theta=\hat{\theta}} + \frac{1}{2}(\theta - \hat{\theta})^T \left.\frac{\partial^2 \mathcal{L}}{\partial \theta \partial \theta^T}\right|_{\theta=\hat{\theta}} (\theta - \hat{\theta})$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + (\theta - \hat{\theta}) \frac{\partial \mathcal{L}}{\partial \theta}\bigg|_{\theta=\hat{\theta}} + \frac{1}{2}(\theta - \hat{\theta})^T \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \theta^T}\bigg|_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \dots$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + (\theta - \hat{\theta}) \cancel{\frac{\partial \mathcal{L}}{\partial \theta}\bigg|_{\theta=\hat{\theta}}} + \frac{1}{2}(\theta - \hat{\theta})^T \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \theta^T}\bigg|_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \ldots$$

0 since $\hat{\theta}$ is the MLE

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$\hat{\theta}$ = maximum-likelihood estimate

So now, what's the Taylor series of $\mathcal{L}(x, \theta)$ near $\hat{\theta}$?

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + (\theta - \hat{\theta}) \cancel{\frac{\partial \mathcal{L}}{\partial \theta}\bigg|_{\theta=\hat{\theta}}} + \frac{1}{2}(\theta - \hat{\theta})^T \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \theta^T}\bigg|_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \ldots$$

0 since $\hat{\theta}$ is the MLE

Thus,

$$\mathcal{L}(\theta) = \mathcal{L}(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^T \frac{\partial^2 \mathcal{L}}{\partial \theta \partial \theta^T}\bigg|_{\theta=\hat{\theta}} (\theta - \hat{\theta}) + \ldots$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Similarly:

$$\pi(\theta) = \pi(\hat{\theta}) + (\theta - \hat{\theta})^T \left.\frac{\partial \pi(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}} + \dots$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Similarly:

$$\pi(\theta) = \pi(\hat{\theta}) + (\theta - \hat{\theta})^T \frac{\partial \pi(\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}} + \dots$$

Putting the two Taylor series together we get:

$$p(x) \sim \int e^{\mathcal{L}(\hat{\theta}) + \frac{1}{2}(\theta-\hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta-\hat{\theta})} \cdot \left[\pi(\hat{\theta}) + (\theta - \hat{\theta})^T \nabla_\theta \pi(\hat{\theta})\right] d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

Similarly:

$$\pi(\theta) = \pi(\hat{\theta}) + (\theta - \hat{\theta})^T \frac{\partial \pi(\theta)}{\partial \theta}\bigg|_{\theta=\hat{\theta}} + \dots$$

Putting the two Taylor series together we get:

$$p(x) \sim \int e^{\mathcal{L}(\hat{\theta}) + \frac{1}{2}(\theta-\hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta-\hat{\theta})} \cdot \left[\pi(\hat{\theta}) + (\theta - \hat{\theta})^T \nabla_\theta \pi(\hat{\theta})\right] d\theta$$

$$= \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

where: $$J(\theta, \hat{\theta}) = (\theta - \hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta - \hat{\theta})$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$\int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

$$||$$

$$e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$\int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2}J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2}J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

||　　　　　　　　||

$$e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2}J(\theta,\hat{\theta})} d\theta \qquad e^{\mathcal{L}(\hat{\theta})} \nabla_\theta \pi(\hat{\theta}) \int e^{\frac{1}{2}J(\theta,\hat{\theta})} (\theta - \hat{\theta}) d\theta$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$\int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

$$\| \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \|$$

$$e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} d\theta \qquad e^{\mathcal{L}(\hat{\theta})} \nabla_\theta \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) d\theta$$

but: $J(\theta,\hat{\theta}) = (\theta - \hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta - \hat{\theta})$

is symmetric in its arguments

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$\int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

$$\| \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \|$$

$$e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} d\theta \qquad e^{\mathcal{L}(\hat{\theta})} \nabla_\theta \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) d\theta$$

but: $J(\theta,\hat{\theta}) = (\theta - \hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta - \hat{\theta})$

is symmetric in its arguments

$\int F(\theta,\hat{\theta})(\theta - \hat{\theta}) d\theta = 0$ for any symmetric function **F**

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

$$\int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} \pi(\hat{\theta}) d\theta + \int e^{\mathcal{L}(\hat{\theta})} e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) \nabla_\theta \pi(\hat{\theta}) d\theta$$

$$\| \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \| \to 0$$

$$e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} d\theta \qquad\qquad e^{\mathcal{L}(\hat{\theta})} \nabla_\theta \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta,\hat{\theta})} (\theta - \hat{\theta}) d\theta$$

but: $J(\theta,\hat{\theta}) = (\theta - \hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta - \hat{\theta})$

is symmetric in its arguments

$\int F(\theta,\hat{\theta})(\theta - \hat{\theta}) d\theta = 0$ for any symmetric function **F**

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So we're left with:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) \int e^{\frac{1}{2} J(\theta, \hat{\theta})} d\theta$$

where:

$$J(\theta, \hat{\theta}) = (\theta - \hat{\theta})^T \nabla^2_\theta \mathcal{L}(\hat{\theta})(\theta - \hat{\theta})$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So we're left with:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})}\pi(\hat{\theta}) \int -e^{\frac{n}{2}J(\theta,\hat{\theta})}d\theta$$

where:

$$J(\theta,\hat{\theta}) = -\frac{1}{n}(\theta-\hat{\theta})^T\nabla^2_\theta\mathcal{L}(\hat{\theta})(\theta-\hat{\theta})$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So we're left with:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})}\pi(\hat{\theta}) \int -e^{\frac{n}{2}J(\theta,\hat{\theta})}d\theta$$

where:

$$J(\theta,\hat{\theta}) = -\frac{1}{n}(\theta-\hat{\theta})^T\nabla^2_\theta\mathcal{L}(\hat{\theta})(\theta-\hat{\theta})$$

But since the above is normal distribution:

$$\int e^{-\frac{1}{2}(\hat{\theta}-\theta)^T\nabla^2_\theta\mathcal{L}(\hat{\theta})(\theta-\hat{\theta})}d\theta = \frac{(2\pi)^{p/2}}{\sigma_n^p}\frac{1}{\sqrt{|\nabla^2_\theta\mathcal{L}(\hat{\theta})|}}$$

where $\boldsymbol{p}$ = dimension of the parameter space

$\boldsymbol{\sigma}$ = std of process $\quad \sigma \sim \sqrt{n}$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) (2\pi)^{p/2} n^{-p/2} H(\hat{\theta})$$

$$H(\hat{\theta}) = (\nabla^2_\theta \log f(x_0, \theta))^{-1/2}$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) (2\pi)^{p/2} n^{-p/2} \boxed{H(\hat{\theta})}$$

$O(1)$ in $\boldsymbol{n}$

$$H(\hat{\theta}) = (\nabla^2_\theta \log f(x_0, \theta))^{-1/2}$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) (2\pi)^{p/2} n^{-p/2} \boxed{H(\hat{\theta})}$$

$O(1)$ in $\boldsymbol{n}$

$$H(\hat{\theta}) = (\nabla^2_\theta \log f(x_0, \theta))^{-1/2}$$

Thus:

$$-2 \log p(x) = -2 \log f(x, \hat{\theta}) + p \log(n) + O(1)$$

Theoretical Tangent

Following Konishi and Kitagawa (§9.1.3)

So:

$$p(x) \sim e^{\mathcal{L}(\hat{\theta})} \pi(\hat{\theta}) (2\pi)^{p/2} n^{-p/2} \boxed{H(\hat{\theta})} \qquad O(1) \text{ in } \boldsymbol{n}$$

$$H(\hat{\theta}) = (\nabla^2_\theta \log f(x_0, \theta))^{-1/2}$$

Thus:

$$-2 \log p(x) = -2 \log f(x, \hat{\theta}) + p \log(n) + O(1)$$

Whence:

$$\mathrm{BIC} = -2 \log f(x, \hat{\theta}) + p \log(n)$$