Machine Learning

Dirichlet Processes

Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

November 4, 2019

http://www.carlhenrik.com

Introduction

Bayes Rule

$$p(\theta|\mathcal{Y}) = p(\mathcal{Y}|\theta)p(\theta)\frac{1}{p(\mathcal{Y})}$$

Evidence

$$p(\mathcal{Y}) = \int p(\mathcal{Y}|\theta)p(\theta)\mathrm{d}\theta$$

Regression Models

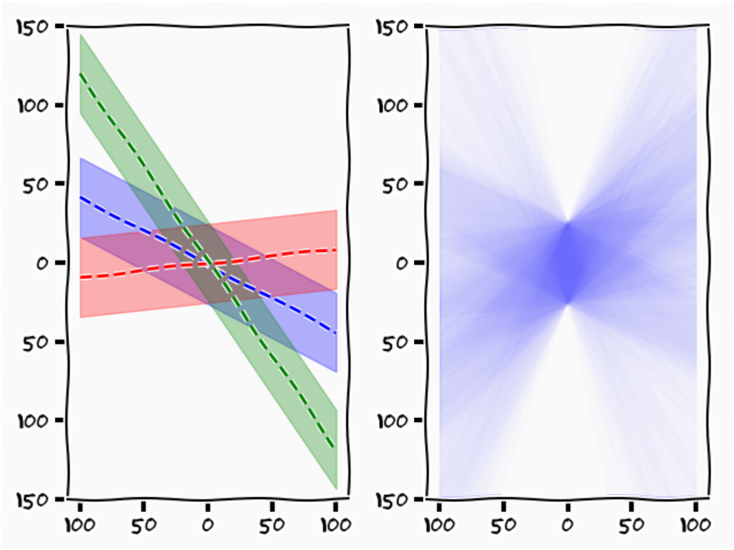
Linear Linear Model

$$p(y_i|x_i, \mathbf{w}) = \mathcal{N}(w_0 + w_1 \cdot x_i, \beta^{-1})$$

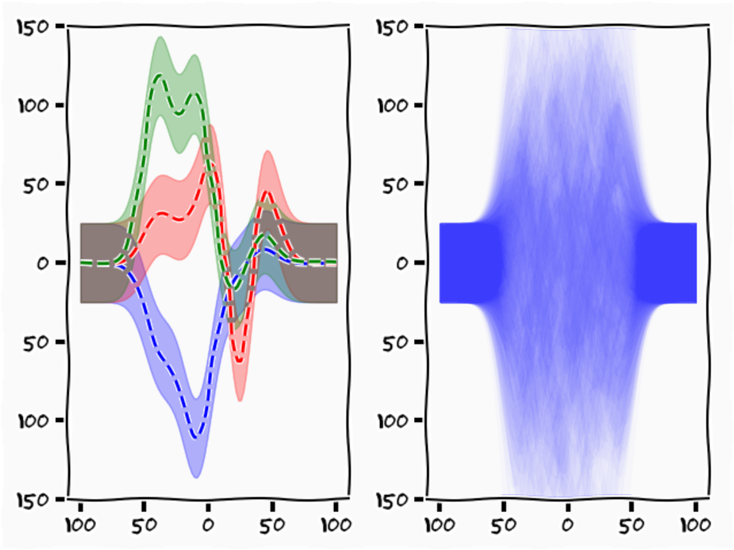
Basis function

$$p(y_i|x_i, \mathbf{w}) = \mathcal{N}(\sum_{i=1}^{6} w_i \phi(x_i), \beta^{-1})$$

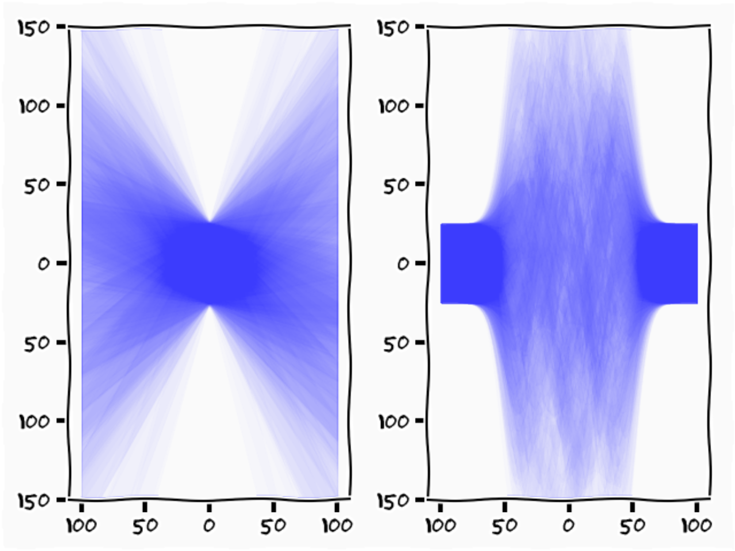
Model Linear Linear



Model Linear Basis

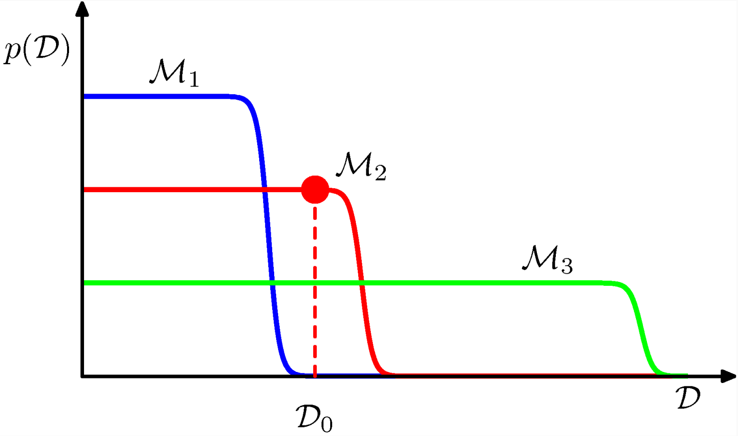


Evidence

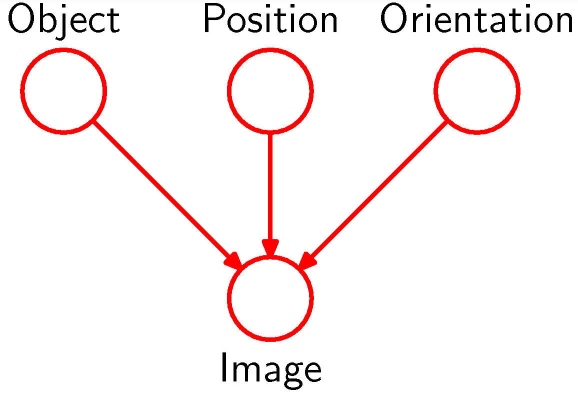


150
100
50
0
50
100
150
100
50
0
50
100
150
100
50
0
50
100
150
100
50
0
50
100

Model Selection



Explaining Away



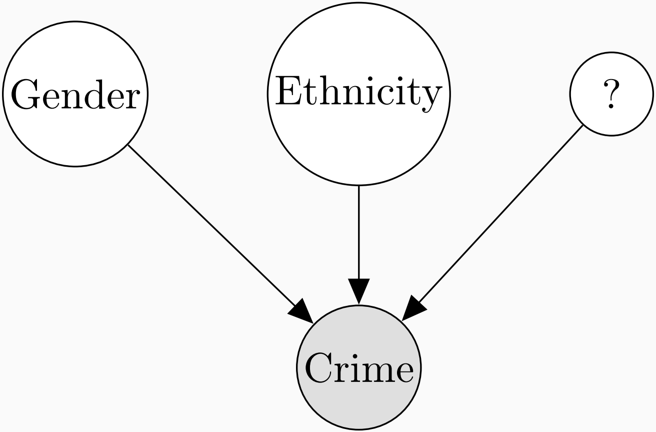
- The Object variable *explains away* variance associated with objects from the image
  - → position won't contain object variations
  - → orientation won't contain object variations

Explaining Away

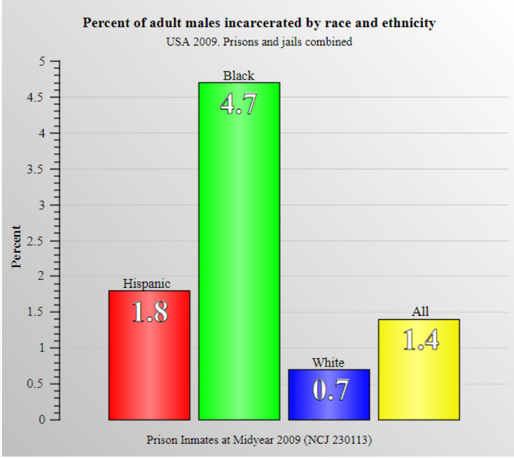
$$p(\text{Image}) = \int p(\text{Image}|\text{Object}, \text{Position}, \text{Orientation})p(\text{Object})p(\text{Position})$$

- $p(\text{Object}|\text{Image})$ what is the object
- $p(\text{Object}|\text{Image}, \text{Orientation})$ what is the object given that I know the orientation

Explaining Away

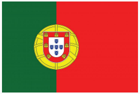


Explaining Away[1]



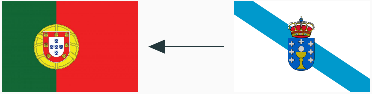
[1] https://artificialintelligence-news.com/2019/01/22/ai-sentencing-people-risk-assessment/

Hierarchical Knowledge



$p(\text{Portugal})$

Hierarchical Knowledge

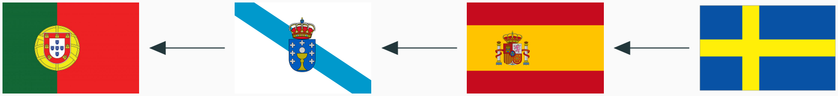


$$p(\text{Portugal}) = \int p(\text{Portugal}|\text{Galicia})p(\text{Galicia})\mathrm{d}\text{Galicia}$$

Hierarchical Knowledge

$$p(\text{Portugal}) = \int p(\text{Portugal}|\text{Galicia})p(\text{Galicia}|\text{Spain})p(\text{Spain})$$

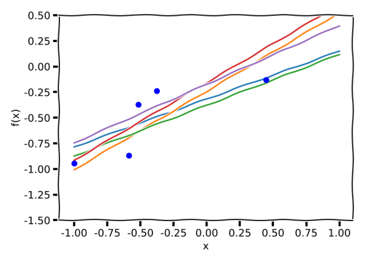
Hierarchical Knowledge

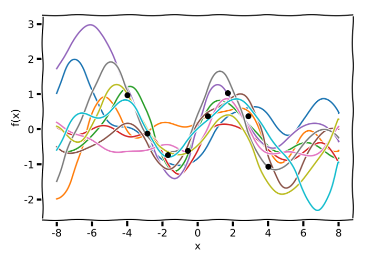


$$p(\text{Portugal}) = \int p(\text{Portugal}|\text{Galicia})p(\text{Galicia}|\text{Spain})p(\text{Spain}|\text{Sweden})p(\text{Sweden})$$

Non-parametrics

Non-parametrics





Parametric model

- Number of parameters fixed with respect to sample size
- fixed parameter space

Non-parametric model

- Number of parameters grows with sample size
- ∞ -dimensional parameter space

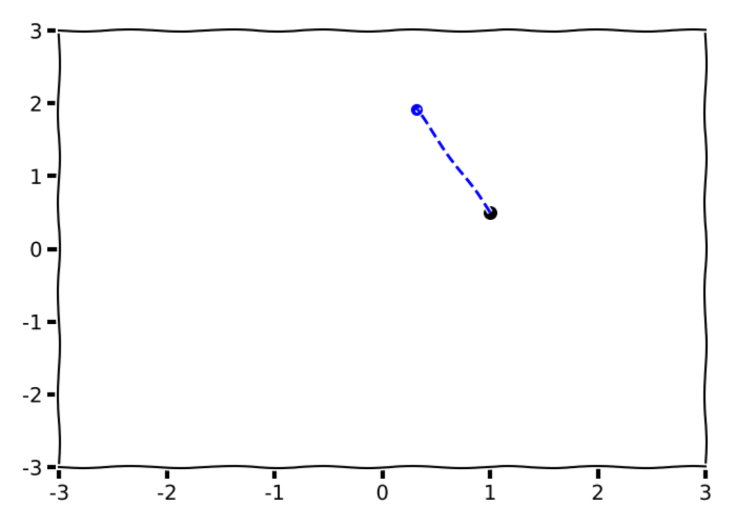
Nearest Neighbour

- Training data: $\{\mathsf{x}_i, y_i\}_{i=1}^{N}$
- Test data: $\{\mathsf{x}_i\}_{i=1}^{M}$
- Inference

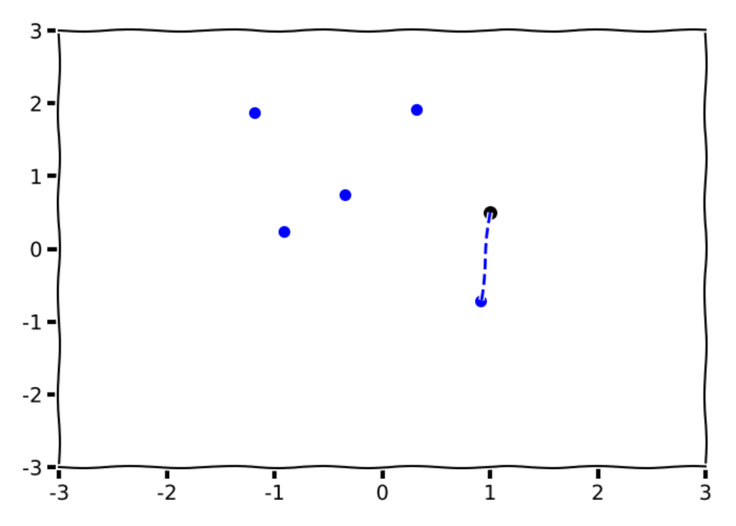
$$\hat{i} = \text{argmin}_i D(\mathsf{x}_*, \mathsf{x}_i)$$

- Complexity grows with number of training data
- Does not generalise at all

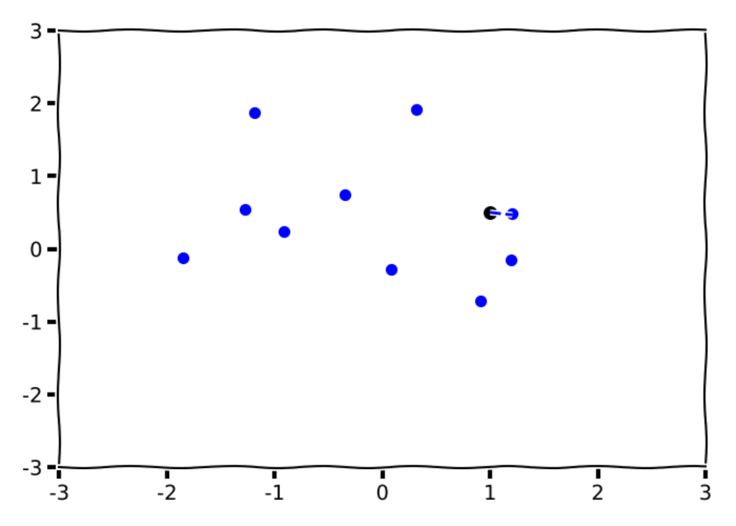
Nearest Neighbour



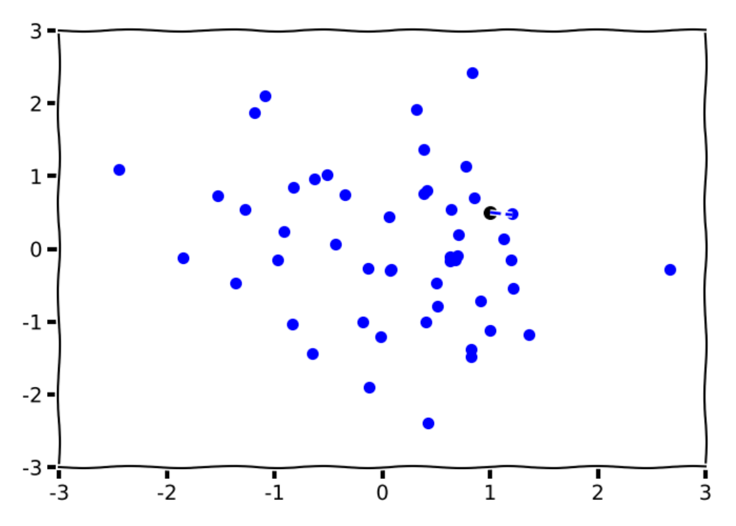
Nearest Neighbour



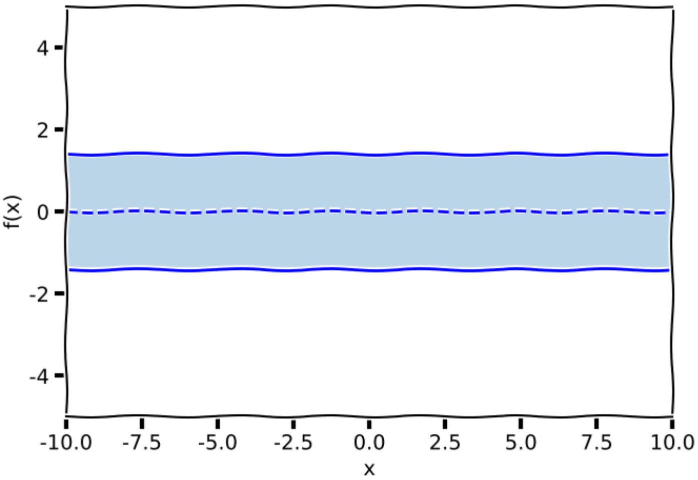
Nearest Neighbour



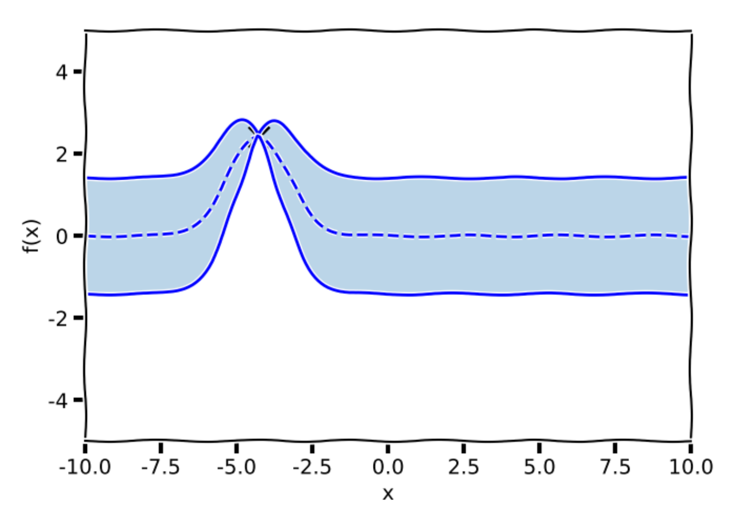
Nearest Neighbour



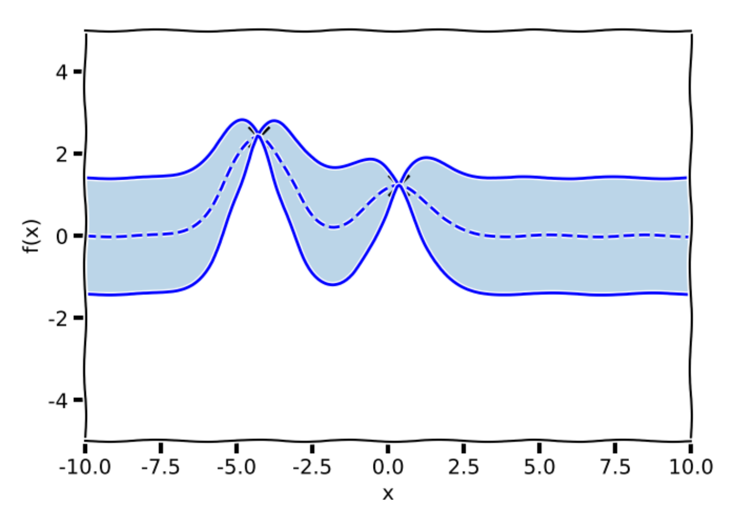
Gaussian Processes



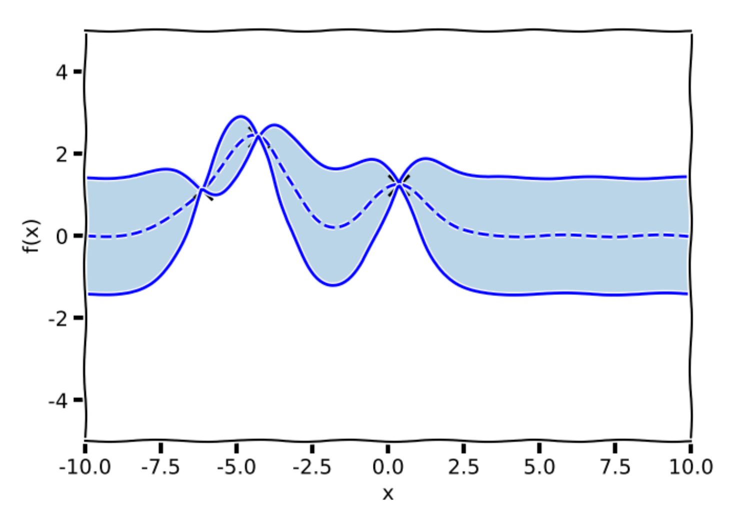
Gaussian Processes



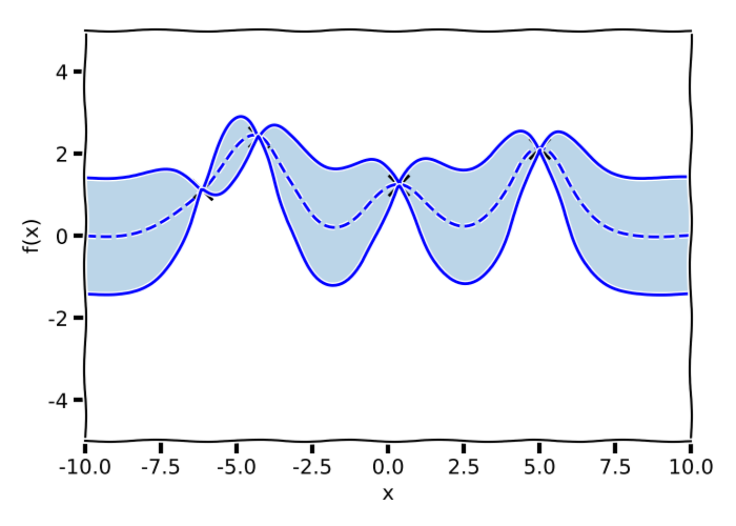
Gaussian Processes



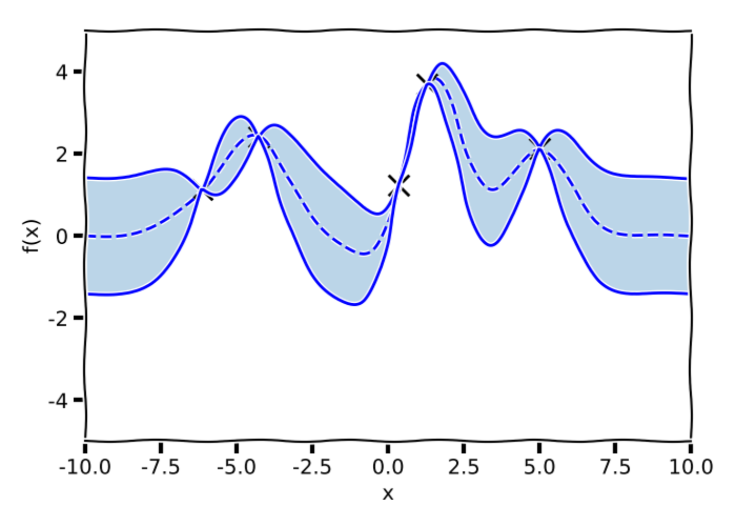
Gaussian Processes



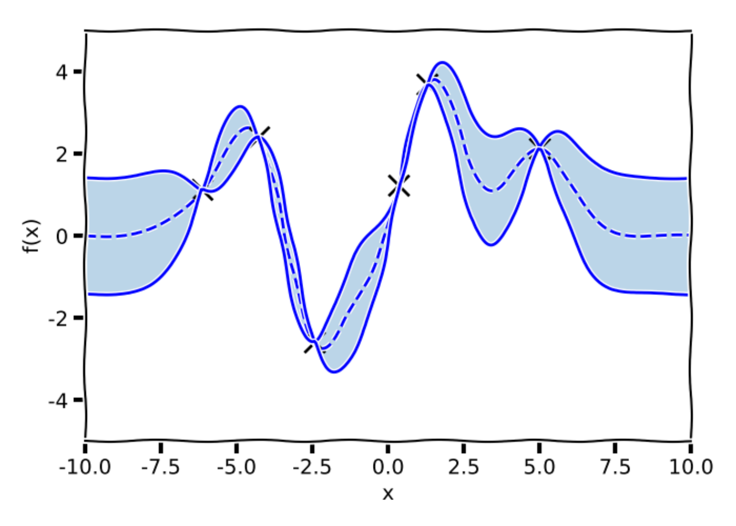
Gaussian Processes



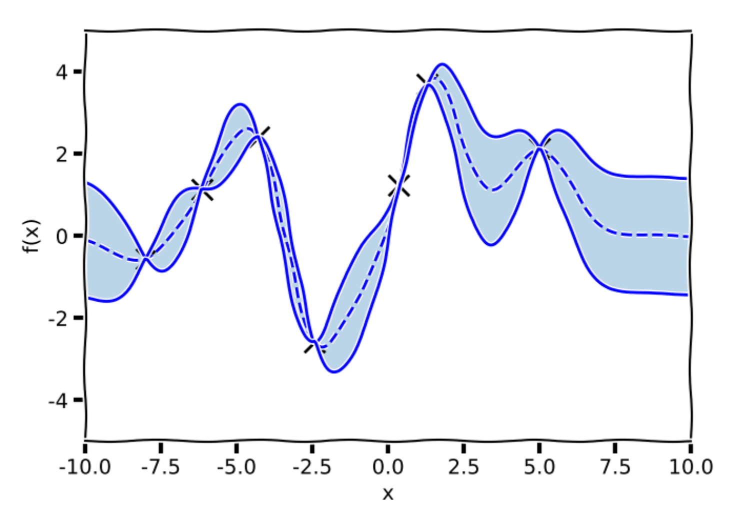
Gaussian Processes



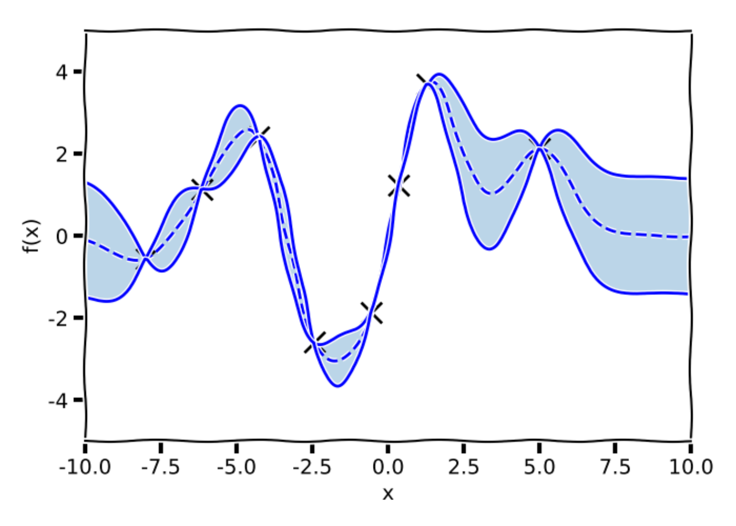
Gaussian Processes

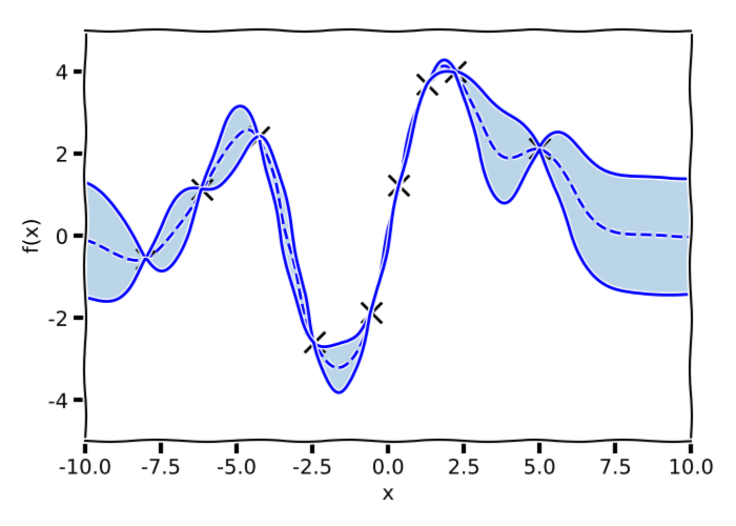


Gaussian Processes

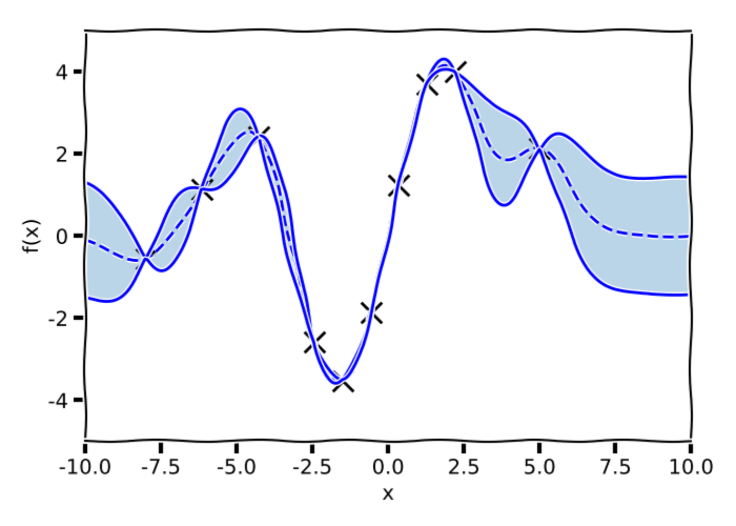


Gaussian Processes

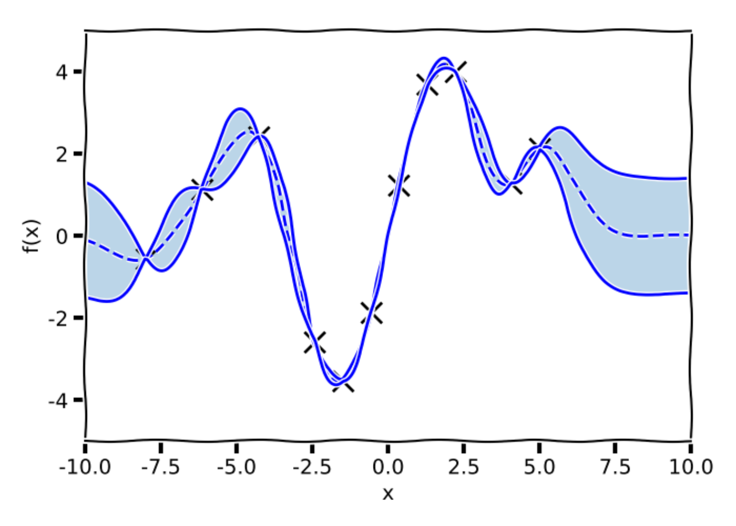



4
2
0
-2
-4
f(x)
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0
7.5
10.0
x

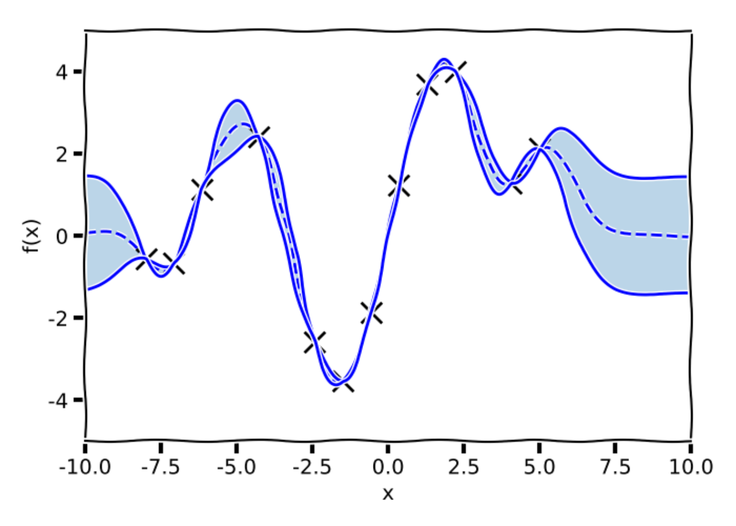
Gaussian Processes



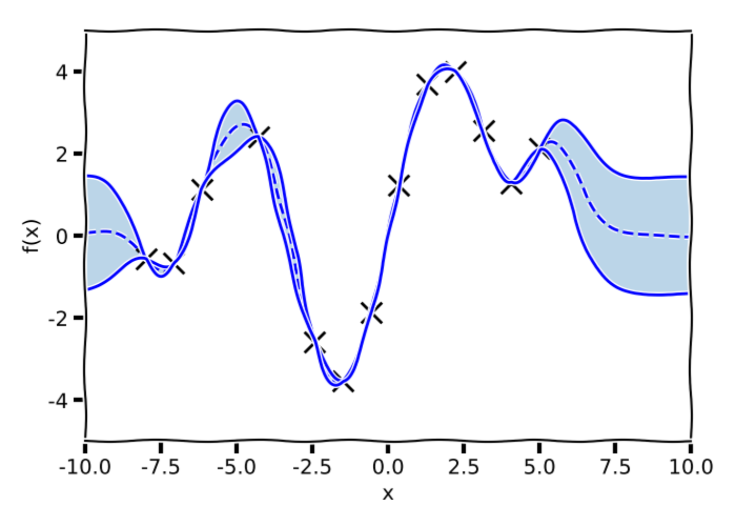
Gaussian Processes



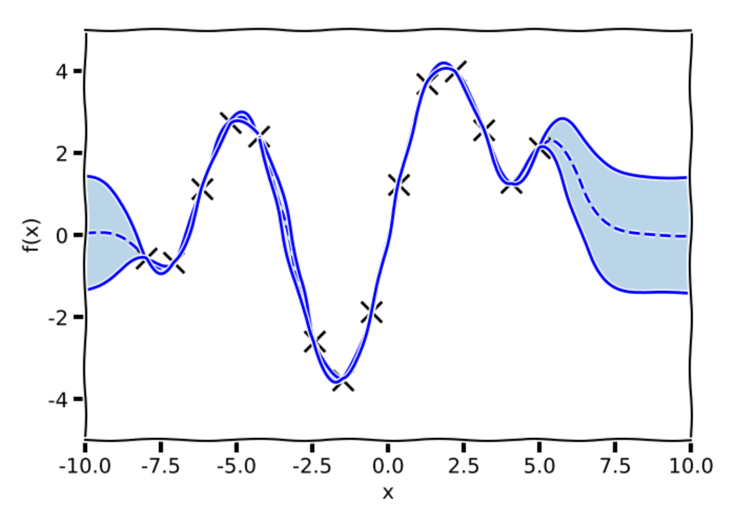
Gaussian Processes



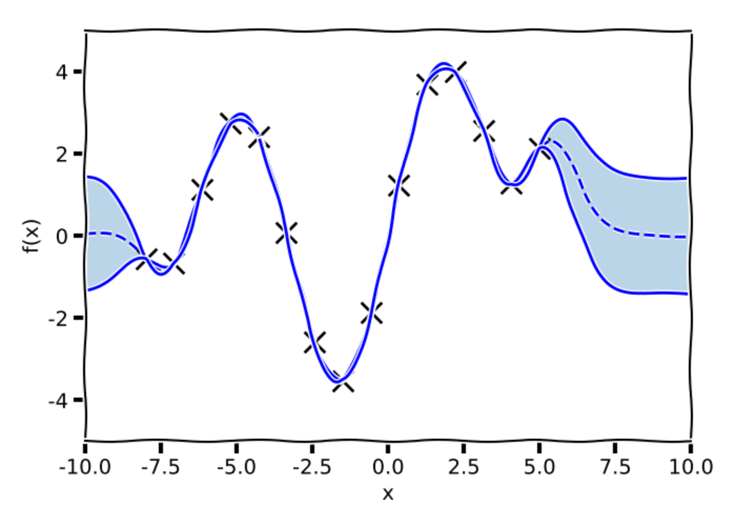
Gaussian Processes



Gaussian Processes



Gaussian Processes



Process $\rightarrow$ Distribution $\rightarrow$ value

- Each evaluation of a process is a distribution

$$\mathcal{N}(0, \Sigma) \sim \mathcal{N}(0, k(\mathbf{X}, \mathbf{X}))$$

Process → Distribution → value

- Each evaluation of a process is a distribution

$$\mathcal{N}(0, \Sigma) \sim \mathcal{N}(0, k(\mathbf{X}, \mathbf{X}))$$

- Each evaluation of a distribution is a value

$$y \sim \mathcal{N}(y|0, \Sigma)$$

Process $\rightarrow$ Distribution $\rightarrow$ value

- Each evaluation of a process is a distribution

$$\mathcal{N}(0, \Sigma) \sim \mathcal{N}(0, k(\mathbf{X}, \mathbf{X}))$$

- Each evaluation of a distribution is a value

$$y \sim \mathcal{N}(y|0, \Sigma)$$

- Parametric models are definedy by distributions and non-parametric by processes

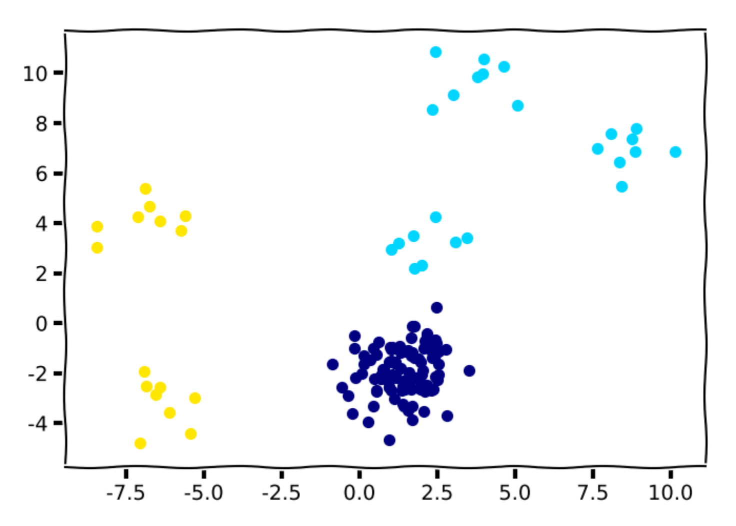
Gaussian Process

- Formulate process
- Evaluate process at specific location $x \rightarrow$ distribution
- Evaluate distribution at any location y
- GP is defined over uncountable infinite space

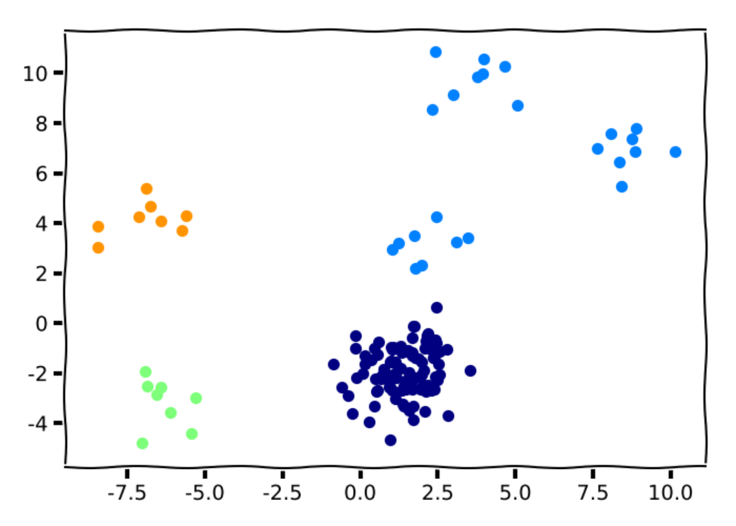
Gaussian Process

- Formulate process
- Evaluate process at specific location $x \rightarrow$ distribution
- Evaluate distribution at any location y
- GP is defined over uncountable infinite space
- *What about countable objects?*

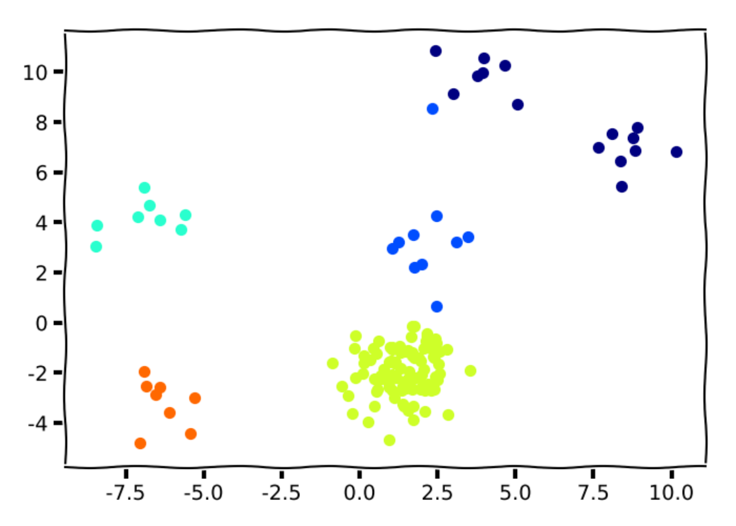
Gaussian Mixture Model



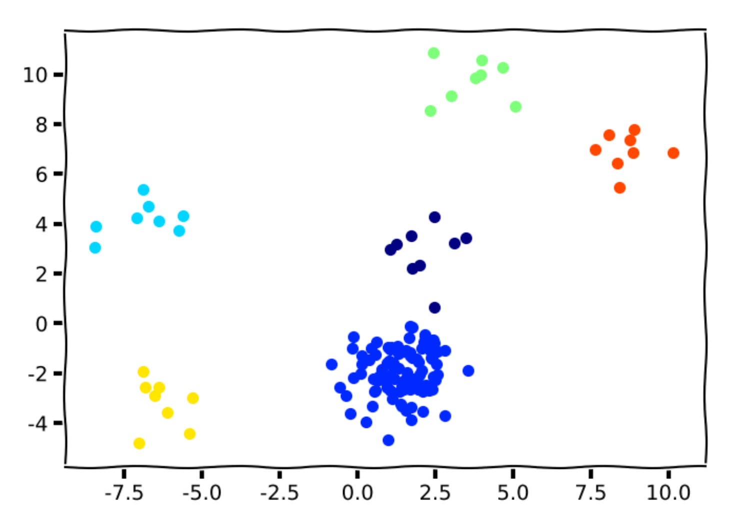
Gaussian Mixture Model



Gaussian Mixture Model



Gaussian Mixture Model



Gaussian Mixture Models

$$p(\mathbf{X}) = \sum_{k=1}^{K} p(\mathbf{X}|k)p(k) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{X}|\boldsymbol{\mu}_k, \Sigma_k)p(k)$$

- Represent the probability of $\mathbf{X}$ as a combination or *mixture* of distributions
- What should K be?
- Can we make K infinite?

Non-parametrics

Gaussian Process

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|f)p(f|\mathbf{X}, \theta)\mathrm{d}f$$

Infinite Mixture Model

$$p(\mathbf{X}) = \sum_{k=1}^{\infty} p(\mathbf{X}|k)p(k) = \sum_{k=1}^{\infty} \mathcal{N}(\mathbf{X}|\boldsymbol{\mu}_k, \Sigma_k)p(k)$$

- When we build models we describe how the data has been generated

Unsupervised Linear Regression

1. Sample (pick) a weight

Clustering

Generative Models

- When we build models we describe how the data has been generated

Unsupervised Linear Regression

1. Sample (pick) a weight
2. Sample (pick) a input location

Clustering

Generative Models

- When we build models we describe how the data has been generated

Unsupervised Linear Regression

1. Sample (pick) a weight
2. Sample (pick) a input location
3. Generate observation

Clustering

Generative Models

- When we build models we describe how the data has been generated

Unsupervised Linear Regression

1. Sample (pick) a weight
2. Sample (pick) a input location
3. Generate observation

Clustering

1. Sample cluster identity

Generative Models

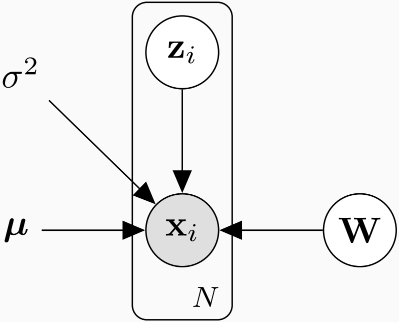
- When we build models we describe how the data has been generated

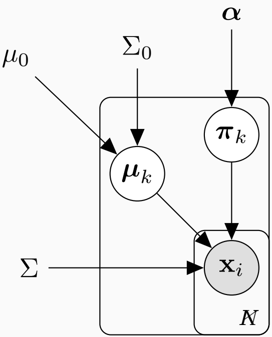
Unsupervised Linear Regression

1. Sample (pick) a weight
2. Sample (pick) a input location
3. Generate observation

Clustering

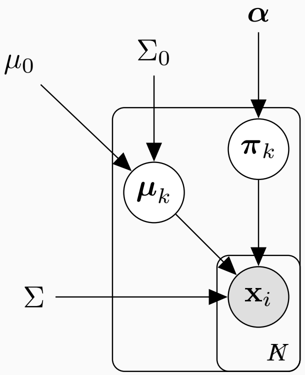
1. Sample cluster identity
2. Sample point from cluster





- Graphical model clearly shows generative proceedure

Gaussian Mixture Model



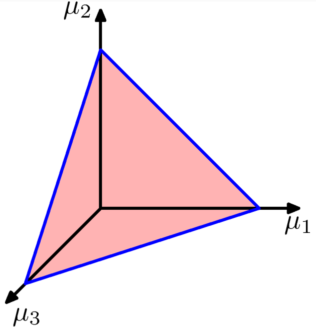
1. Sample proportions
2. Sample cluster id given proportions
3. Sample cluster mean
4. Sample data

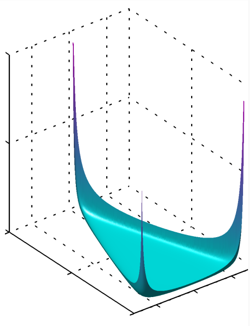
Distributions over partitionings [1] Ch 2.2.0

$$\text{Mult}(m_1, m_2, \ldots, m_k | \boldsymbol{\mu}, N) = \binom{N}{m_1, m_2, \ldots, m_K} \prod_{k=1}^{K} \mu_k^{m_k}$$

- Joint distribution over $m_1, m_2, \ldots, m_K$
- The parameter $\boldsymbol{\mu}$ says how likely each component is

Dirichlet Distribution

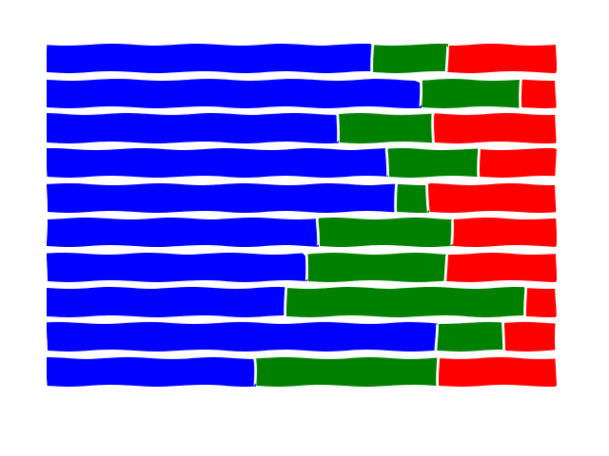




- Conjugate prior to multinomial

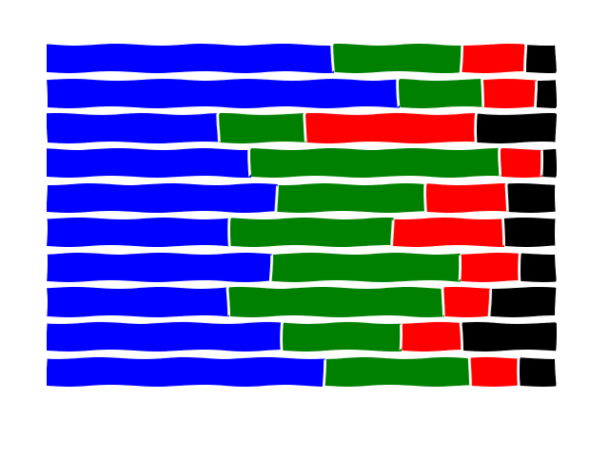
$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdot \ldots \cdot \Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

Dirichlet Distribution



$\mathrm{Dir}(10, 5, 3)$

Dirichlet Distribution



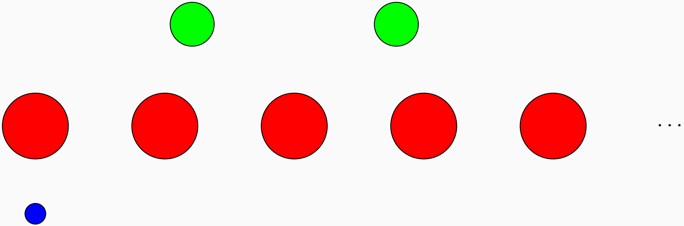
$\mathrm{Dir}(7, 5, 3, 2)$

Dirichlet Processes

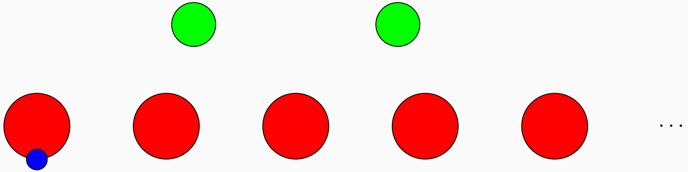
Dirichlet Process

- Is the infinite dimensional generalisation of a Dirichlet distribution
    - just as Gaussian process is of Gaussian distribution
- Generates a partitioning of (possibly) infinite number of elements
- Not as intuitive to write down
- Best written constructively

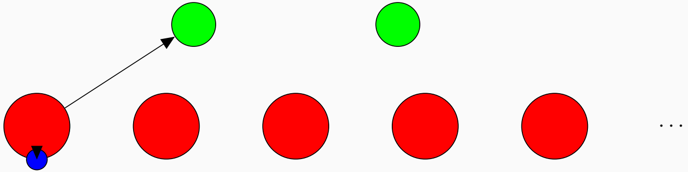
Chinese Resturant Process



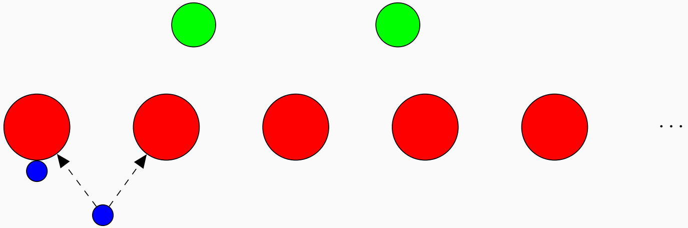
Chinese Resturant Process



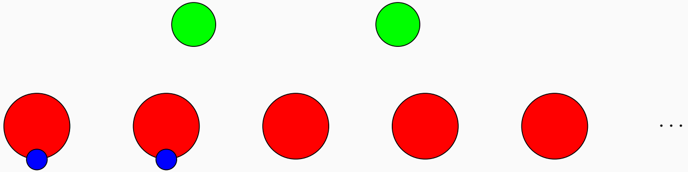
Chinese Resturant Process



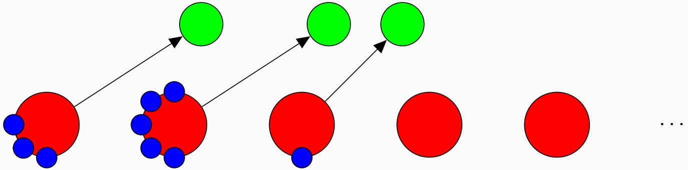
Chinese Resturant Process



Chinese Resturant Process

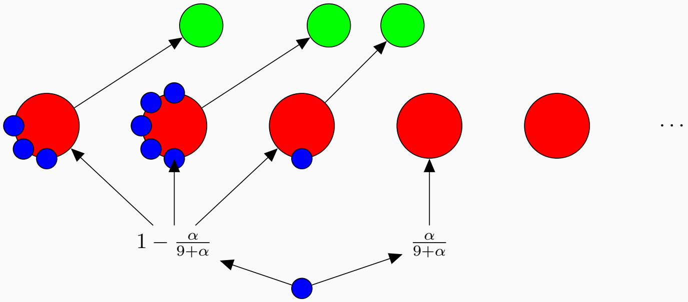


Chinese Resturant Process

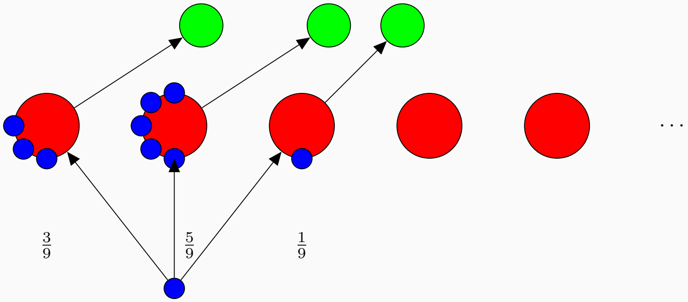


- Go to new table $\frac{\alpha}{N-1+\alpha}$
- If not choose table as $\frac{n_i}{N}$ where n_i number of diners at table *in*

Chinese Resturant Process



Chinese Resturant Process

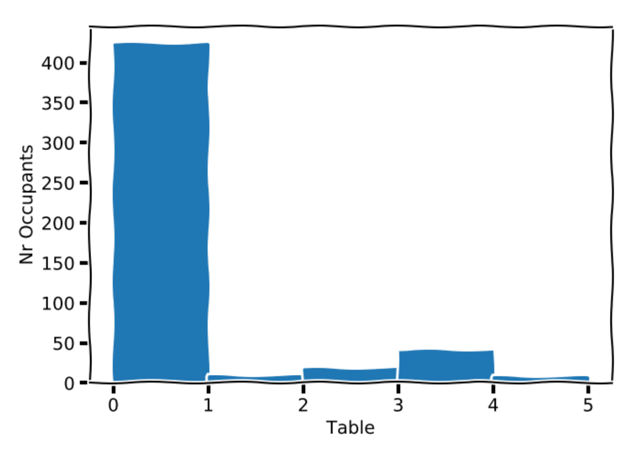


Chinese Resturant Process Code

Code

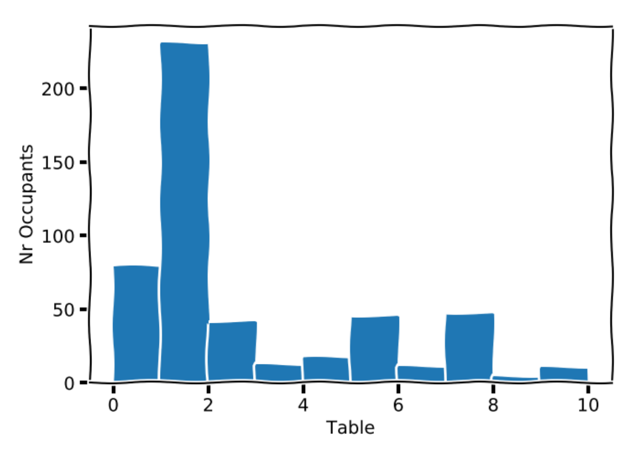
```
def chrp(N, alpha):
    table_assignments = np.zeros(N)
    next_open_table = 0
    for i in range(0,N):
        r = np.random.random()
        if r < (alpha/(i+alpha)):
            table_assignments[i] = next_open_table
            next_open_table += 1
        else:
            index = int(np.round((i-1)*np.random.random()
            table_assignments[i] = table_assignments[inde

    return table_assignments
```

Chinese Resturant Process



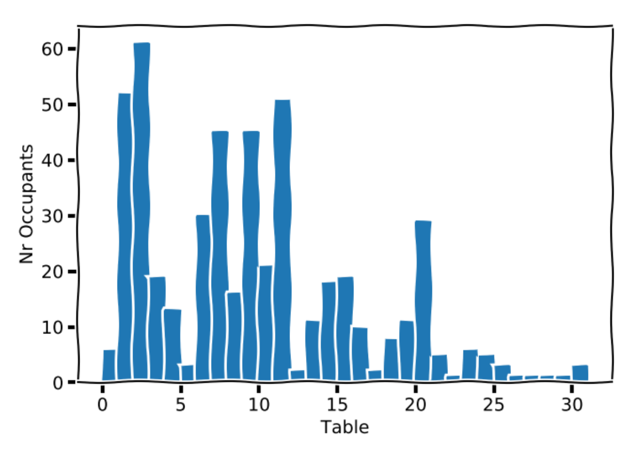
$$N = 500 \quad \alpha = 1.0$$

Chinese Resturant Process



$$N = 500 \quad \alpha = 2.0$$

Chinese Resturant Process



$$N = 500 \quad \alpha = 10.0$$

Chinese Resturant Process

N	α	μ_K	σ_K
500	3	14.3	8.81
500	5	21.6	15.64
500	10	39.1	27.29
500	20	64.3	63.8
500	100	180	69.69

Infinite Mixture Model

1. Pick first data-point

Infinite Mixture Model

1. Pick first data-point
2. Pick the first mixture

Infinite Mixture Model

1. Pick first data-point
2. Pick the first mixture
3. Pick parameters associated with this mixture μ_k, σ_k

Infinite Mixture Model

1. Pick first data-point
2. Pick the first mixture
3. Pick parameters associated with this mixture μ_k, σ_k
4. Next data-point

Infinite Mixture Model

1. Pick first data-point
2. Pick the first mixture
3. Pick parameters associated with this mixture μ_k, σ_k
4. Next data-point
5. Associate with a new mixture or create new

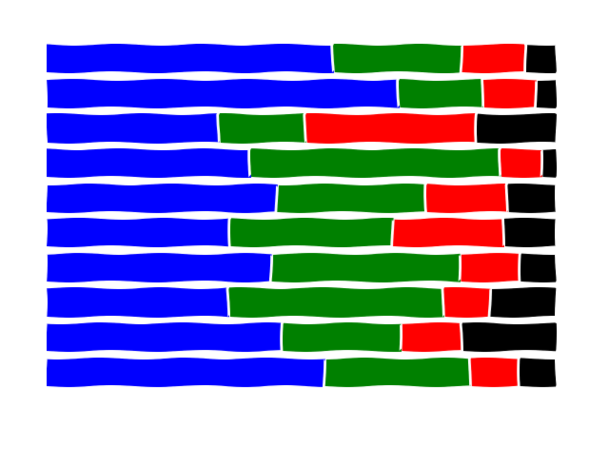
Infinite Mixture Model

1. Pick first data-point
2. Pick the first mixture
3. Pick parameters associated with this mixture μ_k, σ_k
4. Next data-point
5. Associate with a new mixture or create new
6. If new, pick new parameters

Infinite Mixture Model

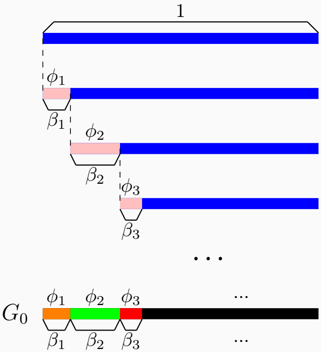
1. Pick first data-point
2. Pick the first mixture
3. Pick parameters associated with this mixture μ_k, σ_k
4. Next data-point
5. Associate with a new mixture or create new
6. If new, pick new parameters
7. Repeat

Dirichlet Distribution



$\mathrm{Dir}(7, 5, 3, 2)$

Stick-Breaking



$$G \sim \mathrm{DP}(\mathcal{H}, \alpha)$$

$$\hat{\beta}_k \sim \mathrm{Beta}(1, \alpha)$$

$$\beta_k = \hat{\beta}_k \prod_{l=1}^{k-1} (1 - \hat{\beta}_l)$$

$$\Phi \sim \mathcal{H}$$

$$G_0 = \sum_{k=1}^{\infty} \beta_k \Phi_k$$

This is all really different

- A probability measure is a measure on how much we believe in a specific setting of a variable

This is all really different

- A probability measure is a measure on how much we believe in a specific setting of a variable
- We have derived a formulation that provides measure on any partition

This is all really different

- A probability measure is a measure on how much we believe in a specific setting of a variable
- We have derived a formulation that provides measure on any partition
    - We can now search for the configuration that is most likely: ML or MAP etc.

This is all really different

- A probability measure is a measure on how much we believe in a specific setting of a variable
- We have derived a formulation that provides measure on any partition
  - We can now search for the configuration that is most likely: ML or MAP etc.
  - We can try to derive the posterior over the partition

This is all really different

- A probability measure is a measure on how much we believe in a specific setting of a variable
- We have derived a formulation that provides measure on any partition
  - We can now search for the configuration that is most likely: ML or MAP etc.
  - We can try to derive the posterior over the partition
- *We are so far only looking at how to formulate models*

Summary

Summary

- Dirichlet processes are priors over countably infinite sets

Summary

- Dirichlet processes are priors over countably infinite sets
- Allows for models dealing with infinite partitions

Summary

- Dirichlet processes are priors over countably infinite sets
- Allows for models dealing with infinite partitions
    - Infinite Gaussian Mixture Models

Summary

- Dirichlet processes are priors over countably infinite sets
- Allows for models dealing with infinite partitions
    - Infinite Gaussian Mixture Models
- Take home message: generative models

Summary

- Dirichlet processes are priors over countably infinite sets
- Allows for models dealing with infinite partitions
  - Infinite Gaussian Mixture Models
- Take home message: generative models
- Tomorrow: Latent Dirichlet Allocation

eof

References

Christopher M. Bishop.
Pattern Recognition and Machine Learning (Information Science and Statistics).
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.