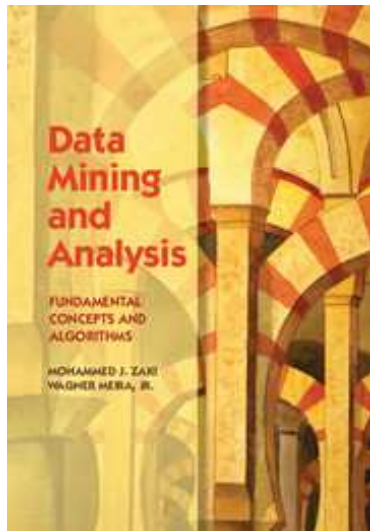


Data Mining for Social and Cyber-Physical Systems

Wagner Meira Jr.¹

¹Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

August 2, 2017



Cambridge Press, 2014, 624 pages.

PDF freely available at
<http://dataminingbook.info/>

- Tabular
 - categorical
 - numeric
- Text
- Graphs
- Sound
- Image
- Video

- Storage
- Accessing
- Engineering
 - Integration
 - Cleaning
 - Transformation
- Visualization

Concept

Automatic extraction of knowledge or patterns that are interesting (novel, useful, implicit, etc.) from large volumes of data.

Tasks

- Data engineering
- Characterization
- Prediction

Concept

A model aims to represent the nature or reality from a specific perspective. A model is an artificial construction where all extraneous details have been removed or abstracted, while keeping the key features necessary for analysis and understanding.

Data Mining Models

Frequent Patterns

Task

Among all possible sets of entities, which ones are the most frequent? Or better, determine the sets of items that co-occur in a database more frequently than a given threshold.

Application Scenario

Market-basket problem: Given that a customer purchased items in set A , what are the most likely items to be purchased in the future?

Data Mining Models

Clustering

Task

Given a similarity criterion, what is the entity partition that groups together the most similar entities?

Application Scenario

Customer segmentation: Partition a customer base into groups of similar customers, supporting different policies and strategies for each group.

Data Mining Models

Classification

Task

Given some knowledge about a domain, including classes or categories of entities, and a sample whose class is unknown, predict the class of the latter based on the existing knowledge.

Application Scenario

Credit scoring: A bank needs to decide whether it will loan money to a given person. It may use past experience with other persons who present a similar profile to decide whether or not it is worth giving the loan.

Paradigms

- **Combinatorial**
- Probabilistic
- Algebraic
- Graph-based

Domain

Models partition (or select) entities based on their attributes and their combinations. Search space is discrete and finite, although potentially very large.

Task

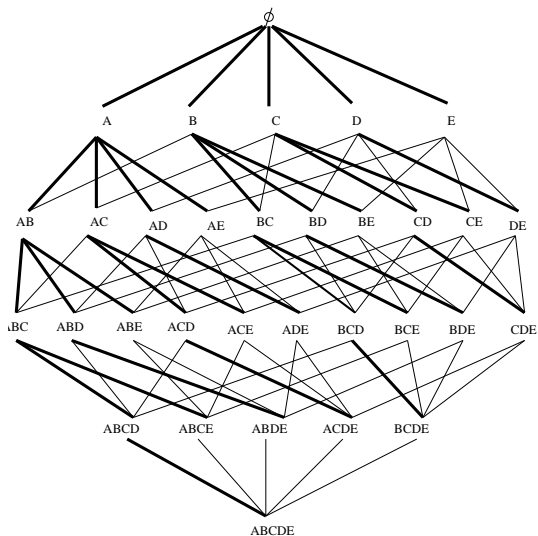
Determine the best model according to a quality metric.

Strategies

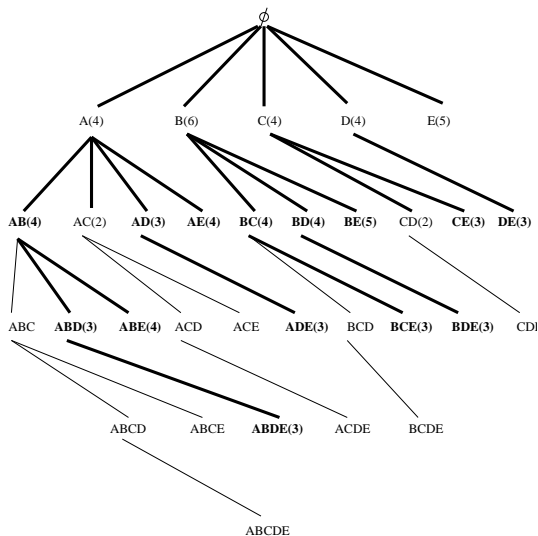
- Pruning exhaustive search
- Heuristic approximation

- Frequent Itemset Mining
- k-Means
- DBScan
- Decision trees

Combinatorial Models



Combinatorial Models



Paradigms

- Combinatorial
- **Probabilistic**
- Algebraic
- Graph-based

Domain

Models are based on one or more probability density function(s) (PDF). Given a model and a dataset, search its parameter space, which may be continuous and/or discrete.

Task

Determine the best parameter models for a dataset, according to an optimization metric.

Strategies

- Direct
- Iterative

- Expectation-Maximization
- DenClue
- Naive Bayes

Probabilistic Models

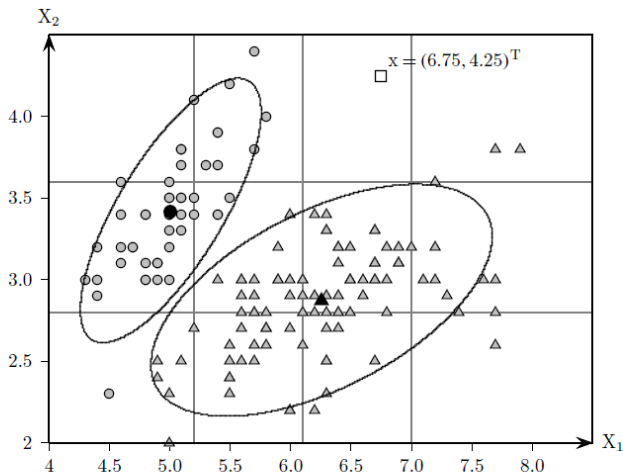


Figure 18.1. Iris data: X_1 :sepal length versus X_2 :sepal width. The class means are show in black; the density contours are also shown. The square represents a test point labeled x .

Probabilistic Models

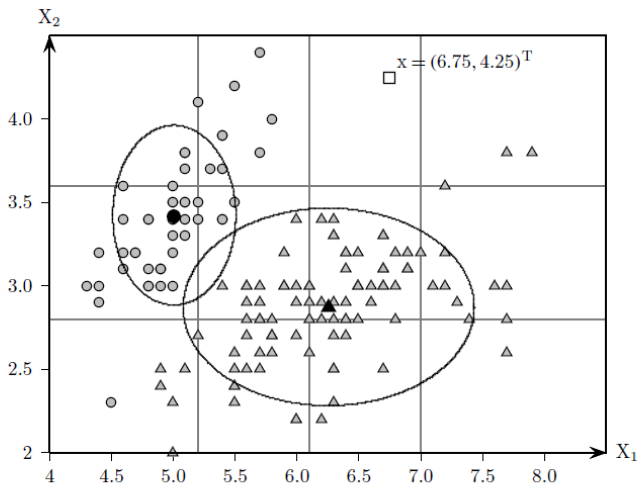
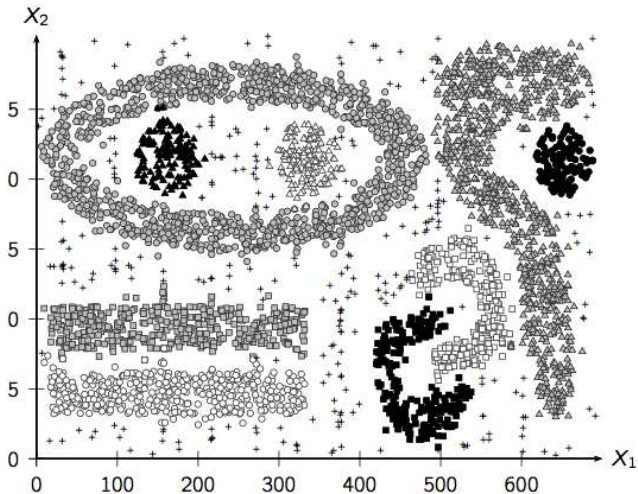
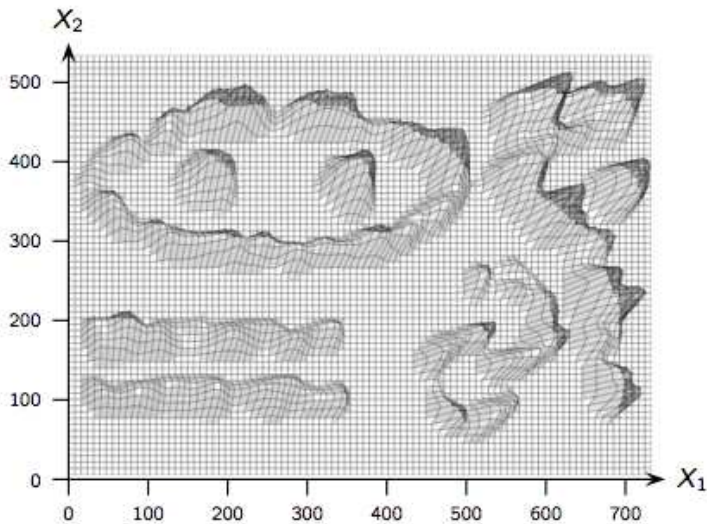


Figure 18.2. Naive Bayes: X_1 :sepal length versus X_2 :sepal width. The class means are shown in black; the density contours are also shown. The square represents a test point labeled x .

Probabilistic Models



Probabilistic Models



Paradigms

- Combinatorial
- Probabilistic
- **Algebraic**
- Graph-based

Domain

Problem is modeled using linear algebra, enabling several existing algebraic models and algorithms.

Task

Determine the best models and their parameters, according to an optimization metric.

Strategies

- Direct
- Iterative

- Principal Component Analysis
- Support Vector Machines

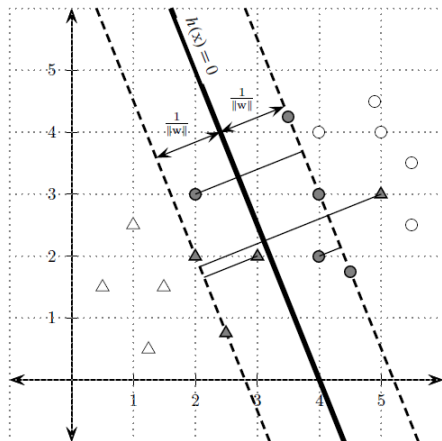


Figure 21.3. Soft margin hyperplane: the shaded points are the support vectors. The margin is $1/\|w\|$ as illustrated, and points with positive slack values are also shown (thin black line).

Paradigms

- Combinatorial
- Probabilistic
- Algebraic
- **Graph-based**

Domain

Input data is modeled as a graph, enabling not just richer representations but also several existing models and algorithms.

Task

Determine the best representation and technique, according to an optimization metric.

Challenge

How can we handle the larger complexity and numerosity induced by graphs?

- Frequent Subgraph Mining
- Spectral Clustering

Graph-based Models

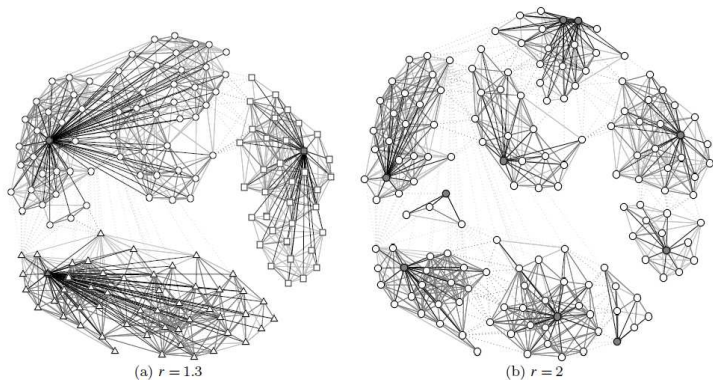


Figure 16.6. MCL on Iris graph.

A massively connected world produces

- huge,
- incomplete,
- noisy,
- heterogeneous and
- asynchronous streams of data

Challenges for Data Mining

- Heterogeneous data
- Incomplete information
- Noisy data
- Dynamic behavior
- Complex relationships
- Lack of scalability

Heterogeneous Data

- Types of data
 - numerical, categorical, spatial, temporal, relations
- Characteristics
 - Variable density and representativity
 - Variable granularity
 - Best abstraction level?

- CPS Data are always incomplete
 - Not measured
 - Not measured frequently enough
- Issues
 - How much data is enough?
 - How to fill missing data?
 - How to augment data?
 - How to infer more complex behaviors?

- Data is not accurate as a consequence of measurement issues
- Issues:
 - Noise vs. outliers
 - Noise reduction (information loss?)
 - Noise tolerance (e.g., probabilistic models)

- Data mining models usually assume that the past will occur in the future
- Drift is a common phenomena:
 - Temporal
 - Spatial
 - Environmental
 - Entity-related characteristics and habits
- Tasks
 - Identify drifts
 - Handle drifts

Complex Relationships

- SCP systems comprise a large spectrum of entities and associated relationships
- Model issues:
 - Explicit vs. Implicit relationships
 - Home address and Income vs. Home address and Safety
 - Directed vs. Undirected relationships
 - weather \rightarrow traffic jam vs. traffic jam \nrightarrow weather
- Mining issues:
 - Significance/similarity measures
 - Complex patterns

- Algorithmical enhancement
 - it is not simple nor usual
- Parallelization
 - algorithms are usually irregular and I/O intensive.
- Sampling
 - fairness
 - representativity

Requirements for Models and Techniques

- Transferability: how can we transfer knowledge among domains?
- Fairness: how can we avoid discrimination?
- Transparency: how can we understand the models and the outcomes?
- Accountability: who is responsible for any damage?

- Data management: several solutions either general or specialized for all kinds of data
- Data mining: several implementations of each technique
- User demands: does the data scientist need to program?
 - NO! He or she needs to think algorithmically.

Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere

Enablers:

- Wide availability of algorithm implementations
- Broad spectrum of databases and storage technologies
- Massively parallel processing commercial solutions
- Mature virtualization technology
- Real time transpiling technology is a reality
- Awareness of the data potential

Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere

Motivations

- Data analysts do not need to program, literally
- Data analysts need to abstract algorithmically tasks
- Cloud-fashion web-based platforms provide good interactive support
- Visual programming is a need

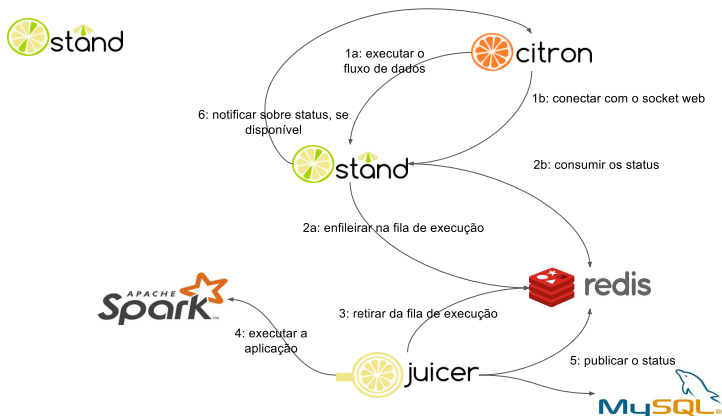
Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere



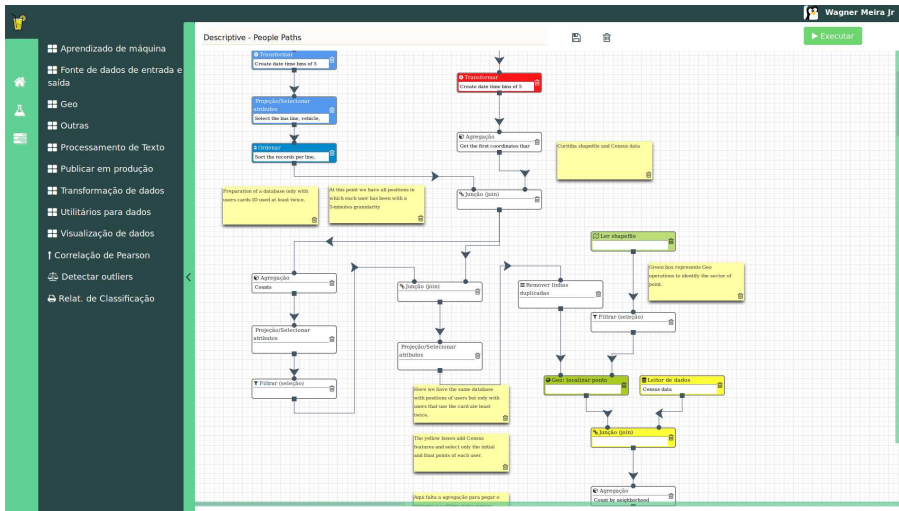
Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere



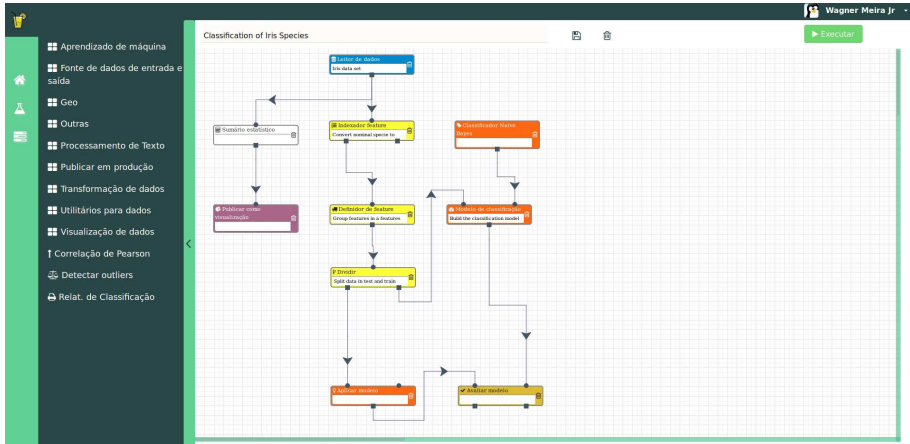
Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere



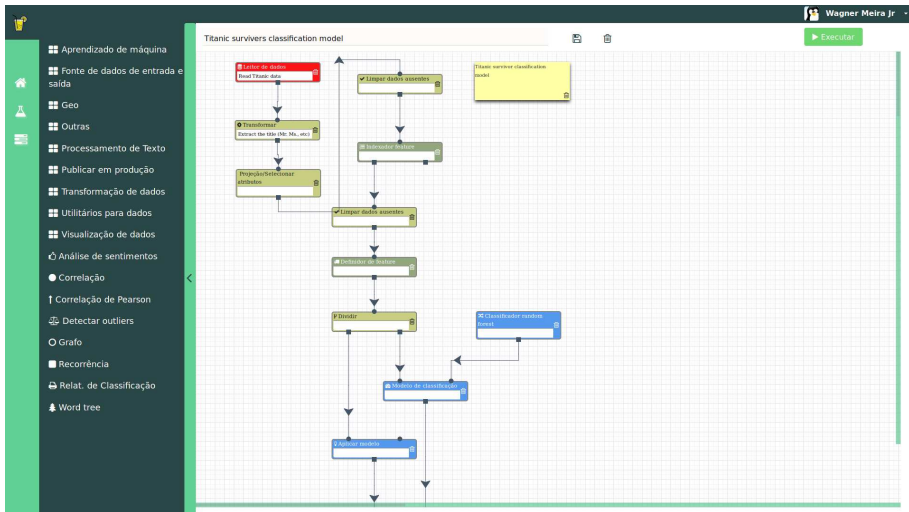
Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere



Lemonade

Live Exploration and Mining of a Non-trivial Amount of Data from Everywhere



Representative-based Clustering

Given a dataset with n points in a d -dimensional space, $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^n$, and given the number of desired clusters k , the goal of representative-based clustering is to partition the dataset into k groups or clusters, which is called a *clustering* and is denoted as $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$.

For each cluster C_i there exists a representative point that summarizes the cluster, a common choice being the mean (also called the *centroid*) μ_i of all points in the cluster, that is,

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} \mathbf{x}_j$$

where $n_i = |C_i|$ is the number of points in cluster C_i .

A brute-force or exhaustive algorithm for finding a good clustering is simply to generate all possible partitions of n points into k clusters, evaluate some optimization score for each of them, and retain the clustering that yields the best score. However, this is clearly infeasible, since there are $O(k^n/k!)$ clusterings of n points into k groups.

K-means Algorithm: Objective

The *sum of squared errors* scoring function is defined as

$$SSE(\mathcal{C}) = \sum_{i=1}^k \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

The goal is to find the clustering that minimizes the SSE score:

$$\mathcal{C}^* = \operatorname{argmin}_{\mathcal{C}} \{SSE(\mathcal{C})\}$$

K-means employs a greedy iterative approach to find a clustering that minimizes the SSE objective. As such it can converge to a local optima instead of a globally optimal clustering.

K-means Algorithm: Objective

K-means initializes the cluster means by randomly generating k points in the data space. Each iteration of K-means consists of two steps: (1) cluster assignment, and (2) centroid update.

Given the k cluster means, in the cluster assignment step, each point $\mathbf{x}_j \in \mathbf{D}$ is assigned to the closest mean, which induces a clustering, with each cluster C_i comprising points that are closer to $\boldsymbol{\mu}_i$ than any other cluster mean. That is, each point \mathbf{x}_j is assigned to cluster C_{j^*} , where

$$j^* = \arg \min_{i=1}^k \left\{ \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2 \right\}$$

Given a set of clusters C_i , $i = 1, \dots, k$, in the centroid update step, new mean values are computed for each cluster from the points in C_j .

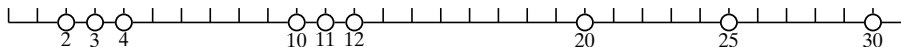
The cluster assignment and centroid update steps are carried out iteratively until we reach a fixed point or local minima.

K-Means Algorithm

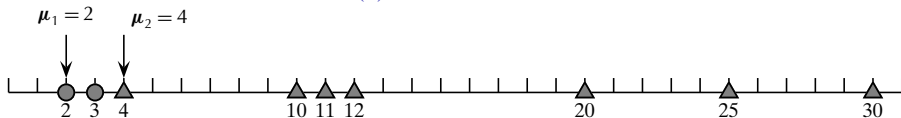
K-MEANS (\mathbf{D} , k , ϵ):

```
1  $t = 0$ 
2 Randomly initialize  $k$  centroids:  $\mu_1^t, \mu_2^t, \dots, \mu_k^t \in \mathbb{R}^d$ 
3 repeat
4    $t \leftarrow t + 1$ 
5    $C_j \leftarrow \emptyset$  for all  $j = 1, \dots, k$ 
   // Cluster Assignment Step
6   foreach  $\mathbf{x}_j \in \mathbf{D}$  do
7      $j^* \leftarrow \operatorname{argmin}_j \left\{ \|\mathbf{x}_j - \mu_j^t\|^2 \right\}$  // Assign  $\mathbf{x}_j$  to closest centroid
8      $C_{j^*} \leftarrow C_{j^*} \cup \{\mathbf{x}_j\}$ 
   // Centroid Update Step
9   foreach  $i = 1$  to  $k$  do
10     $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$ 
11 until  $\sum_{i=1}^k \|\mu_i^t - \mu_i^{t-1}\|^2 \leq \epsilon$ 
```

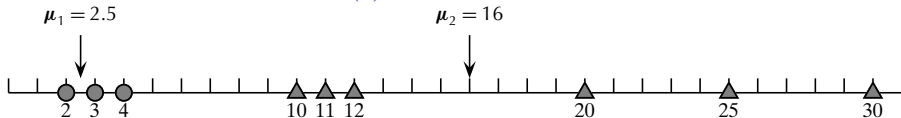
K-means in One Dimension



(a) Initial dataset

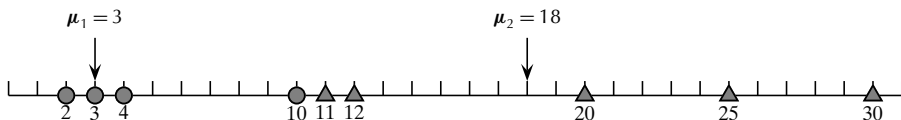


(b) Iteration: $t = 1$



(c) Iteration: $t = 2$

K-means in One Dimension (contd.)



(d) Iteration: $t = 3$

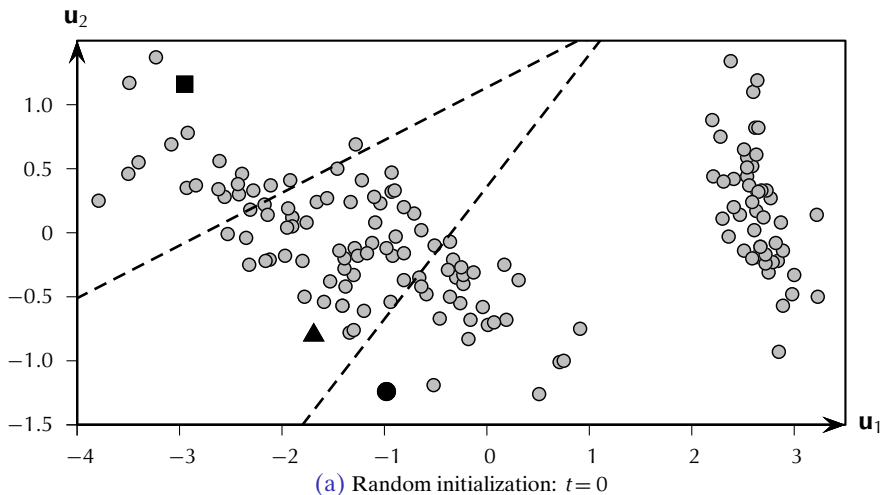


(e) Iteration: $t = 4$

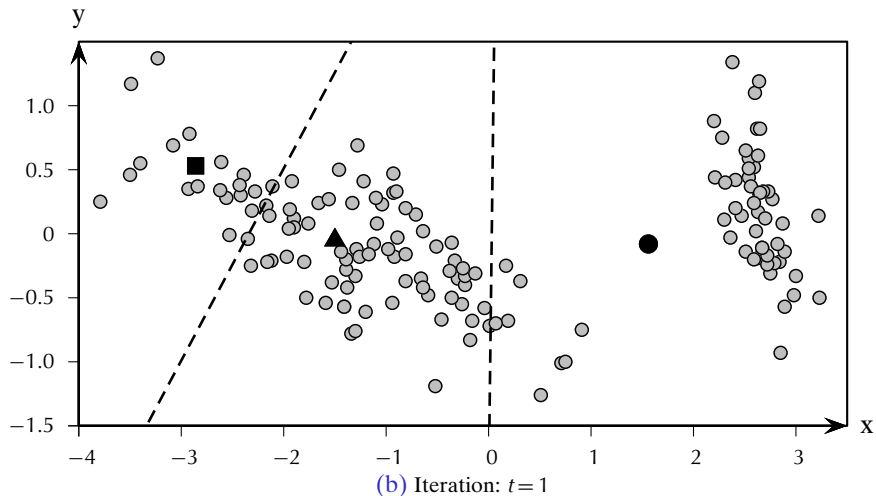


(f) Iteration: $t = 5$ (converged)

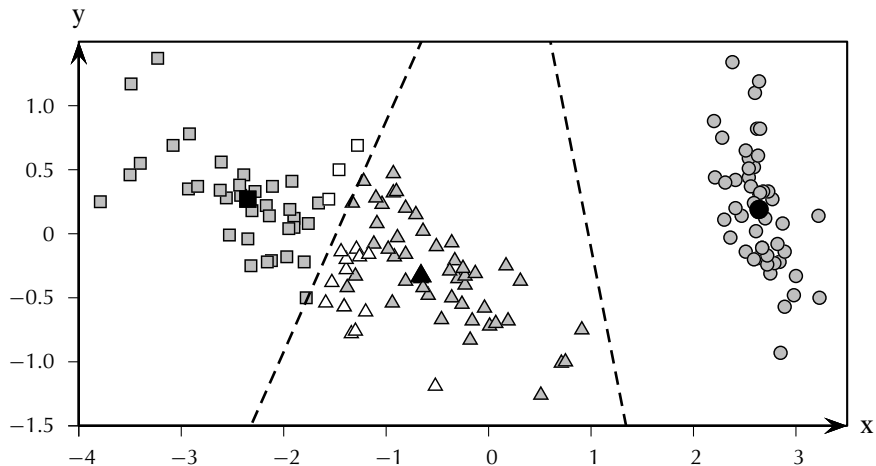
K-means in 2D: Iris Principal Components



K-means in 2D: Iris Principal Components



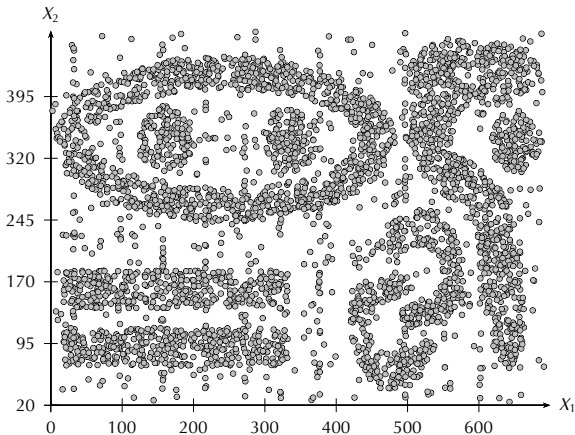
K-means in 2D: Iris Principal Components



(c) Iteration: $t = 8$ (converged)

Density-based Clustering

Density-based methods are able to mine nonconvex clusters, where distance-based methods may have difficulty.



The DBSCAN Approach: Neighborhood and Core Points

Define a ball of radius ϵ around a point $\mathbf{x} \in \mathbb{R}^d$, called the ϵ -neighborhood of \mathbf{x} , as follows:

$$N_\epsilon(\mathbf{x}) = B_d(\mathbf{x}, \epsilon) = \{\mathbf{y} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$$

Here $\delta(\mathbf{x}, \mathbf{y})$ represents the distance between points \mathbf{x} and \mathbf{y} . which is usually assumed to be the Euclidean

We say that \mathbf{x} is a *core point* if there are at least *minpts* points in its ϵ -neighborhood, i.e., if $|N_\epsilon(\mathbf{x})| \geq \text{minpts}$.

A *border point* does not meet the *minpts* threshold, i.e., $|N_\epsilon(\mathbf{x})| < \text{minpts}$, but it belongs to the ϵ -neighborhood of some core point \mathbf{z} , that is, $\mathbf{x} \in N_\epsilon(\mathbf{z})$.

If a point is neither a core nor a border point, then it is called a *noise point* or an outlier.

The DBSCAN Approach: Reachability and Density-based Cluster

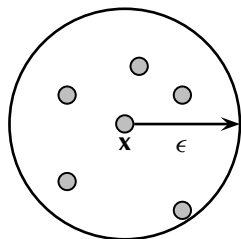
A point \mathbf{x} is *directly density reachable* from another point \mathbf{y} if $\mathbf{x} \in N_\epsilon(\mathbf{y})$ and \mathbf{y} is a core point.

A point \mathbf{x} is *density reachable* from \mathbf{y} if there exists a chain of points, $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_l$, such that $\mathbf{x} = \mathbf{x}_0$ and $\mathbf{y} = \mathbf{x}_l$, and \mathbf{x}_i is directly density reachable from \mathbf{x}_{i-1} for all $i = 1, \dots, l$. In other words, there is set of core points leading from \mathbf{y} to \mathbf{x} .

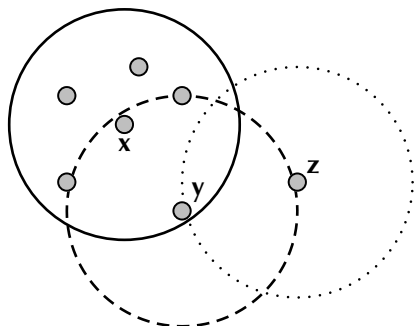
Two points \mathbf{x} and \mathbf{y} are *density connected* if there exists a core point \mathbf{z} , such that both \mathbf{x} and \mathbf{y} are density reachable from \mathbf{z} .

A *density-based cluster* is defined as a maximal set of density connected points.

Core, Border and Noise Points



(d) Neighborhood of a Point



(e) Core, Border, and Noise Points

DBSCAN Density-based Clustering Algorithm

DBSCAN computes the ϵ -neighborhood $N_\epsilon(\mathbf{x}_i)$ for each point \mathbf{x}_i in the dataset \mathbf{D} , and checks if it is a core point. It also sets the cluster id $id(\mathbf{x}_i) = \emptyset$ for all points, indicating that they are not assigned to any cluster.

Starting from each unassigned core point, the method recursively finds all its density connected points, which are assigned to the same cluster.

Some border point may be reachable from core points in more than one cluster; they may either be arbitrarily assigned to one of the clusters or to all of them (if overlapping clusters are allowed).

Those points that do not belong to any cluster are treated as outliers or noise.

Each DBSCAN cluster is a maximal connected component over the core point graph.

DBSCAN is sensitive to the choice of ϵ , in particular if clusters have different densities. The overall complexity of DBSCAN is $O(n^2)$.

DBSCAN Algorithm

DBSCAN (\mathbf{D} , ϵ , $minpts$):

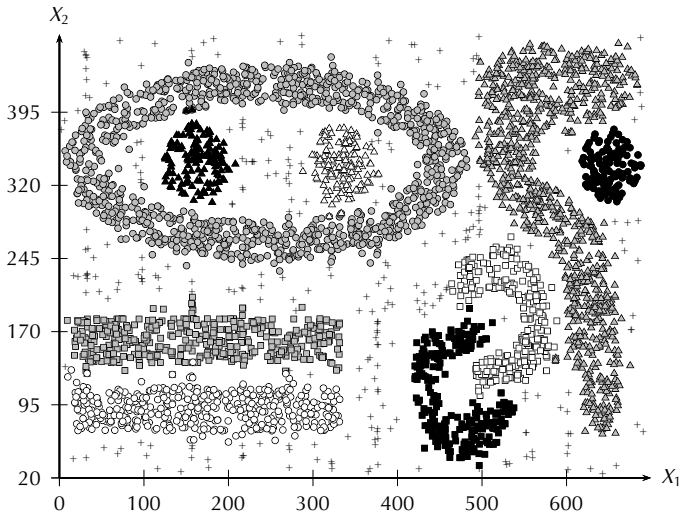
```
1 Core  $\leftarrow \emptyset$ 
2 foreach  $\mathbf{x}_i \in \mathbf{D}$  do // Find the core points
3   Compute  $N_\epsilon(\mathbf{x}_i)$ 
4    $id(\mathbf{x}_i) \leftarrow \emptyset$  // cluster id for  $\mathbf{x}_i$ 
5   if  $N_\epsilon(\mathbf{x}_i) \geq minpts$  then Core  $\leftarrow$  Core  $\cup \{\mathbf{x}_i\}$ 
6  $k \leftarrow 0$  // cluster id
7 foreach  $\mathbf{x}_i \in$  Core, such that  $id(\mathbf{x}_i) = \emptyset$  do
8    $k \leftarrow k + 1$ 
9    $id(\mathbf{x}_i) \leftarrow k$  // assign  $\mathbf{x}_i$  to cluster id  $k$ 
10  DENSITYCONNECTED ( $\mathbf{x}_i, k$ )
11  $\mathcal{C} \leftarrow \{C_i\}_{i=1}^k$ , where  $C_i \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = i\}$ 
12 Noise  $\leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = \emptyset\}$ 
13 Border  $\leftarrow \mathbf{D} \setminus \{Core \cup Noise\}$ 
14 return  $\mathcal{C}$ , Core, Border, Noise
```

DENSITYCONNECTED (\mathbf{x} , k):

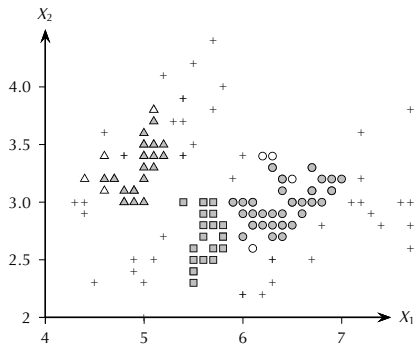
```
15 foreach  $\mathbf{y} \in N_\epsilon(\mathbf{x})$  do
16    $id(\mathbf{y}) \leftarrow k$  // assign  $\mathbf{y}$  to cluster id  $k$ 
17   if  $\mathbf{y} \in$  Core then DENSITYCONNECTED ( $\mathbf{y}, k$ )
```

Density-based Clusters

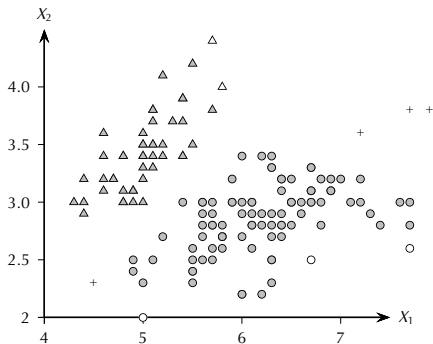
$\epsilon = 15$ and $minpts = 10$



DBSCAN Clustering: Iris Dataset



(a) $\epsilon = 0.2$, $minpts = 5$



(b) $\epsilon = 0.36$, $minpts = 3$

It is the same of DBSCAN, except for the similarity metric:

- Spatial threshold: distance
- Temporal threshold: offset in minutes

You may create any kind of neighborhood (e.g., topic). There is not really integration.

Kernel Density Estimation

There is a close connection between density-based clustering and density estimation. The goal of density estimation is to determine the unknown probability density function by finding the dense regions of points, which can in turn be used for clustering.

Kernel density estimation is a nonparametric technique that does not assume any fixed probability model of the clusters. Instead, it tries to directly infer the underlying probability density at each point in the dataset.

Univariate Density Estimation

Assume that X is a continuous random variable, and let x_1, x_2, \dots, x_n be a random sample. We directly estimate the cumulative distribution function from the data by counting how many points are less than or equal to x :

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n I(x_i \leq x)$$

where I is an indicator function.

We estimate the density function by taking the derivative of $\hat{F}(x)$

$$\hat{f}(x) = \frac{\hat{F}(x + \frac{h}{2}) - \hat{F}(x - \frac{h}{2})}{h} = \frac{k/n}{h} = \frac{k}{nh}$$

where k is the number of points that lie in the window of width h centered at x . The density estimate is the ratio of the fraction of the points in the window (k/n) to the volume of the window (h).

Kernel density estimation relies on a *kernel function* K that is non-negative, symmetric, and integrates to 1, that is, $K(x) \geq 0$, $K(-x) = K(x)$ for all values x , and $\int K(x)dx = 1$.

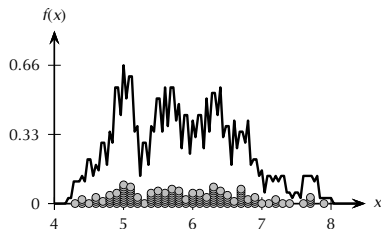
Discrete Kernel Define the **discrete kernel** function K , that computes the number of points in a window of width h

$$K(z) = \begin{cases} 1 & \text{If } |z| \leq \frac{1}{2} \\ 0 & \text{Otherwise} \end{cases}$$

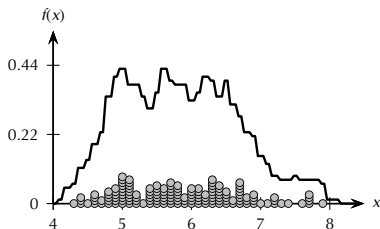
The density estimate $\hat{f}(x)$ can be rewritten in terms of the kernel function as follows:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

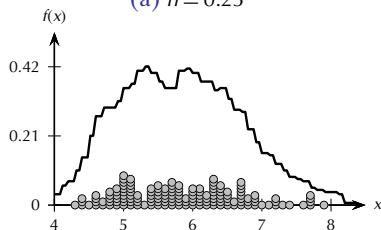
Kernel Density Estimation: Discrete Kernel (Iris 1D)



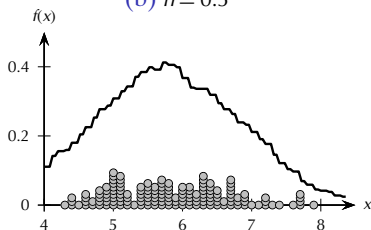
(a) $h = 0.25$



(b) $h = 0.5$



(c) $h = 1.0$



(d) $h = 2.0$

The discrete kernel yields a non-smooth (or jagged) density function.

Kernel Density Estimation: Gaussian Kernel

The width h is a parameter that denotes the spread or smoothness of the density estimate. The discrete kernel function has an abrupt influence.

Define a more smooth transition of influence via a Gaussian kernel:

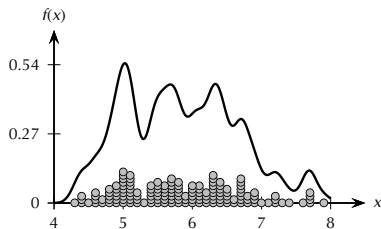
$$K(z) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{z^2}{2}\right\}$$

Thus, we have

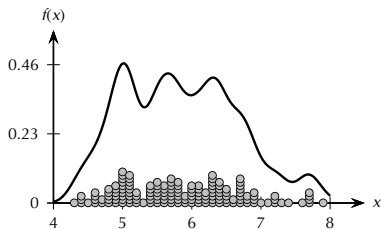
$$K\left(\frac{x-x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x-x_i)^2}{2h^2}\right\}$$

Here x , which is at the center of the window, plays the role of the mean, and h acts as the standard deviation.

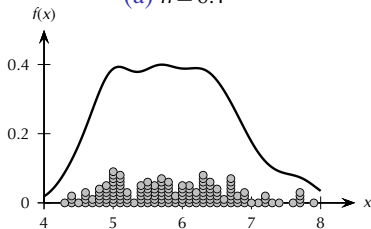
Kernel Density Estimation: Gaussian Kernel (Iris 1D)



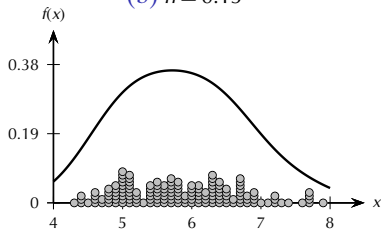
(a) $h = 0.1$



(b) $h = 0.15$



(c) $h = 0.25$



(d) $h = 0.5$

When h is small the density function has many local maxima. A large h results in a unimodal distribution.

Multivariate Density Estimation

To estimate the probability density at a d -dimensional point $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$, we define the d -dimensional “window” as a hypercube in d dimensions, that is, a hypercube centered at \mathbf{x} with edge length h . The volume of such a d -dimensional hypercube is given as

$$\text{vol}(H_d(h)) = h^d$$

The density is estimated as the fraction of the point weight lying within the d -dimensional window centered at \mathbf{x} , divided by the volume of the hypercube:

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

where the multivariate kernel function K satisfies the condition $\int K(\mathbf{z}) d\mathbf{z} = 1$.

Multivariate Density Estimation: Discrete and Gaussian Kernel

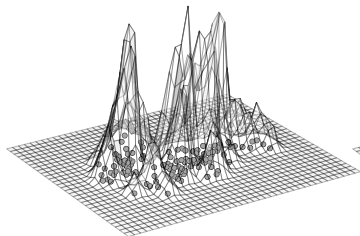
Discrete Kernel: For any d -dimensional vector $\mathbf{z} = (z_1, z_2, \dots, z_d)^T$, the discrete kernel function in d -dimensions is given as

$$K(\mathbf{z}) = \begin{cases} 1 & \text{If } |z_j| \leq \frac{1}{2}, \text{ for all dimensions } j = 1, \dots, d \\ 0 & \text{Otherwise} \end{cases}$$

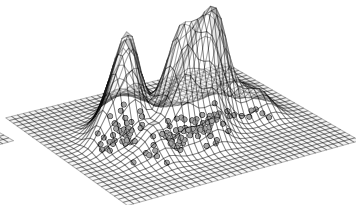
Gaussian Kernel: The d -dimensional Gaussian kernel is given as

$$K(\mathbf{z}) = \frac{1}{(2\pi)^{d/2}} \exp \left\{ -\frac{\mathbf{z}^T \mathbf{z}}{2} \right\}$$

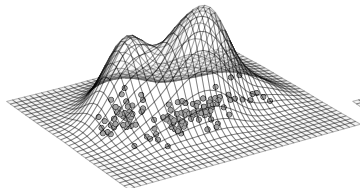
Density Estimation: Iris 2D Data (Gaussian Kernel)



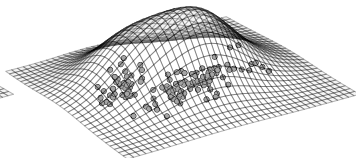
(a) $h = 0.1$



(b) $h = 0.2$



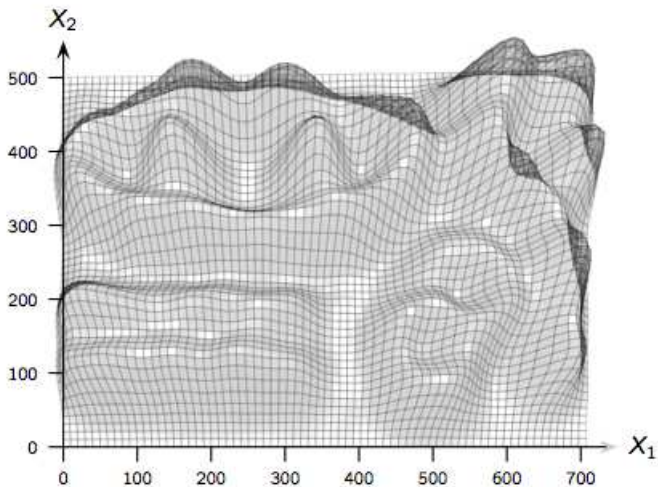
(c) $h = 0.35$



(d) $h = 0.6$

Density Estimation: Density-based Dataset

Gaussian kernel, $h = 20$



Nearest Neighbor Density Estimation

In kernel density estimation we implicitly fixed the volume by fixing the width h , and we used the kernel function to find out the number or weight of points that lie inside the fixed volume region.

An alternative approach to density estimation is to fix k , the number of points required to estimate the density, and allow the volume of the enclosing region to vary to accommodate those k points. This approach is called the k nearest neighbors (KNN) approach to density estimation.

Given k , the number of neighbors, we estimate the density at \mathbf{x} as follows:

$$\hat{f}(\mathbf{x}) = \frac{k}{n \text{vol}(S_d(h_{\mathbf{x}}))}$$

where $h_{\mathbf{x}}$ is the distance from \mathbf{x} to its k th nearest neighbor, and $\text{vol}(S_d(h_{\mathbf{x}}))$ is the volume of the d -dimensional hypersphere $S_d(h_{\mathbf{x}})$ centered at \mathbf{x} , with radius $h_{\mathbf{x}}$.

DENCLUE Density-based Clustering: Attractor and Gradient

A point \mathbf{x}^* is called a *density attractor* if it is a local maxima of the probability density function f .

The density gradient at a point \mathbf{x} is the multivariate derivative of the probability density estimate

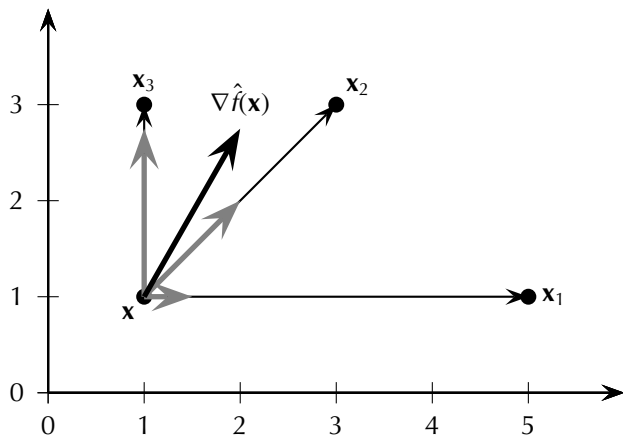
$$\nabla \hat{f}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n \frac{\partial}{\partial \mathbf{x}} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

For the Gaussian kernel the gradient at a point \mathbf{x} is given as

$$\nabla \hat{f}(\mathbf{x}) = \frac{1}{nh^{d+2}} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \cdot (\mathbf{x}_i - \mathbf{x})$$

This equation can be thought of as having two parts for each point: a vector $(\mathbf{x}_i - \mathbf{x})$ and a scalar *influence* value $K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$.

The Gradient Vector



DENCLUE: Density Attractor

We say that \mathbf{x}^* is a *density attractor* for \mathbf{x} , or alternatively that \mathbf{x} is *density attracted* to \mathbf{x}^* , if a hill climbing process started at \mathbf{x} converges to \mathbf{x}^* .

That is, there exists a sequence of points $\mathbf{x} = \mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \dots \rightarrow \mathbf{x}_m$, starting from \mathbf{x} and ending at \mathbf{x}_m , such that $\|\mathbf{x}_m - \mathbf{x}^*\| \leq \epsilon$, that is, \mathbf{x}_m converges to the attractor \mathbf{x}^* .

Setting the gradient to the zero vector leads to the following *mean-shift* update rule:

$$\mathbf{x}_{t+1} = \frac{\sum_{i=1}^n K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \mathbf{x}_i}{\sum_{i=1}^n K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$$

where t denotes the current iteration and \mathbf{x}_{t+1} is the updated value for the current vector \mathbf{x}_t .

A cluster $C \subseteq \mathbf{D}$, is called a *center-defined cluster* if all the points $\mathbf{x} \in C$ are density attracted to a unique density attractor \mathbf{x}^* , such that $\hat{f}(\mathbf{x}^*) \geq \xi$, where ξ is a user-defined minimum density threshold.

An arbitrary-shaped cluster $C \subseteq \mathbf{D}$ is called a *density-based cluster* if there exists a set of density attractors $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_m^*$, such that

- 1 Each point $\mathbf{x} \in C$ is attracted to some attractor \mathbf{x}_i^* .
- 2 Each density attractor has density above ξ .
- 3 Any two density attractors \mathbf{x}_i^* and \mathbf{x}_j^* are *density reachable*, that is, there exists a path from \mathbf{x}_i^* to \mathbf{x}_j^* , such that for all points \mathbf{y} on the path, $\hat{f}(\mathbf{y}) \geq \xi$.

The DENCLUE Algorithm

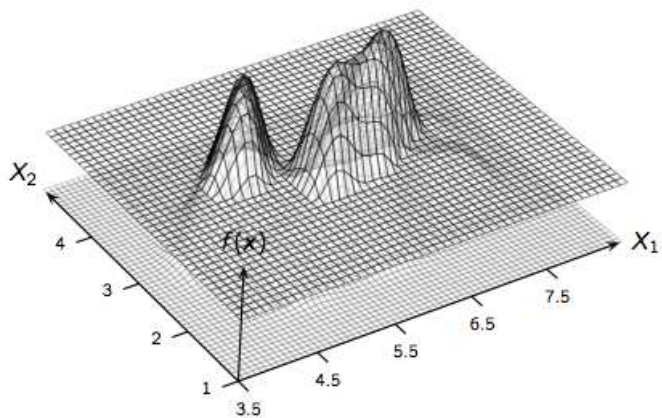
```
DENCLUE ( $\mathbf{D}, h, \xi, \epsilon$ ):  
1  $\mathcal{A} \leftarrow \emptyset$   
2 foreach  $\mathbf{x} \in \mathbf{D}$  do // find density attractors  
3    $\mathbf{x}^* \leftarrow \text{FINDATTRACTOR}(\mathbf{x}, \mathbf{D}, h, \epsilon)$   
4   if  $\hat{f}(\mathbf{x}^*) \geq \xi$  then  
5      $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{x}^*\}$   
6      $R(\mathbf{x}^*) \leftarrow R(\mathbf{x}^*) \cup \{\mathbf{x}\}$   
7  
8  
9  
10  $\mathcal{C} \leftarrow \{\text{maximal } C \subseteq \mathcal{A} \mid \forall \mathbf{x}_i^*, \mathbf{x}_j^* \in C, \mathbf{x}_i^* \text{ and } \mathbf{x}_j^* \text{ are density reachable}\}$   
11 foreach  $C \in \mathcal{C}$  do // density-based clusters  
12   foreach  $\mathbf{x}^* \in C$  do  $C \leftarrow C \cup R(\mathbf{x}^*)$   
13  
14 return  $\mathcal{C}$ 
```


The DENCLUE Algorithm: Find Attractor

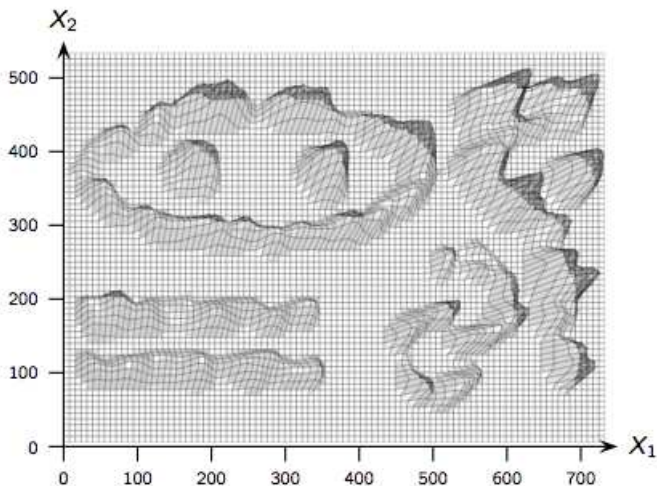
FINDATTRACTOR ($\mathbf{x}, \mathbf{D}, h, \epsilon$):

```
2  $t \leftarrow 0$ 
3  $\mathbf{x}_t \leftarrow \mathbf{x}$ 
4 repeat
5      $\mathbf{x}_{t+1} \leftarrow \frac{\sum_{i=1}^n K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_i}{\sum_{i=1}^n K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$ 
6      $t \leftarrow t + 1$ 
7 until  $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$ 
8 return  $\mathbf{x}_t$ 
```

DENCLUE: Iris 2D Data



DENCLUE: Density-based Dataset



Clustering fragmented trajectories

Given that

- u_1, u_2, \dots, u_n be individuals.
- N_{u_i} be the number of points collected for individual u_i , that is, the length.
- $p_{1_{u_i}}, p_{2_{u_i}}, \dots, p_{N_{u_i}}$ the points from individual u_i .

The problem of determining *fragmented or noncontiguous clusters* consists of finding the groups of individuals such that those in the same group present similar point density over the sampling space and those in different groups present different densities.

Premises:

- the locations that an individual visits do not have to be contiguous and
- the densities considered must take into account the relative number of points from an individual in a given location or region.

Clustering fragmented trajectories

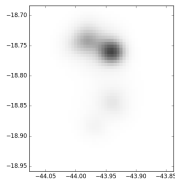
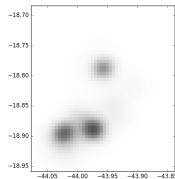
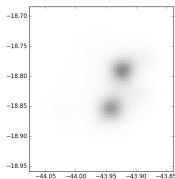
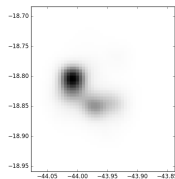
- Challenge: how to cluster people based on parts of the trajectories?
- Rationale: In order to cluster objects by their spatial patterns, we assume that the observed points of the objects that belong to the same cluster were generated by the same process, a Poisson point process. Therefore, we are able to calculate the likelihood of an object of having been generated by this implicit process that rules a cluster.

In our method, we represent each cluster not as a centroid as in K-means, but as a Poisson process, with its intensity in the two-dimensional space. The log-likelihood of an object to belong to a cluster, for the Poisson process, is given by:

$$\sum_{i=1}^N \log \lambda(x_i, y_i) - \int_B \lambda(x, y) dx dy \quad (1)$$

where N is the number of observed positions, w_i is the weight for the position \mathbf{x}_i , and W is equal to $\sum_{i=1}^N w_i$.

Clustering results for Belo Horizonte.



Site: <https://sc.ctweb.inweb.org.br/>

User: aluno_X ($1 \leq X \leq 40$)

Password: sm4rt.Citi3Z

Bayes Classifier

Let the training dataset \mathbf{D} consist of n points \mathbf{x}_i in a d -dimensional space, and let y_i denote the class for each point, with $y_i \in \{c_1, c_2, \dots, c_k\}$.

The Bayes classifier estimates the posterior probability $P(c_i|\mathbf{x})$ for each class c_i , and chooses the class that has the largest probability. The predicted class for \mathbf{x} is given as

$$\hat{y} = \arg \max_{c_i} \{P(c_i|\mathbf{x})\}$$

According to the Bayes theorem, we have

$$P(c_i|\mathbf{x}) = \frac{P(\mathbf{x}|c_i) \cdot P(c_i)}{P(\mathbf{x})}$$

Because $P(\mathbf{x})$ is fixed for a given point, Bayes rule can be rewritten as

$$\hat{y} = \arg \max_{c_i} \{P(c_i|\mathbf{x})\} = \arg \max_{c_i} \left\{ \frac{P(\mathbf{x}|c_i)P(c_i)}{P(\mathbf{x})} \right\} = \arg \max_{c_i} \{P(\mathbf{x}|c_i)P(c_i)\}$$

Estimating the Prior Probability: $P(c_i)$

Let \mathbf{D}_i denote the subset of points in \mathbf{D} that are labeled with class c_i :

$$\mathbf{D}_i = \{\mathbf{x}_j \in \mathbf{D} \mid \mathbf{x}_j \text{ has class } y_j = c_i\}$$

Let the size of the dataset \mathbf{D} be given as $|\mathbf{D}| = n$, and let the size of each class-specific subset \mathbf{D}_i be given as $|\mathbf{D}_i| = n_i$.

The prior probability for class c_i can be estimated as follows:

$$\hat{P}(c_i) = \frac{n_i}{n}$$

Estimating the Likelihood: Numeric Attributes, Parametric Approach

To estimate the likelihood $P(\mathbf{x}|c_i)$, we have to estimate the joint probability of \mathbf{x} across all the d dimensions, i.e., we have to estimate $P(\mathbf{x} = (x_1, x_2, \dots, x_d)|c_i)$.

In the parametric approach we assume that each class c_i is normally distributed, and we use the estimated mean $\hat{\boldsymbol{\mu}}_i$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_i$ to compute the probability density at \mathbf{x}

$$\hat{f}_i(\mathbf{x}) = \hat{f}(\mathbf{x}|\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) = \frac{1}{(\sqrt{2\pi})^d \sqrt{|\hat{\boldsymbol{\Sigma}}_i|}} \exp \left\{ -\frac{(\mathbf{x} - \hat{\boldsymbol{\mu}}_i)^T \hat{\boldsymbol{\Sigma}}_i^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_i)}{2} \right\}$$

The posterior probability is then given as

$$P(c_i|\mathbf{x}) = \frac{\hat{f}_i(\mathbf{x})P(c_i)}{\sum_{j=1}^k \hat{f}_j(\mathbf{x})P(c_j)}$$

The predicted class for \mathbf{x} is:

$$\hat{y} = \arg \max_{c_i} \left\{ \hat{f}_i(\mathbf{x})P(c_i) \right\}$$

Bayes Classifier Algorithm

BAYESCLASSIFIER ($\mathbf{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^n$):

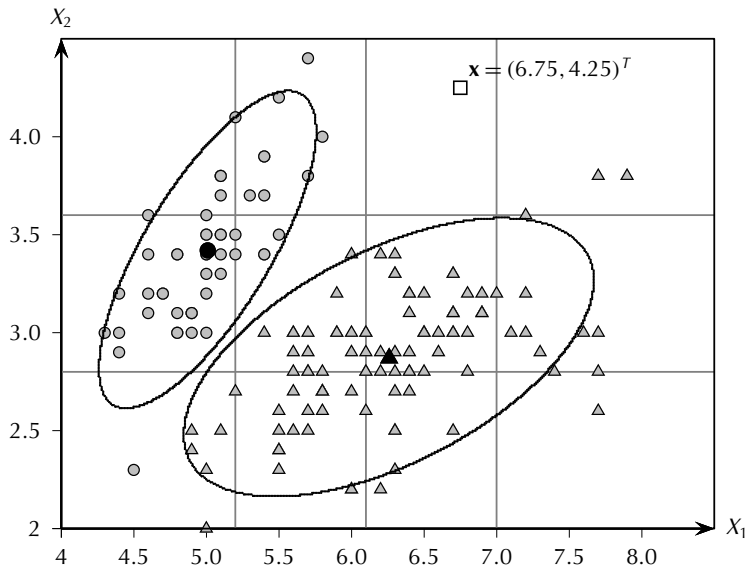
```
1 for  $i = 1, \dots, k$  do
2    $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \dots, n\}$  // class-specific subsets
3    $n_i \leftarrow |\mathbf{D}_i|$  // cardinality
4    $\hat{P}(c_i) \leftarrow n_i/n$  // prior probability
5    $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$  // mean
6    $\mathbf{Z}_i \leftarrow \mathbf{D}_i - \mathbf{1}_{n_i} \hat{\boldsymbol{\mu}}_i^T$  // centered data
7    $\hat{\boldsymbol{\Sigma}}_i \leftarrow \frac{1}{n_i} \mathbf{Z}_i^T \mathbf{Z}_i$  // covariance matrix
8 return  $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i$  for all  $i = 1, \dots, k$ 
```

TESTING (\mathbf{x} and $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i$, for all $i \in [1, k]$):

```
9  $\hat{y} \leftarrow \underset{c_i}{\operatorname{argmax}} \{f(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i) \cdot P(c_i)\}$ 
10 return  $\hat{y}$ 
```

Bayes Classifier: Iris Data

X_1 : sepal length versus X_2 : sepal width



Bayes Classifier: Categorical Attributes

Let X_j be a categorical attribute over the domain $dom(X_j) = \{a_{j1}, a_{j2}, \dots, a_{jm_j}\}$. Each categorical attribute X_j is modeled as an m_j -dimensional multivariate Bernoulli random variable \mathbf{X}_j that takes on m_j distinct vector values $\mathbf{e}_{j1}, \mathbf{e}_{j2}, \dots, \mathbf{e}_{jm_j}$, where \mathbf{e}_{jr} is the r th standard basis vector in \mathbb{R}^{m_j} and corresponds to the r th value or symbol $a_{jr} \in dom(X_j)$.

The entire d -dimensional dataset is modeled as the vector random variable $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_d)^T$. Let $d' = \sum_{j=1}^d m_j$; a categorical point $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ is therefore represented as the d' -dimensional binary vector

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_d \end{pmatrix} = \begin{pmatrix} \mathbf{e}_{1r_1} \\ \vdots \\ \mathbf{e}_{dr_d} \end{pmatrix}$$

where $\mathbf{v}_j = \mathbf{e}_{jr_j}$ provided $x_j = a_{jr_j}$ is the r_j th value in the domain of X_j .

Bayes Classifier: Categorical Attributes

The probability of the categorical point \mathbf{x} is obtained from the joint probability mass function (PMF) for the vector random variable \mathbf{X} :

$$P(\mathbf{x}|c_i) = f(\mathbf{v}|c_i) = f(\mathbf{X}_1 = \mathbf{e}_{1r_1}, \dots, \mathbf{X}_d = \mathbf{e}_{dr_d} | c_i)$$

The joint PMF can be estimated directly from the data \mathbf{D}_i for each class c_i as follows:

$$\hat{f}(\mathbf{v}|c_i) = \frac{n_i(\mathbf{v})}{n_i}$$

where $n_i(\mathbf{v})$ is the number of times the value \mathbf{v} occurs in class c_i .

However, to avoid zero probabilities we add a *pseudo-count* of 1 for each value

$$\hat{f}(\mathbf{v}|c_i) = \frac{n_i(\mathbf{v}) + 1}{n_i + \prod_{j=1}^d m_j}$$

Discretized Iris Data: sepal length and sepal width

Bins	Domain
[4.3, 5.2]	Very Short (a_{11})
(5.2, 6.1]	Short (a_{12})
(6.1, 7.0]	Long (a_{13})
(7.0, 7.9]	Very Long (a_{14})

(a) Discretized sepal length

Bins	Domain
[2.0, 2.8]	Short (a_{21})
(2.8, 3.6]	Medium (a_{22})
(3.6, 4.4]	Long (a_{23})

(b) Discretized sepal width

Class-specific Empirical Joint Probability Mass Function

Class: c_1		X_2			\hat{f}_{X_1}
		Short (e_{21})	Medium (e_{22})	Long (e_{23})	
X_1	Very Short (e_{11})	1/50	33/50	5/50	39/50
	Short (e_{12})	0	3/50	8/50	13/50
	Long (e_{13})	0	0	0	0
	Very Long (e_{14})	0	0	0	0
\hat{f}_{X_2}		1/50	36/50	13/50	

Class: c_2		X_2			\hat{f}_{X_1}
		Short (e_{21})	Medium (e_{22})	Long (e_{23})	
X_1	Very Short (e_{11})	6/100	0	0	6/100
	Short (e_{12})	24/100	15/100	0	39/100
	Long (e_{13})	13/100	30/100	0	43/100
	Very Long (e_{14})	3/100	7/100	2/100	12/100
\hat{f}_{X_2}		46/100	52/100	2/100	

Consider a test point $\mathbf{x} = (5.3, 3.0)^T$ corresponding to the categorical point (Short, Medium), which is represented as $\mathbf{v} = (\mathbf{e}_{12}^T \quad \mathbf{e}_{22}^T)^T$.

The prior probabilities of the classes are $\hat{P}(c_1) = 0.33$ and $\hat{P}(c_2) = 0.67$. The likelihood and posterior probability for each class is given as

$$\hat{P}(\mathbf{x}|c_1) = \hat{f}(\mathbf{v}|c_1) = 3/50 = 0.06$$

$$\hat{P}(\mathbf{x}|c_2) = \hat{f}(\mathbf{v}|c_2) = 15/100 = 0.15$$

$$\hat{P}(c_1|\mathbf{x}) \propto 0.06 \times 0.33 = 0.0198$$

$$\hat{P}(c_2|\mathbf{x}) \propto 0.15 \times 0.67 = 0.1005$$

In this case the predicted class is $\hat{y} = c_2$.

Iris Data: Test Case with Pseudo-counts

The test point $\mathbf{x} = (6.75, 4.25)^T$ corresponds to the categorical point (Long, Long), and it is represented as $\mathbf{v} = (\mathbf{e}_{13}^T \quad \mathbf{e}_{23}^T)^T$.

Unfortunately the probability mass at \mathbf{v} is zero for both classes. We adjust the PMF via pseudo-counts noting that the number of possible values are $m_1 \times m_2 = 4 \times 3 = 12$.

The likelihood and prior probability can then be computed as

$$\hat{P}(\mathbf{x}|c_1) = \hat{f}(\mathbf{v}|c_1) = \frac{0+1}{50+12} = 1.61 \times 10^{-2}$$

$$\hat{P}(\mathbf{x}|c_2) = \hat{f}(\mathbf{v}|c_2) = \frac{0+1}{100+12} = 8.93 \times 10^{-3}$$

$$\hat{P}(c_1|\mathbf{x}) \propto (1.61 \times 10^{-2}) \times 0.33 = 5.32 \times 10^{-3}$$

$$\hat{P}(c_2|\mathbf{x}) \propto (8.93 \times 10^{-3}) \times 0.67 = 5.98 \times 10^{-3}$$

Thus, the predicted class is $\hat{y} = c_2$.

Bayes Classifier: Challenges

The main problem with the Bayes classifier is the lack of enough data to reliably estimate the joint probability density or mass function, especially for high-dimensional data.

For numeric attributes we have to estimate $O(d^2)$ covariances, and as the dimensionality increases, this requires us to estimate too many parameters.

For categorical attributes we have to estimate the joint probability for all the possible values of \mathbf{v} , given as $\prod_j |\text{dom}(X_j)|$. Even if each categorical attribute has only two values, we would need to estimate the probability for 2^d values. However, because there can be at most n distinct values for \mathbf{v} , most of the counts will be zero.

Naive Bayes classifier addresses these concerns.

Naive Bayes Classifier: Numeric Attributes

The naive Bayes approach makes the simple assumption that all the attributes are independent, which implies that the likelihood can be decomposed into a product of dimension-wise probabilities:

$$P(\mathbf{x}|c_i) = P(x_1, x_2, \dots, x_d|c_i) = \prod_{j=1}^d P(x_j|c_i)$$

The likelihood for class c_i , for dimension X_j , is given as

$$P(x_j|c_i) \propto f(x_j|\hat{\mu}_{ij}, \hat{\sigma}_{ij}^2) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_{ij}} \exp\left\{-\frac{(x_j - \hat{\mu}_{ij})^2}{2\hat{\sigma}_{ij}^2}\right\}$$

where $\hat{\mu}_{ij}$ and $\hat{\sigma}_{ij}^2$ denote the estimated mean and variance for attribute X_j , for class c_i .

Naive Bayes Classifier: Numeric Attributes

The naive assumption corresponds to setting all the covariances to zero in $\widehat{\Sigma}_i$, that is,

$$\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{i2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & \sigma_{id}^2 \end{pmatrix}$$

The naive Bayes classifier thus uses the sample mean $\hat{\mu}_i = (\hat{\mu}_{i1}, \dots, \hat{\mu}_{id})^T$ and a *diagonal* sample covariance matrix $\widehat{\Sigma}_i = \text{diag}(\sigma_{i1}^2, \dots, \sigma_{id}^2)$ for each class c_i . In total $2d$ parameters have to be estimated, corresponding to the sample mean and sample variance for each dimension X_j .

Naive Bayes Algorithm

NAIVEBAYES ($\mathbf{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^n$):

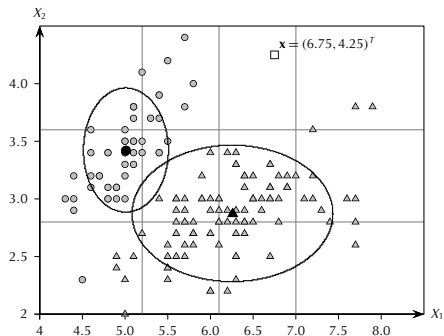
```
1 for  $i = 1, \dots, k$  do
2    $\mathbf{D}_i \leftarrow \{\mathbf{x}_j \mid y_j = c_i, j = 1, \dots, n\}$  // class-specific subsets
3    $n_i \leftarrow |\mathbf{D}_i|$  // cardinality
4    $\hat{P}(c_i) \leftarrow n_i/n$  // prior probability
5    $\hat{\boldsymbol{\mu}}_i \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \mathbf{x}_j$  // mean
6    $\mathbf{Z}_i = \mathbf{D}_i - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}_i^T$  // centered data for class  $c_i$ 
7   for  $j = 1, \dots, d$  do // class-specific variance for  $X_j$ 
8      $\hat{\sigma}_{ij}^2 \leftarrow \frac{1}{n_i} \mathbf{Z}_{ij}^T \mathbf{Z}_{ij}$  // variance
9    $\hat{\boldsymbol{\sigma}}_i = (\hat{\sigma}_{i1}^2, \dots, \hat{\sigma}_{id}^2)^T$  // class-specific attribute variances
10 return  $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i$  for all  $i = 1, \dots, k$ 
```

TESTING (\mathbf{x} and $\hat{P}(c_i), \hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\sigma}}_i$, for all $i \in [1, k]$):

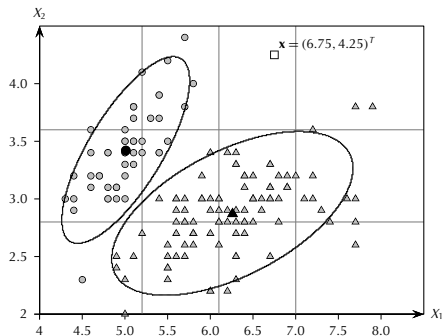
```
11  $\hat{y} \leftarrow \arg \max_{c_i} \left\{ \hat{P}(c_i) \prod_{j=1}^d f(x_j \mid \hat{\boldsymbol{\mu}}_{ij}, \hat{\sigma}_{ij}^2) \right\}$ 
12 return  $\hat{y}$ 
```

Naive Bayes versus Full Bayes Classifier: Iris 2D Data

X_1 : sepal length versus X_2 : sepal width



(a) Naive Bayes



(b) Full Bayes

Naive Bayes: Categorical Attributes

The independence assumption leads to a simplification of the joint probability mass function

$$P(\mathbf{x}|c_i) = \prod_{j=1}^d P(x_j|c_i) = \prod_{j=1}^d f(\mathbf{X}_j = \mathbf{e}_{jr_j} | c_i)$$

where $f(\mathbf{X}_j = \mathbf{e}_{jr_j} | c_i)$ is the probability mass function for \mathbf{X}_j , which can be estimated from \mathbf{D}_i as follows:

$$\hat{f}(\mathbf{v}_j | c_i) = \frac{n_i(\mathbf{v}_j)}{n_i}$$

where $n_i(\mathbf{v}_j)$ is the observed frequency of the value $\mathbf{v}_j = \mathbf{e}_{jr_j}$ corresponding to the r_j th categorical value a_{jr_j} for the attribute X_j for class c_i .

If the count is zero, we can use the pseudo-count method to obtain a prior probability. The adjusted estimates with pseudo-counts are given as

$$\hat{f}(\mathbf{v}_j | c_i) = \frac{n_i(\mathbf{v}_j) + 1}{n_i + m_j}$$

where $m_j = |\text{dom}(X_j)|$.

Nonparametric Approach: K Nearest Neighbors Classifier

We consider a non-parametric approach for likelihood estimation using the nearest neighbors density estimation.

Let \mathbf{D} be a training dataset comprising n points $\mathbf{x}_i \in \mathbb{R}^d$, and let \mathbf{D}_i denote the subset of points in \mathbf{D} that are labeled with class c_i , with $n_i = |\mathbf{D}_i|$.

Given a test point $\mathbf{x} \in \mathbb{R}^d$, and K , the number of neighbors to consider, let r denote the distance from \mathbf{x} to its K th nearest neighbor in \mathbf{D} .

Consider the d -dimensional hyperball of radius r around the test point \mathbf{x} , defined as

$$B_d(\mathbf{x}, r) = \{\mathbf{x}_i \in \mathbf{D} \mid \delta(\mathbf{x}, \mathbf{x}_i) \leq r\}$$

Here $\delta(\mathbf{x}, \mathbf{x}_i)$ is the distance between \mathbf{x} and \mathbf{x}_i , which is usually assumed to be the Euclidean distance, i.e., $\delta(\mathbf{x}, \mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|_2$. We assume that $|B_d(\mathbf{x}, r)| = K$.

Nonparametric Approach: K Nearest Neighbors Classifier

Let K_i denote the number of points among the K nearest neighbors of \mathbf{x} that are labeled with class c_i , that is

$$K_i = \{ \mathbf{x}_j \in B_d(\mathbf{x}, r) \mid y_j = c_i \}$$

The class conditional probability density at \mathbf{x} can be estimated as the fraction of points from class c_i that lie within the hyperball divided by its volume, that is

$$\hat{f}(\mathbf{x}|c_i) = \frac{K_i/n_i}{V} = \frac{K_i}{n_i V}$$

where $V = \text{vol}(B_d(\mathbf{x}, r))$ is the volume of the d -dimensional hyperball. The posterior probability $P(c_i|\mathbf{x})$ can be estimated as

$$P(c_i|\mathbf{x}) = \frac{\hat{f}(\mathbf{x}|c_i)\hat{P}(c_i)}{\sum_{j=1}^k \hat{f}(\mathbf{x}|c_j)\hat{P}(c_j)}$$

However, because $\hat{P}(c_i) = \frac{n_i}{n}$, we have

$$\hat{f}(\mathbf{x}|c_i)\hat{P}(c_i) = \frac{K_i}{n_i V} \cdot \frac{n_i}{n} = \frac{K_i}{nV}$$

Nonparametric Approach: K Nearest Neighbors Classifier

The posterior probability is given as

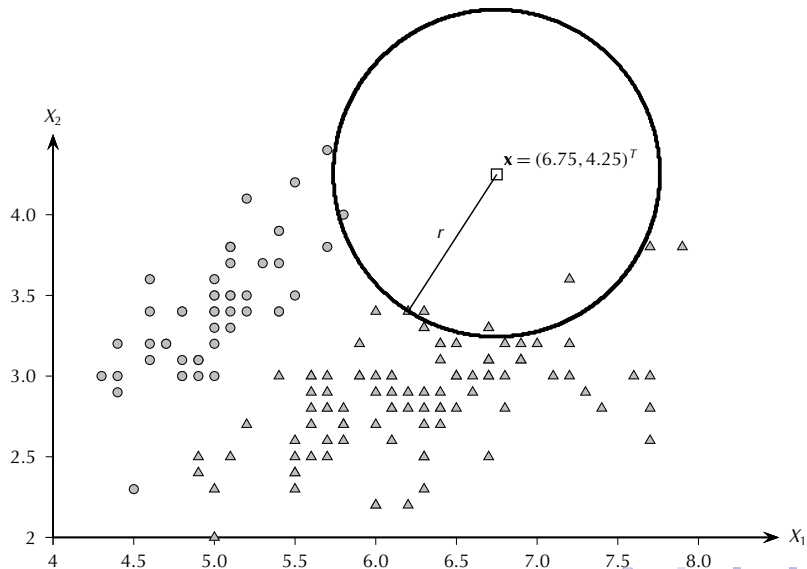
$$P(C_i|\mathbf{x}) = \frac{\frac{K_i}{nV}}{\sum_{j=1}^k \frac{K_j}{nV}} = \frac{K_i}{K}$$

Finally, the predicted class for \mathbf{x} is

$$\hat{y} = \arg \max_{c_i} \{P(C_i|\mathbf{x})\} = \arg \max_{c_i} \left\{ \frac{K_i}{K} \right\} = \arg \max_{c_i} \{K_i\}$$

Because K is fixed, the KNN classifier predicts the class of \mathbf{x} as the majority class among its K nearest neighbors.

Iris Data: K Nearest Neighbors Classifier



Motivation

- There is an increasing use of the Web in events of overall interest such as politics and sports.
- Major motivations are the lack of a central control and the fast information propagation.
- Recently, there has been an emphasis on "what you are doing" instead of "who you are".

Challenge

Qualify, quantify, and summarize the content being exchanged in the various Internet-related media on line and evaluate its impact on specific events.

On line tool for capturing, analyzing and presenting the dynamics of a given scenario on the Web.

Scenarios

- Soccer World Cup
- Olympics
- Brazilian National Soccer League
- Brazilian Elections
- Public Safety
- Brand reputation
- **Dengue Epidemics**

Background on dengue

- Dengue is a mosquito-borne infection that causes a severe flu-like illness, and sometimes a potentially lethal complication
- Approximately 2 billion people from more than 100 countries are at risk of infection and about 50 million infections occur every year worldwide
- Outbreaks tend to occur every year during the rainy season but there is large variation of the degree of the epidemic in areas with similar rainfall

Background on dengue

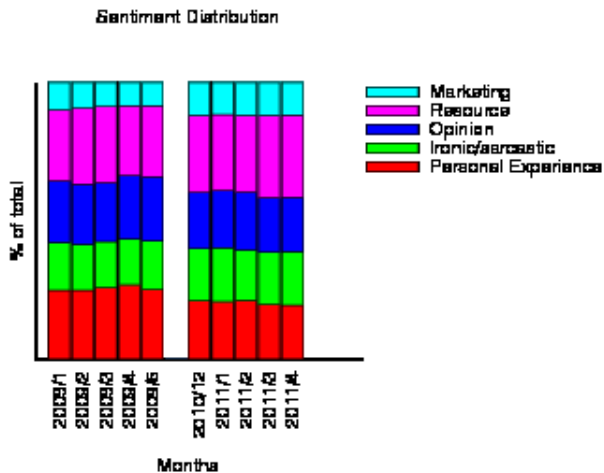
- Current strategies for prediction of dengue epidemics are based on surveillance of insects, which provide only a rough estimate of cases
- Once disease outbreaks are detected in a certain area, efforts need to be concentrated to avoid further cases and to optimize treatment and staff - number of cases may reach several hundred thousands
- In Brazil, where there is a epidemics accounting system, detection of important outbreaks may take a few weeks, leading to loss of precious time to address the epidemic

- To analyze how dengue epidemics manifests in Twitter and to what extent that information can be used for surveillance.
- To design and implement an active surveillance framework that analyzes how social media reflects epidemics based on a combination of four dimensions: volume, location, time, and public perception.
- To exploit user generated content available in online social media to predict the dengue epidemics.

- Active dengue surveillance based on four dimensions:
 - Public perception
 - Volume
 - Location
 - Time
- Methodology steps
 - Content analysis
 - Correlation analysis
 - Spatio-temporal analysis
 - Surveillance

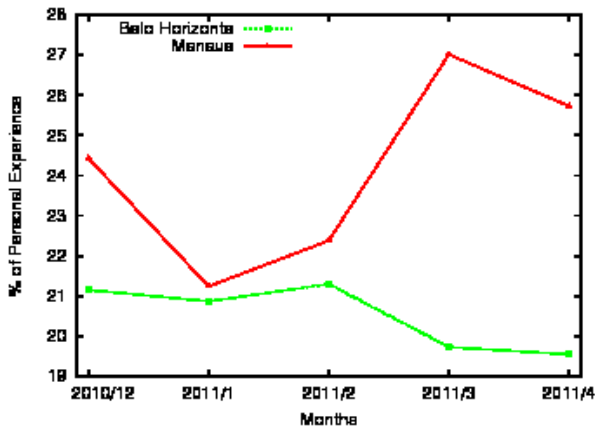
- Determine the sentiment categories
 - **Personal experience:** “You know I have had dengue?”
 - **Ironic/sarcastic tweets:** “My life looks like a dengue-prone steady water”
 - **Opinion:** “The campaign against dengue is very cool”
 - **Resource:** “Dengue virus type 4 in circulation”
 - **Marketing:** “Everybody must fight dengue. Brazil relies on you”

Sentiment distribution over time



Content analysis

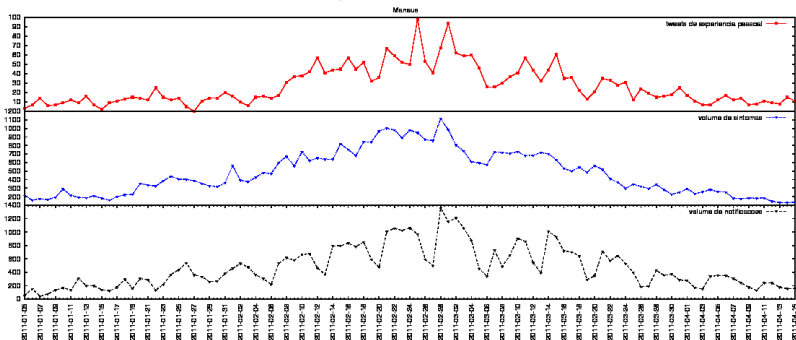
Is personal experience a good indicator of dengue's incidence?



Manaus

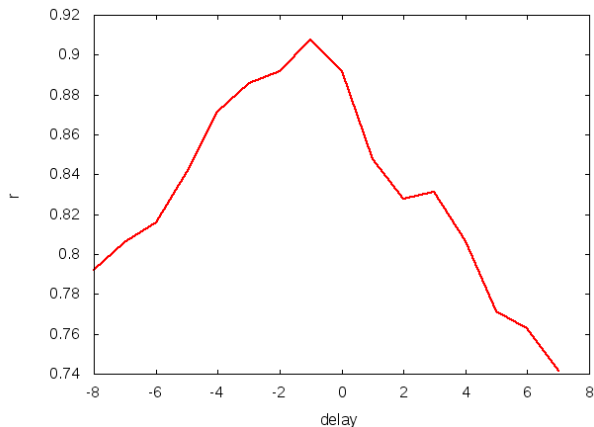
Personal experience, notifications and symptom perception

From November, 2010 to May, 2011



Manaus

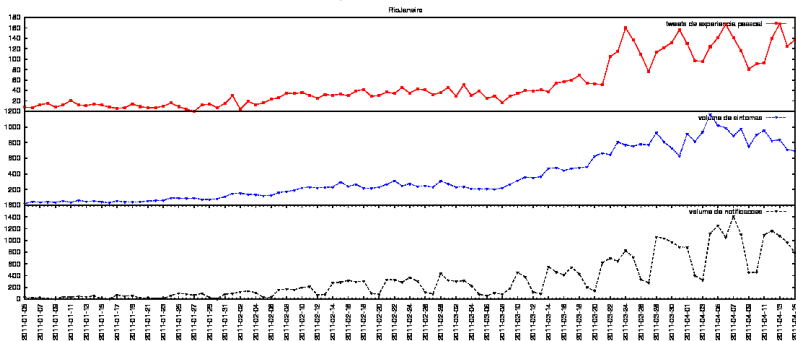
Cross-correlation between personal experience and symptom perception from November, 2010 to May, 2011



Rio de Janeiro

Personal experience, notifications and symptom perception

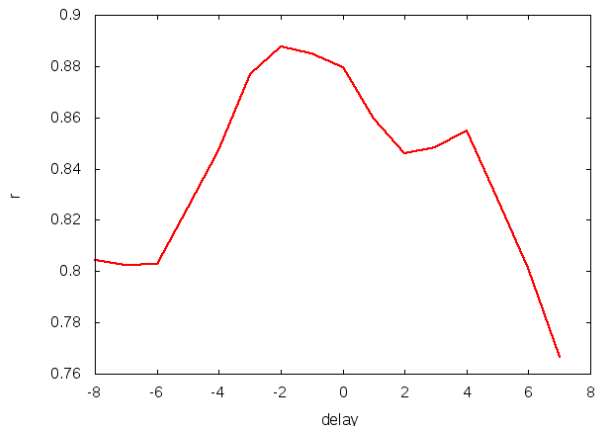
From November, 2010 to May, 2011



Correlation Analysis

Rio de Janeiro

Cross-correlation between personal experience and symptom perception from November, 2010 to May, 2011



Spatio-temporal analysis

- Evaluated two metrics
 - the volume of tweets
 - the PTPE value

Spatio-temporal analysis

- Evaluated two metrics

- the volume of tweets
- the PTPE value

Rand Index = 0.8506

Rand Index = 0.8914

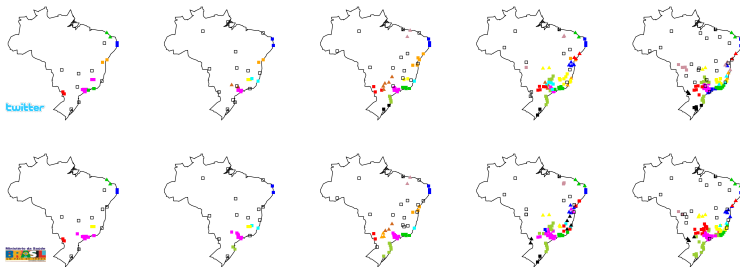
Spatio-temporal analysis

- Evaluated two metrics

- the volume of tweets
- the PTPE value

Rand Index = 0.8506

Rand Index = 0.8914



- Strategy: Analyze the ratio of personal experience tweets weekly.
- Intuition: a sudden increase in this ratio indicates a surge
- Visual metaphors
 - maps
 - temporal graphs

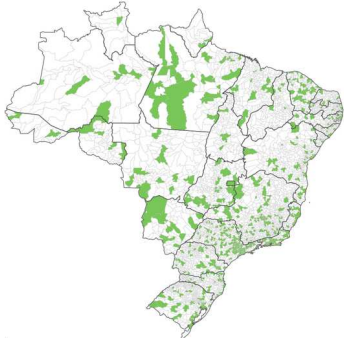
SEMANA DE REFERÊNCIA COMEÇANDO EM

28/03/2013

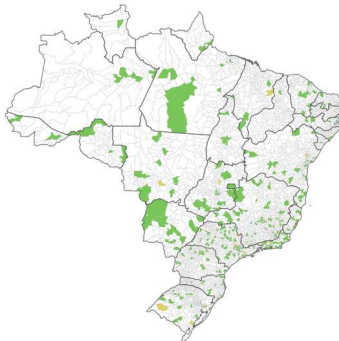
Selecione Outra ▾

Mapas Relativos a Dengue no Brasil

Incidência Relativa



Tendência Relativa



- Twitter data are useful for epidemics surveillance.
- Enablers:
 - Dengue is an urban disease, as it is the Internet usage in Brazil.
 - Dengue-related tweets are easy to collect.
 - People talk about dengue spontaneously.
- Tweets associated with “personal experience” present high correlation with dengue incidence.
- Simple alarm systems are effective to detect dengue surges.

Hot-Spot Mining from Case-Control Trajectories

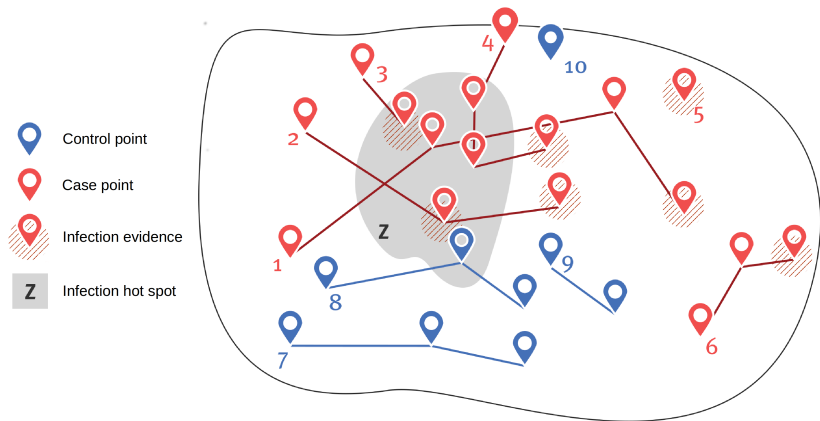
Problem

- Detecting **infection hot spots** from case-control trajectories.
- We target the identification of **infection hot spots related to Dengue** through geo-tagged tweets.

Challenges

- Trajectories may be composed of **unlimited** and **distinct** number of points.
- The place where a person tweets about dengue is not necessarily where she got infected.

Problem Definition



- The Visit Model finds **the most likely zones that a diseased person visits**.
- Let $V_{i,z}$ the random number of tweets in Z among the n_i total number from the i -th individual.
- Let $p = p(Z)$ the probability that, giving that a case individual is tweeting, she does it from within Z . ($\bar{p} = \bar{p}(Z)$ is the analogous for a control individual).
- For a user who is a case, $P(V_{i,z} \geq 1) = 1 - (1 - p)^{n_i}$. For a control user, it is equals to $1 - (1 - \bar{p})^{n_i}$.

- The log-likelihood for the Visit Model can be written as:

$$\ell_1(Z, p, \bar{p}) = \log(1 - p) \cdot N(\bar{Z}) + \log(1 - \bar{p}) \cdot M(\bar{Z}) +$$

$$\sum_{i=1}^N \mathbb{1}[V_{i,Z} \geq 1] \cdot \log(1 - (1 - p)^{n_i}) +$$

$$\sum_{i=N+1}^{N+M} \mathbb{1}[V_{i,Z} \geq 1] \cdot \log(1 - (1 - \bar{p})^{n_i})$$

where N is the number of cases and M , the number of controls.

Infection Model

- The Infection Model finds **the most likely zones where a person gets infected while visiting**.
- We want to estimate the probability that someone issues a dengue-labelled tweet given that she visited k times the region Z .
- Let $r = r(Z)$ be the infection risk inside the candidate cluster ($\bar{r} = r(\bar{Z})$ is the same outside Z).
- Let I_i be the binary indicator that individual i is a case. So,

$$\begin{aligned}\mathbb{P}(I_i = 1 | V_{i,Z} = k_i) &= 1 - \mathbb{P}(I_i = 0 | V_{i,Z} = k_i) \\ &= 1 - (1 - r)^{k_i} (1 - \bar{r})^{n_i - k_i}\end{aligned}$$

- The log-likelihood for the Infection Model can be written as:

$$\ell_2(Z, r, \bar{r}) = \sum_{i=1}^{N+M} l_i \cdot \log(1 - (1 - r)^{k_i} \cdot (1 - \bar{r})^{n_i - k_i}) + (1 - l_i) \cdot (k_i \cdot \log(1 - r) + (n_i - k_i) \cdot \log(1 - \bar{r}))$$

Visit Model:

$\mathbb{P}(\text{Tweets from } Z \mid \text{Is a case})$

Infection Model:

$\mathbb{P}(\text{Is a case} \mid K \text{ Tweets from } Z)$

- The test statistic for the Visit Model is:

$$T_1 = \ell_1(\hat{Z}, \hat{p}, \hat{\hat{p}}) = \sup_{\substack{Z \in \mathcal{Z} \\ \hat{p}(Z) > \hat{\hat{p}}(Z)}} \ell_1(\hat{Z}, \hat{p}(Z), \hat{\hat{p}}(Z))$$

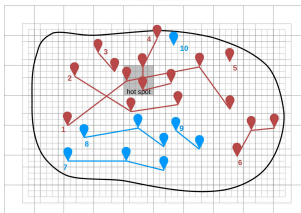
- The exact p-value for the null hypothesis is given by:

$$p_1 = \frac{1}{B} (1 + \#\{T_1^{(k)} \geq T_1, k = 1, \dots, B-1\})$$

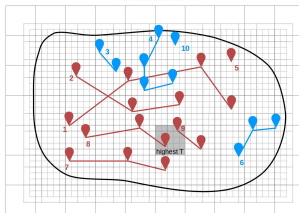
- Both are analogous for the Infection Model.

Evaluating the Data Evidence

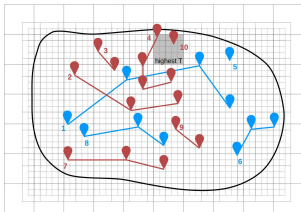
Real Labels



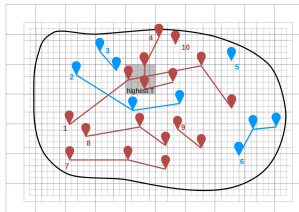
Simulation 1



Simulation 2



Simulation B



- We collected the data from Twitter.
- Geo-located tweets from Brazil.
- Period: Jan 1st, 2015 – Dec 31th, 2015.
- Total of 106,784,441 tweets.
- City-level analysis.
- Selected 11 municipalities, including cities facing strong surges.

Table: Data summary. We present the total number of tweets collected in the city (#msg); the respective number of unique users (#unq_usr); the number of case and control individuals (#case_usr and #ctrl_usr) and the respective number of tweets they issued (#case_msg and #ctrl_msg).

City name	#msg	#unq_usr	#case_msg	#case_usr	#ctrl_msg	#ctrl_usr
Belém	1,049,433	19,611	8,134	23	18,416	65
B. Horizonte	3,134,497	50,360	60,968	104	168,820	302
Curitiba	1,694,301	35,775	3,028	18	9,066	54
Goiânia	566,114	16,849	15,933	54	33,750	147
Natal	522,689	16,689	3,847	15	8,748	42
R. de Janeiro	9,875,435	167,567	71,115	163	213,168	490
São Paulo	6,965,165	174,544	167,772	413	486,264	1229
Campinas	574,226	20,335	37,313	90	64,442	226
Limeira	91,454	2,991	11,614	47	16,830	108
SJ. Campos	407,143	9,697	19,883	58	40,251	148
Sorocaba	230,224	7,471	32,734	91	39,352	206

- For each selected city we applied the **Visit** and **Infection** Models.
- The zones Z are defined by overlaying different grids on the map and each cell corresponds to a zone to be scanned.
- We also set the number of Monte Carlo replicas to 999 and significance level as 0.05.

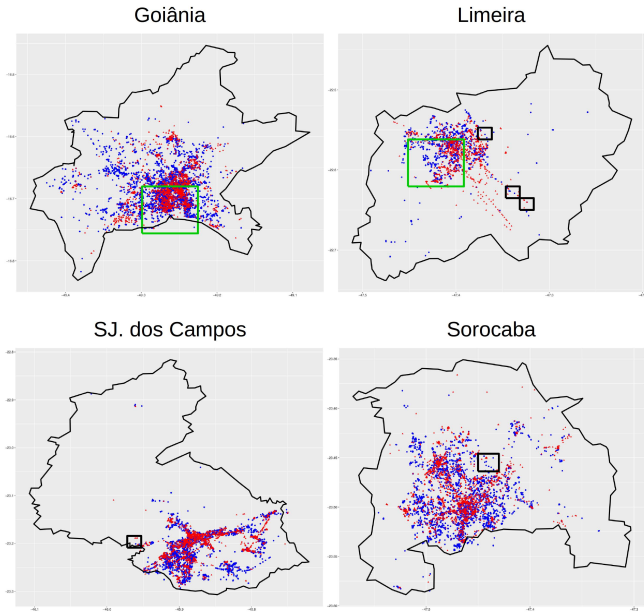


Figure: Zones found by the Visit (green) and Infection (black) Models.

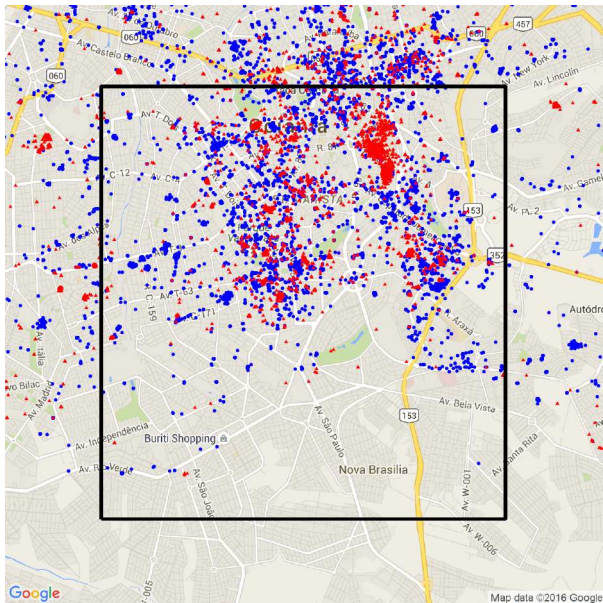


Figure: Zoom in to the hot spot in Goiânia.

Principles for Algorithmic Transparency and Accountability

- 1 Awareness
- 2 Access and redress
- 3 Accountability
- 4 Explanation
- 5 Data provenance
- 6 Auditability
- 7 Validation and testing

Principles for Algorithmic Transparency and Accountability

1. Awareness

Owners, designers, builders, users, and other stakeholders of analytic systems should be aware of the possible biases involved in their design, implementation, and use and the potential harm that biases can cause to individuals and society.

Principles for Algorithmic Transparency and Accountability

2. Access and redress

Regulators should encourage the adoption of mechanisms that enable questioning and redress for individuals and groups that are adversely affected by algorithmically informed decisions.

Principles for Algorithmic Transparency and Accountability

3. Accountability

Institutions should be held responsible for decisions made by the algorithms that they use, even if it is not feasible to explain in detail how the algorithms produce their results.

Principles for Algorithmic Transparency and Accountability

4. Explanation

Systems and institutions that use algorithmic decision-making are encouraged to produce explanations regarding both the procedures followed by the algorithm and the specific decisions that are made. This is particularly important in public policy contexts.

Principles for Algorithmic Transparency and Accountability

5. Data Provenance

A description of the way in which the training data was collected should be maintained by the builders of the algorithms, accompanied by an exploration of the potential biases induced by the human or algorithmic data-gathering process. Public scrutiny of the data provides maximum opportunity for corrections. However, concerns over privacy, protecting trade secrets, or revelation of analytics that might allow malicious actors to game the system can justify restricting access to qualified and authorized individuals.

Principles for Algorithmic Transparency and Accountability

6. Auditability

Models, algorithms, data, and decisions should be recorded so that they can be audited in cases where harm is suspected.

Principles for Algorithmic Transparency and Accountability

7. Validation and Testing

Institutions should use rigorous methods to validate their models and document those methods and results. In particular, they should routinely perform tests to assess and determine whether the model generates discriminatory harm. Institutions are encouraged to make the results of such tests public.

Google is not 'just' a platform. It frames, shapes and distorts how we see the world

Carole Cadwalladr



Last week, we reported how extremist sites 'game' the search engine, boosting their propaganda. In response, the web giant appears to have modified some results, but would like us not to notice

Google

did the holocaust happen

did the holocaust happen
did the holocaust happen during ww2
did the holocaust really happen yahoo
did the holy grail exist

Top 10 reasons why the holocaust didn't happen. - Stormfront

<https://www.stormfront.org> » General » History & Revisionism

19 Dec 2008 - 10 posts - 6 authors

The Holocaust Lie more than anything else keeps us down. The twin ... You can believe what you want, but I believe the holocaust did happen.

Holocaust denial - Wikipedia

https://en.wikipedia.org/wiki/Holocaust_denial

Holocaust denial is the act of denying the genocide of Jews and other groups in the Holocaust ... denial movement bases its approach on the predetermined idea that the Holocaust, as understood by mainstream historiography, did not occur.

Laws against Holocaust denial - Criticism - Order of magnitude

The Holocaust Never: IT NEVER HAPPENED LET B

Advertisement

NOVAS ROTAS TAMBÉM PARA:

Abidjan
Alicante
Budapeste
Colônia
Gran Canaria

☰ **MOTHERBOARD** It's Our Fault That AI Thinks White Names Are More 'Pleasant' Than

f SHARE

🐦 TWEET

⋮

It's Our Fault That AI Thinks White Names Are More 'Pleasant' Than Black Names



JORDAN PEARSON

Aug 26 2016, 10:00am



Image: Shutterstock

ADVERTISEMENT

LATAM
AIRLINES

Ofertas imperdíveis
para você descobrir o
melhor do mundo.

Transferindo dados de dt.adsafeprotected.com...

...e unsettling trends in our

Discrimination

- To discriminate is to treat someone differently
(Unfair) discrimination is based on group membership, not individual merit
- People's decisions include objective and subjective elements
Hence, they can be discriminate
- Algorithmic inputs include only objective elements
Hence, can they discriminate?

- On the web: race and gender stereotypes reinforced
Results for "CEO" in Google Images: 11% female, US 27% female CEOs
Also in Google Images, "doctors" are mostly male, "nurses" are mostly female
- Geography and race: the "Tiger Mom Tax"
Pricing of SAT tutoring by The Princeton Review in the US doubles for Asians, due to geographical price discrimination

Self-perpetuating algorithmic biases

- Credit scoring algorithm suggests Joe has high risk of defaulting
- Hence, Joe needs to take a loan at a higher interest rate
- Hence, Joe has to make payments that are more onerous
- Hence, Joe's risk of defaulting has increased

Sources of algorithmic bias

- Data as a social mirror
Protected attributes redundantly encoded in observables
- Correctness and completeness
Garbage in, garbage out (GIGO)

Data mining assumptions might not hold

- Data mining assumptions are not always observed in reality
 - Variables might not be independently identically distributed
 - Samples might be biased
 - Labels might be incorrect
- Errors might be concentrated in a particular class
- Sometimes, we might be seeking more simplicity than what is possible

Main concerns: data and algorithms

- Data inputs:
 - Poorly selected (e.g., observe only car trips, not bicycle trips)
 - Incomplete, incorrect, or outdated
 - Selected with bias (e.g., smartphone users)
 - Perpetuating and promoting historical biases (e.g., hiring people that "fit the culture")
- Algorithmic processing:
 - Poorly designed matching systems
 - Personalization and recommendation services that narrow instead of expand user options
 - Decision making systems that assume correlation implies causation
 - Algorithms that do not compensate for datasets that disproportionately represent populations
 - Output models that are hard to understand or explain hinder detection and mitigation of bias

Connection between privacy and discrimination

- Finding if people having attribute X were discriminated is like inferring attribute X from a database in which:
 - the attribute X was removed
 - a new attribute (the decision), which is based on X , was added
- This is similar to trying to reconstruct a column from a privacy-scrubbed dataset

Goal: Develop a non-discriminatory decision-making process while preserving as much as possible the quality of the decision.

Steps:

- 1 Defining anti-discrimination/fairness constraints
- 2 Transforming data/algorithm/model to satisfy the constraints
- 3 Measuring data/model utility

Fairness-aware data mining

- Pre-processing:** input data transformations to minimize discrimination while accuracy is maximized (e.g., suppression, massaging, reweighing, sampling).
- In-processing:** novel algorithms that achieve the same goal (e.g., change split criterion and leaf relabeling in decision trees). A classifier is fair if it is not affected by the presence of sensitive data in the training set.
- Post-processing:** output models should not discriminate, how to clean the traces of discrimination (e.g., pattern sanitization, which is similar to anonymization).

The screenshot shows the top portion of a New York Times article. At the top left, there are navigation links for 'SECTIONS', 'HOME', and 'SEARCH'. The New York Times logo is centered at the top. On the right, there are buttons for 'SUBSCRIBE NOW', 'LOG IN', and a settings gear icon. Below the logo is the 'The Upshot' sub-brand. A 'DANGER ZONE' tag is visible. The article title is 'Inside the Algorithm That Tries to Predict Gun Violence in Chicago'. The byline reads 'By JEFF ASHER and ROB ARTHUR' with the date 'JUNE 13, 2017'. Social media sharing icons for Facebook, Twitter, Email, Print, and a bookmark icon are present. The main text begins with 'Gun violence in Chicago has surged since late 2015, and much of the news media attention on how the city plans to address this problem has focused on the Strategic Subject List, or S.S.L.' The second paragraph explains that the list is made by an algorithm to predict who is most likely to be involved in a shooting. The third paragraph states that the city has placed a version of the list online through its open data portal. A bulleted point follows, stating that violence is less concentrated at the top than public comments suggest. On the right side, there is a 'RELATED COVERAGE' section with four article thumbnails and titles: 'When a Computer Program Keeps You in Jail', 'Drug Deaths in America Are Rising Faster Than Ever', 'This small Indiana county sends more people to prison than San Francisco and Durham, N.C., combined. Why?', and 'Your Rabbi? Probably a Democrat. Your Baptist Pastor? Probably a Republican. Your Priest? Who Knows.' The bottom of the page features a navigation bar with icons for back, forward, and search, and a footer with the text 'Meira Jr. (UFMG) Data Mining Chapter 18: Probabilistic Classification 159 / 171'.

SECTIONS HOME SEARCH

The New York Times

SUBSCRIBE NOW LOG IN

THE UPSHOT

FOLLOW US: GET THE UPSHOT IN YOUR INBOX

DANGER ZONE

Inside the Algorithm That Tries to Predict Gun Violence in Chicago

By JEFF ASHER and ROB ARTHUR JUNE 13, 2017

Gun violence in Chicago has surged since late 2015, and much of the [news media attention](#) on how the city plans to address this problem has focused on the Strategic Subject List, or S.S.L.

The list is made by an algorithm that tries to predict who is most likely to be involved in a shooting, either as perpetrator or victim. The algorithm is not public, but the city has now placed a [version of the list](#) — without names — online through its open data portal, making it possible for the first time to see how Chicago evaluates risk.

We analyzed that information and found that the assigned risk scores — and what characteristics go into them — are sometimes at odds with the Chicago Police Department’s public statements and cut against some common perceptions.

- Violence in the city is less concentrated at the top — among a group of about 1,400 people with the highest risk scores — than some public comments from the Chicago police have suggested.

RELATED COVERAGE

Opinion | Op-Ed Contributor
When a Computer Program Keeps You in Jail JUNE 13, 2017

Drug Deaths in America Are Rising Faster Than Ever JUNE 5, 2017

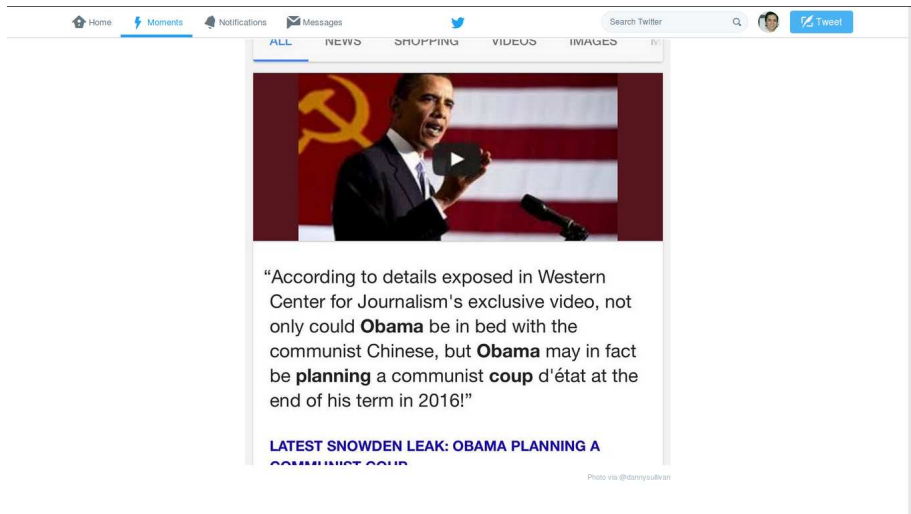
UNEQUAL JUSTICE
This small Indiana county sends more people to prison than San Francisco and Durham, N.C., combined. Why? SEPT 2, 2016

Your Rabbi? Probably a Democrat. Your Baptist Pastor? Probably a Republican. Your Priest? Who Knows. JUNE 12, 2017

TAKING A TOLL
How Prejudice Can Harm Your Health


Meira Jr. (UFMG) Data Mining Chapter 18: Probabilistic Classification 159 / 171

Transparency?



Home Moments Notifications Messages Search Twitter Tweet

ALL NEWS SHOPPING VIDEOS IMAGES



“According to details exposed in Western Center for Journalism's exclusive video, not only could **Obama** be in bed with the communist Chinese, but **Obama** may in fact be **planning** a communist **coup d'état** at the end of his term in 2016!”

LATEST SNOWDEN LEAK: OBAMA PLANNING A COMMUNIST COUP

Photo via @dannysullivan

Transparency may imply, in a broader sense, model interpretability:

- **Trust:** Confidence that a model will perform well. More specifically, not only how often it performs well, but also for which cases.
- **Causality:** To what extent may we generalize associations to infer properties?
- **Transferability:** Capacity of transferring learned skills to unfamiliar situations.
- **Informativeness:** How actionable is the pattern or model?
- **Fair and Ethical-Decision Making:** Are the models fair? Do they follow ethical patterns?

- **Transparency:** How does the model work?
- **Post-hoc explanations:** What else can the model tell me?

- **Simulatability:** A human should be able to take the input data together with the parameters and, in reasonable time, *compute* the model.
- **Decomposability:** Each part of the model (input, parameter, calculation) admits an intuitive explanation.
- **Algorithmic transparency:** We should be able to understand how the model was built, i.e., its principles, capabilities and limitations.

- **Text explanations:** Build an additional model that explains textually the outputs of a primal model.
- **Visualizations:** Render visualizations of the model and its outputs to ease understanding and usage.
- **Local explanations:** Zoom in the search space associated with input data and build a *local* model.
- **Explanation by example:** Report which training samples resemble the input data.

Tim Berners-Lee calls for tighter regulation of online political advertising

Inventor of the worldwide web described in an open letter how it has become a sophisticated and targeted industry, drawing on huge pools of personal data



Tim Berners-Lee: 'Targeted advertising allows a campaign to say completely different, possibly conflicting things to different groups. Is that democratic?' Photograph: Bloomberg/Bloomberg via Getty Images

Esperando por ophan.theguardian.com...

1599



Advertisement

An advertisement for LATAM Airlines. The top right corner features the LATAM AIRLINES logo. The main text reads "Novo MERCADO LATAM". Below this, it says "Mais e melhores opções para você comprar a bordo." At the bottom, there is a pink button that says "SAIBA MAIS". The background of the ad shows various food items like nuts, popcorn, and bread.

Social Media's Silent Filter

Under-the-radar workers have scrubbed objectionable material from Facebook and other sites since well before the fake-news controversy.

SARAH T. ROBERTS | MAR 8, 2017 | TECHNOLOGY



A few months ago, in the wake of the fake-news debacle surrounding the election, Facebook [announced partnerships](#) with four independent fact-checking organizations to stomp out the spread of misinformation on its site. If investigators from at least two of these organizations—*Snopes*, *PolitiFact*, ABC News, and FactCheck.org, all members of the Poynter International Fact Checking Network—flag an article as bogus, that article [now shows up](#) in people's News Feeds with a banner marking it as disputed.

Facebook has said its employees have a hand in this process by separating personal posts from links that present themselves as news, but maintains that they play no role in judging the actual content of the flagged articles themselves. "We believe in giving people a voice and that we cannot become arbiters of truth ourselves," [wrote](#) Adam Mosseri, the vice president of Facebook's News Feed team, in introducing the change.

The announcement was an early step in Facebook's ongoing revision of how it

- What are the personal rights regarding his/her collected data?
- What are the acceptable uses of data?
- Who is liable when something goes wrong?
- How can we report on algorithmical *abuse*?

- Approved in April, 2016.
- Effective in 2018
- Three basic rights:
 - Right to access
 - Right to be forgotten
 - Right to explanation

Article 22: Automated individual decision making, including profiling

- 1 The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarity significantly affects him or her.
- 2 Paragraph 1 shall not apply if the decision:
 - 1 is necessary for entering into, or performance of, a contract between the data subject and the data controller.
 - 2 is authorised by Union or Member State law to which the controller is subject and which also lays down suitable measures to safeguard the data subject's right and freedoms and legitimate interests.
 - 3 is based on data subject's explicit consent.

Right to explanation

Basically, is the right for some interpretability and fairness assurance, but it is challenging:

- intentional concealment on the part of the institutions;
- gaps in technical literacy which mean that having access to technical details is not enough;
- a mismatch between the computational models and the demands of human-scale reasoning and styles of interpretation.

Site: <https://sc.ctweb.inweb.org.br/>

User: aluno_X ($1 \leq X \leq 40$)

Password: sm4rt.Citi3Z