Machine Learning for the Social Sciences:
Algorithms and Societal Impact

Gian Maria Campedelli, PhD
gianmaria.campedelli@unitn.it
Department of Sociology and Social Research - University of Trento

November 24, 2020

Contents

Quick Recap

In the first lecture we covered:

- *Train and Test*
- *Generalizability*
- *Algorithmic Performance Evalutation*
- *Logistic Regression*
- *Regularization Methods*
- *Tree-based Methods (Part 1)*

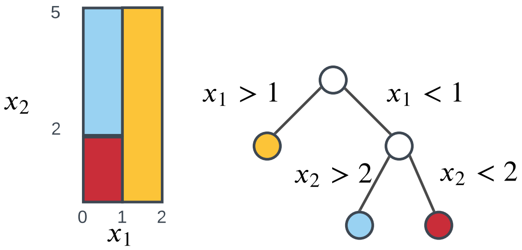
Algorithms

PROS:

- They are generally good when interpretability is necessary (*White box*)
- Little data preparation
- Computational cost: acceptable

CONS:

- Risk of overfitting/instability
- Low performance compared to other methods



What Alternatives?

To improve the predictive performance of single **Decision trees** there are three main alternative:

- **Bagging**
- **Random Forests**
- **Gradient Boosting Machines**

Bagging/1

Bagging, also called **Bootstrap Aggregation**: a general-purpose approach to reduce the variance of a ML/Stat method.

If we have n independent observations $Z_1, ..., Z_N$ each with variance σ^2: variance of the mean $\bar{Z}$ is given by $\frac{\sigma^2}{N} \Rightarrow$ averaging a set of observations reduces **variance**.

Unfeasible because in ML we do not have access to multiple training sets (most of the time)

Bagging/2

Solution: we can bootstrap! We use multiple samples from the same training set

Steps:

1. We gather/generate B distinct training sets (each with roughly 2/3 of all data points)
2. Train the algorithm on each bth bootstrapped training set to obtain $\hat{f}_b(x) \Rightarrow$ prediction on x from bootstrapped set
3. Final step $\Rightarrow$ average all predictions:
   - **Regression:**

$$\hat{f}_{bag} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x) \tag{1}$$

   - **Classification:**

$$\hat{y}_0^{bag} = \underset{k=1,\ldots K}{\operatorname{argmax}} \sum_{b=1}^{B} I\left(\hat{f}_{x_0}^{b} = k\right) \tag{2}$$

Bagging/3

There is quite a straightforward issue with **Bagging**. *Think about it*.

Think about a problem with p predictors, and a bagging procedure. *What can go wrong?*

Bagging/3

There is quite a straightforward issue with **Bagging**. *Think about it*.

Think about a problem with p predictors, and a bagging procedure. *What can go wrong?*

In presence of a very strong predictor, the trees will look similar $\rightarrow$ highly correlated quantities/low reduction in variance

Here they come: Random Forests

Random Forests (Breiman 2001) exactly aim at solving the issue of bagging.

They propose a tweak in order to decorrelate individual trees $\rightarrow$ *diminishing variance*

We apply the **bagging** procedure, but every time a tree has to be split, the algorithm *randomly select m predictors from the set $|p|$* such that $m \approx \sqrt{p}$ (classification) or $m = \frac{p}{3}$ (regression)

Random Forests: Hyperparameters

Besides m, there are several hyperparameters to set:

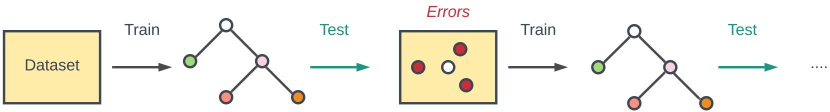
- **Number of Trees:** rule of thumb $\rightarrow p \times 10$
- **Complexity of each tree:** comprises different sub-hyperparams. *Minimum node size* is the most common
- **Sampling scheme:** generally *bootstrapping* where $N(b_1) = N(b_2) = ... = N(B)$
- **Splitting rule:** for classification *Entropy Information Gain* or *Gini Impurity*; for regression *SSR* or *SSE*

Gradient Boosting Machine

Boosting is also a general technique that can be useful in many statistical learning domains.

Contrary to Bagging and RF, **GBM** generates an ensemble of trees *in sequence*: each tree learns from the previous one.

How does it work, practically?



GDB: Algorithmic Structure

GDB works through the following steps:

1. Fit a decision tree to the data ($F_1(x) = y$)
2. Fit the next decision tree to the residuals h of the previous tree $\rightarrow h_1(x) = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right] = y - F_1(x)$
3. Add this new tree $\rightarrow F_2(x) = F_1(x) + h_1(x)$
4. Fit next decision tree to the residuals of $F_2 \rightarrow h_2(x) = y - F_2(x)$
5. Add this new tree $\rightarrow F_3(x) = F_2(x) + h_2(x)$
6. Continue until process (e.g., cross validation) forces to stop

The final learned model represents a *stagewise additive model*:

$$f(x) = \sum_{b=1}^{B} f^b(x) \tag{3}$$

GDB: Remarks

GDB, as other ensemble methods (e.g., **Bagging, RF**) outperforms classic weak learners (e.g., **Decision Trees**).

Particularly, **GDB** represents a state-of-the-art method that often ranks among the best algorithms in many large-scale competitions (e.g., *Kaggle* ones).

Limitations:

- Less interpretable compared to Decision Trees
- Sensitive to outliers

Unsupervised Learning: Intro

Unsupervised Learning concerns the classes of problem in which there is not outcome/target measure, and the goal is therefore to find and describe patterns and associations among inputs.

A more formal definition

Contrary to what happens in supervised learning, unsupervised learning problems do not experience any *teacher* signal. The machine only receives inputs $\mathbf{x}_n \in \mathbb{R}^D$ that form a random p-vector X having joint density $\Pr(X)$

Unsupervised Learning: Tasks

Unsupervised Learning cover many different tasks that are often common in data science applications, including:

- **Clustering**: Finding communities/subgroups of objects that are similar according to a certain criterion (generally *intra-cluster similarity* and *inter-cluster dissimilarity*). Different families of clustering approaches (e.g., distance-based, density-based, hierarchical...)
- **Anomaly Detection**: Detecting instances that significantly deviate from a learned distribution of previous instances
- **Dimensionality Reduction**: Representing X in a lower-dimension feature vector while preserving key properties of the data (overcoming the *curse of dimensionality* reduction
- **Density Estimation**: Constructing an estimate of the probability density function using a set of data points

Unsupervised Learning: Tasks

Unsupervised Learning cover many different tasks that are often common in data science applications, including:

- **Clustering**: Finding communities/subgroups of objects that are similar according to a certain criterion (generally *intra-cluster similarity* and *inter-cluster dissimilarity*). Different families of clustering approaches (e.g., distance-based, density-based, hierarchical...)

Distance and Similarity

Distance and **Similarity** are two key concepts in constructing clustering algorithms.

- **Quantitative data:** $\Rightarrow$ *distance* (e.g., Minkowski, Euclidean, Cosine, Pearson Correlation...)
- **Qualitative data** $\Rightarrow$ *similarity* (e.g., Jaccard, Hamming...)

Evaluation Process

Internal Evaluation: we do not have a ground-truth (as in supervised learning), we can still rely on indicators/coefficients to measure how well a cluster represents data based on the core underlying aim of clustering $\Rightarrow$ *Maximize intra-cluster similarity, maximize inter-cluster dissimilarity*

External Evaluation: we have a ground truth, but we use it afterwards to evaluate how our groups resemble actual sub-spaces in our data

Overview: Internal Evaluation

Internal Evaluation: besides differences, same intuition: *items in the same cluster must be more similar than those in different clusters*. Some strategies:

- **Davies-Bouldin Index (Davies and Bouldin 1979):**

$$DBI = \frac{1}{N}\sum_{i=1}^{N}\max_{j\neq i}\left[\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right] \tag{4}$$

- **Dunn Index (Dunn† 1974):**

$$DI = \frac{min_{1\leq i\leq j\leq n}d(i,j)}{max_{1\leq k\leq n}d'(k)} \tag{5}$$

- **Silhouette Score:**

$$\text{compute } \{a,b,s\} : SC = \max_k \tilde{s}(k) \tag{6}$$

Overview: External Evaluation

Compare groupings based on pre-classified items (often done by experts in the application domain). Some strategies:

- **Purity:**

$$\frac{1}{N}\sum_{m\in M}\max_{d\in D}|m\cap d| \tag{7}$$

- **Rand Index:**

$$RI = \frac{TP+TN}{TP+FP+FN+TN} \tag{8}$$

- *and many more...*

Families of Clustering Approaches

We can distinguish clustering algorithms in several categories:

Families of Clustering Approaches

We can distinguish clustering algorithms in several categories:

- **Centroid-based:** algorithms organizing data in non-hierarchical clusters, based on closeness of points to the centroid of each cluster

Families of Clustering Approaches

We can distinguish clustering algorithms in several categories:

- **Centroid-based:** algorithms organizing data in non-hierarchical clusters, based on closeness of points to the centroid of each cluster
- **Density-based:** algorithms transforming subspaces with high density into clusters

Families of Clustering Approaches

We can distinguish clustering algorithms in several categories:

- **Centroid-based:** algorithms organizing data in non-hierarchical clusters, based on closeness of points to the centroid of each cluster
- **Density-based:** algorithms transforming subspaces with high density into clusters
- **Hierarchical-based:** algorithms creating trees of clusters (well suited for taxonomies, structured data)

Families of Clustering Approaches

We can distinguish clustering algorithms in several categories:

- **Centroid-based:** algorithms organizing data in non-hierarchical clusters, based on closeness of points to the centroid of each cluster
- **Density-based:** algorithms transforming subspaces with high density into clusters
- **Hierarchical-based:** algorithms creating trees of clusters (well suited for taxonomies, structured data)
- **Distribution-based:** algorithms assuming data points are associated with certain distributions (e.g., *Gaussian*)

What We Cover

We are going to cover three algorithms belonging to as many clustering categories:

- *Centroid-based:* **K-Means**
- *Density-based:* **DBSCAN**
- *Hierarchical-based:* **HAC**

k-Means (Lloyd 1982) is **one of the most popular clustering algorithms**. Four main steps:

1. Define/place K centroid into the space represented by the items we want to cluster
2. Assign each item to the cluster with the closest centroid
3. When all objects are assigned, recalculate the positions of the K centroids $\rightarrow c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$
4. Repeat 2 and 3 until centroids do not move anymore

k-Means/2

Steps 2/3 are based on the minimization of the underlying objective function:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| x_i^{(j)} - c_j \right\|^2 \tag{9}$$

with $\left\| x_i^{(j)} - c_j \right\|^2$ being the Euclidean distance between data point $x_i^{(j)}$ and cluster centroid c_j

k-Means/3

Pros:

- Simple to implement
- Computationally efficient

Cons:

- Need to set k (suggestion: test for different values like $2 \leq k \leq 10$)
- Highly sensitive to the initial position of centroids. *Suggestion:* start positioning them very distant from each other as done in **k-means++** (Arthur and Vassilvitskii 2007)
- Difficult to handle outliers/noisy data points

DBSCAN/1

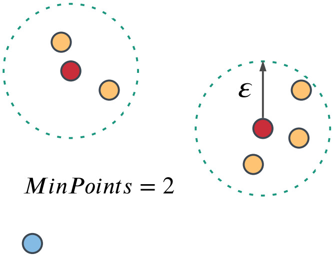
DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) (Ester et al. 1996) performs clustering based on the density of the points in the n-dimensional space.

Contrary to k-means, not all points are necessarily assigned to a given cluster

However, similarly to k-means, a point can be *at most* associated with one cluster

The algorithm exploits two hyperparameters: $minPts$ and ϵ

- $minPts$: the minimum number of points necessary for a region to be considered dense (larger D, larger $minPts$)
- ε: distance measure that is used to locate points in the neighborhood of any other point



DBSCAN/3

Steps:

1. Start with an arbitrary (non-visited) starting point
2. Find the neighborhood of this point using ε
3. If there are at least $minPoints$ in the radius ε, then these belong to that cluster
4. The clusters are then expanded through recursive repetition of the neighborhood search for each neighboring point
5. Procedure stops when all points are visited $\rightarrow$ the points that after the last steps do not belong to any cluster are *noise*

DBSCAN/3

Pros:

- It does not require to pre-specify number of clusters
- Ability to handle clusters of arbitrary size and shape
- Extremely good in separating regions with low density vs regions with high density
- Robust to outliers

Cons:

- Extremely weak performance when dealing with clusters that have similar density
- Computationally expensive
- Specification of $minPoints$ and ε requires experiments/a priori knowledge

Hierarchical Clustering

Hierarchical Clustering works well when dealing with data that have a hierarchical well-defined structure (e.g., taxonomies, documents → very popular in information retrieval)

It can be divided into two different approaches:

- **Agglomerative (or HAC):** each observation is initially treated as a single cluster (*bottom-up*)
- **Divisive:** all observations are initially assigned to a single cluster (*top-down*)

Hierarchical Clustering

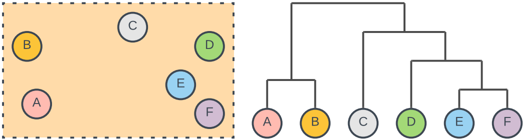
Hierarchical Clustering works well when dealing with data that have a hierarchical well-defined structure (e.g., taxonomies, documents $\rightarrow$ very popular in information retrieval)

It can be divided into two different approaches:

- **Agglomerative (or HAC):** each observation is initially treated as a single cluster (*bottom-up*)

Steps:

1. Create N clusters, with $N = |X|$
2. Combine the two closest data points in a single cluster $\rightarrow$ forming $N - 1$ clusters
3. Take the two closest clusters and combine them in a single cluster $\rightarrow$ forming $N - 2$ clusters
4. Repeat 3) until left with one single cluster



HAC/2

How we define distance/closeness between points/clusters? There exist several linkage methods. The most common are:

- **Complete-linkage:** distance between two clusters is the *longest distance* between two points in each cluster
- **Single-linkage:** distance between two clusters is the *shortest distance* between two points in each cluster (robust against outliers)
- **Average-linkage:** distance between two clusters is the *average distance* between each point in one cluster to every point in the other cluster
- **Centroid-linkage**: distance is the difference between centroid of cluster 1 and centroid of cluster 2, and so on

HAC/3

How to decide when it is time to stop the merging the clusters? Different options:

- Cut at a pre-specified level of similarity
- Cut the dendrogram where the gap between two successive combination similarities is the largets
- Pre-specify k

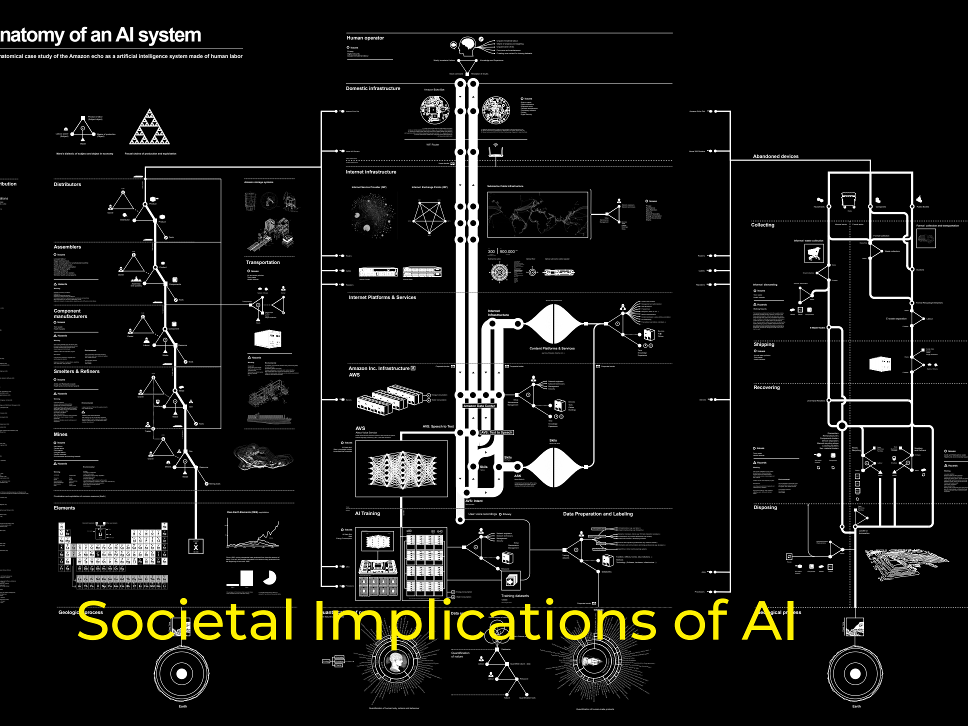
HAC/4

Pros:

- It does not require to pre-specify number of clusters
- Easy implementation
- With moderate data, dendrogram is useful in interpreting the data
- Robust to outliers

Cons:

- With large datasets, difficult to assess the correct number of clusters
- Computationally expensive
- The algorithm doesn't undo previous steps (*errors* stay in the loop)



Societal Implications of AI

Introduction

The excitement for the progress made by algorithms has been followed by concerns over the possible impacts and risks that AI poses to society.

What are these risks? Should we be concerned?

A New Wave of Inquiry/1

In the last years, scholars coming from different fields (foremost *computer science*) helped reinvigorating the debate on AI ethics and on crucial themes such as *transparency, accountability, bias, fairness*.

The scientific community has successfully connected with **activists** for civil rights to call for a radical change in the ways in which algorithms are developed and deployed (Heilweil 2020)

A New Wave of Inquiry/2

There is a growing literature that is increasingly reflecting on these topics.

Perspectives from the social sciences are lagging behind (*what future for *us*?*).

Scholarly works encompass a myriad of different angles concerning the broad field of AI ethics: **formal methods to ensure fairness, legal approaches, philosophical reasoning, ...**

Three examples of harmful algorithms

- Racial bias in facial recognition
- Racial bias in predictive policing
- Algorithmic bias in Education

Racial bias in Facial Recognition

In 2018, a ground-breaking study found that **facial analysis algorithms misclassify Black women about 35% of the time**, with far lower error rates for white men (Buolamwini and Gebru 2018).

The study prompted a vivid debate on the devastating effects that such technologies may have on people's lives, especially when people belong to minorities.

The application of facial recognition tools is especially pervasive in *intelligence, security and policing* domains, where the consequences of certain decisions can ruin one's life forever.

Racial Bias in Predictive Policing

Predictive policing consists of a set of algorithmic techniques used to aid law enforcement agencies in forecasting *when* and *where* a crime will occur.

They had a tremendous success across the US in the last 10 years, but following protests and unclear practical results, agencies are starting to terminate their contracts.

Data that are used to train these algorithms are biased, and reflect systemic racist practices embedded in the society ⇒ **Feedback loops** reinforce the disproportionate targeting of minorities in the United States and in many other countries (Angwing et al. 2016; Lum and Isaac 2016; Heaven 2020).

Weaponized data are even more powerful mechanisms of control, oppression and surveillance in *non-democratic* countries.

Algorithmic bias in Education

Contemporary infamous case: *The 2020 UK A-level grading algorithm*

The algorithm used by the UK's Department for Education downgraded almost 40% of UK students compared to teacher estimations, additionally **favoring elite private schools** over public schools in the final grading

Nationwide protests led to a **U-turn**.

What does this tell to us?

These stories teach us several things.

What does this tell to us?

These stories teach us several things.

1. Algorithms that are fed with biased, corrupted, garbage data will lead to **biased, corrupted and garbage results**

What does this tell to us?

These stories teach us several things.

1. Algorithms that are fed with biased, corrupted, garbage data will lead to **biased, corrupted and garbage results**
2. The glittering veil of algorithmic applications should be **scrutinized**, always

What does this tell to us?

These stories teach us several things.

1. Algorithms that are fed with biased, corrupted, garbage data will lead to **biased, corrupted and garbage results**
2. The glittering veil of algorithmic applications should be **scrutinized**, always
3. **Algorithms are not neutral**. And AI calls scientists to embrace the political dimensions of their actions

What does this tell to us?

These stories teach us several things.

1. Algorithms that are fed with biased, corrupted, garbage data will lead to **biased, corrupted and garbage results**
2. The glittering veil of algorithmic applications should be **scrutinized**, always
3. **Algorithms are not neutral**. And AI calls scientists to embrace the political dimensions of their actions
4. Algorithms have **real-world effects**

What does this tell to us?

These stories teach us several things.

1. Algorithms that are fed with biased, corrupted, garbage data will lead to **biased, corrupted and garbage results**
2. The glittering veil of algorithmic applications should be **scrutinized**, always
3. **Algorithms are not neutral**. And AI calls scientists to embrace the political dimensions of their actions
4. Algorithms have **real-world effects**
5. As social scientists, **we need to play a role** in bridging communities working in this field and work towards fairer and more transparent AI

Final (I swear!) Remarks

A few last things before the end:

Final (I swear!) Remarks

A few last things before the end:

- **Know your data:** value the critical role of descriptive statistics and exploratory data analysis

Final (I swear!) Remarks

A few last things before the end:

- **Know your data:** value the critical role of descriptive statistics and exploratory data analysis
- **Know your problem:** ML will not answer all your questions

Final (I swear!) Remarks

A few last things before the end:

- **Know your data:** value the critical role of descriptive statistics and exploratory data analysis
- **Know your problem:** ML will not answer all your questions
- **Do not follow the hype:** do not try to impress the others and - foremost - do not try to impress yourself

Final (I swear!) Remarks

A few last things before the end:

- **Know your data:** value the critical role of descriptive statistics and exploratory data analysis
- **Know your problem:** ML will not answer all your questions
- **Do not follow the hype:** do not try to impress the others and - foremost - do not try to impress yourself
- **Learn to code:** if you are interested in the application of ML, there is no other path to take than learning to program

Resources

Works on AI ethics and related topics: Superintelligence: Path, Dangers, Strategies (Bostrom 2016); Race After Technology: Abolitionist Tools to the New Jim Code (Benjamin 2019); Automating Inequality (Eubanks 2019); Algorithms of Oppression: How Search Engines Reinforce Racism (Noble 2018), How Humans Judge Machines (Hidalgo et al. 2021); Hello World: How to be Human in the Age of the Machine (Fry 2018)

Scholars working on AI ethics that you should follow: Joy Buolamwini, Timnit Gebru, Rediet Abebe, Ben Green, Luciano Floridi, Nick Bostrom, Kate Crawford, Meredith Whittaker

Conferences to consider if interested in assessing state of the art at the intersection of AI, computing and social sciences: AAAI; KDD; ECML-PKDD; NetSci, Complex Networks; IEEE Big Data; AIES - Artificial Intelligence, Ethics and Society; ACM FAccT

Conferences to consider if interested in assessing state of the art at the intersection of AI, computing and social sciences: PNAS; Science Advances; Journal of Artificial Intelligence Research; Nature Human Behavior; Nature Scientific Reports; EPJ Data Science; Plos One; International Journal of Data Science and Analytics; Journal of Computational Social Science

Thanks for your attention!

email: gianmaria.campedelli@unitn.it
Twitter: @CampedelliGian

Bibliography I

Julia Angwing et al. (2016). “Machine Bias. There’s software used across the country to predict future criminals. And it’s biased against blacks.”. In: *ProPublica*. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

David Arthur and Sergei Vassilvitskii (2007). “K-means++: the advantages of careful seeding”. In: *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms.*

Ruha Benjamin (June 21, 2019). *Race After Technology: Abolitionist Tools for the New Jim Code*. 1. edizione. Medford, MA: Polity Pr. 285 pp. ISBN: 978-1-5095-2640-6.

Nick Bostrom (Apr. 1, 2016). *Superintelligence: Paths, Dangers, Strategies*. Reprint edizione. Oxford, United Kingdom ; New York, NY: Oxford. 432 pp. ISBN: 978-0-19-873983-8.

Bibliography II

Leo Breiman (Oct. 1, 2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (visited on 11/24/2020).

Joy Buolamwini and Timnit Gebru (Jan. 21, 2018). “Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification”. In: *Conference on Fairness, Accountability and Transparency*. Conference on Fairness, Accountability and Transparency. ISSN: 2640-3498. PMLR, pp. 77–91. URL: http://proceedings.mlr.press/v81/buolamwini18a.html (visited on 11/24/2020).

David Davies and Don Bouldin (May 1, 1979). “A Cluster Separation Measure”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-1, pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

Bibliography III

J. C. Dunn† (Jan. 1, 1974). "Well-Separated Clusters and Optimal Fuzzy Partitions". In: *Journal of Cybernetics* 4.1. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01969727408546059, pp. 95–104. ISSN: 0022-0280. DOI: 10.1080/01969727408546059. URL: https://doi.org/10.1080/01969727408546059 (visited on 11/24/2020).

Martin Ester et al. (Aug. 2, 1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, pp. 226–231. (Visited on 11/23/2020).

Virginia Eubanks (Aug. 6, 2019). *Automating Inequality*. Reprint edizione. Picador. 287 pp. ISBN: 978-1-250-21578-9.

Hannah Fry (Sept. 6, 2018). *Hello World: How to be Human in the Age of the Machine*. London New York Toronto Sidney Auckland: Doubleday. 256 pp. ISBN: 978-0-85752-524-6.

Bibliography IV

Will Douglas Heaven (July 2020). "Predictive policing algorithms are racist. They need to be dismantled". In: *MIT Technology Review*.

Rebecca Heilweil (June 11, 2020). "Big tech companies back away from selling facial recognition to police. That's progress.". In: *Vox*. URL: https://www.vox.com/recode/2020/6/10/21287194/amazon-microsoft-ibm-facial-recognition-moratorium-police.

Cesar A. Hidalgo et al. (Feb. 2, 2021). *How Humans Judge Machines*. The MIT Press. 224 pp.

S. Lloyd (Mar. 1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2. Conference Name: IEEE Transactions on Information Theory, pp. 129–137. ISSN: 1557-9654. DOI: 10.1109/TIT.1982.1056489.

Bibliography V

Kristian Lum and William Isaac (2016). “To predict and serve?” In: *Significance* 13.5. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1740-9713.2016.00960.x, pp. 14–19. ISSN: 1740-9713. DOI: 10.1111/j.1740-9713.2016.00960.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2016.00960.x (visited on 09/23/2020).

Safiya Umoja Noble (Feb. 20, 2018). *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York: New York Univ Pr. 229 pp. ISBN: 978-1-4798-3724-3.

Artwork Reference

Title Slide: *Peinture (Etoile Bleue), Joan Miro (1927)*

Algorithms *Structural Constellation "To Ferdinand Hodler", Josef Albers (1954)*

Societal Implications of AI *Anatomy of an AI System, Kate Crawford and Vladan Joler (2018)*

Resources *G2, Jim Drain (2003)*

Closing Slide: *The Persistence of Memory, Salvador Dalì (1931)*