Lecture 1: 9 January, 2024

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
January–April 2024

What is this course about?

Data Mining

- Identify "hidden" patterns in data
- Also data collection, cleaning, uniformization, storage
  - Won't emphasize these aspects

Machine Learning

- "Learn" mathematical models of processes from data
- Supervised learning — learn from experience
- Unsupervised learning — search for structure

Supervised Learning

Extrapolate from historical data

- Predict board exam scores from model exams
- Should this loan application be granted?
- Do these symptoms indicate CoViD-19?

“Manually” labelled historical data is available

- Past exam scores: model exams and board exam
- Customer profiles: age, income, . . . , repayment/default status
- Patient health records, diagnosis

Historical data → model to predict outcome

Supervised learning ...

What are we trying to predict?

Numerical values

- Board exam scores
- House price (valuation for insurance)
- Net worth of a person (for loan eligibility)

Categories

- Email: is this message junk?
- Insurance claim: pay out, or check for fraud?
- Credit card approval: reject, normal, premium

Supervised learning . . .

How do we predict?

- Build a mathematical model
  - Different types of models
  - Parameters to be tuned
- Fit parameters based on input data
  - Different historical data produces different models
  - e.g., each user's junk mail filter fits their individual preferences
- Study different models, how they are built from historical data

Unsupervised learning

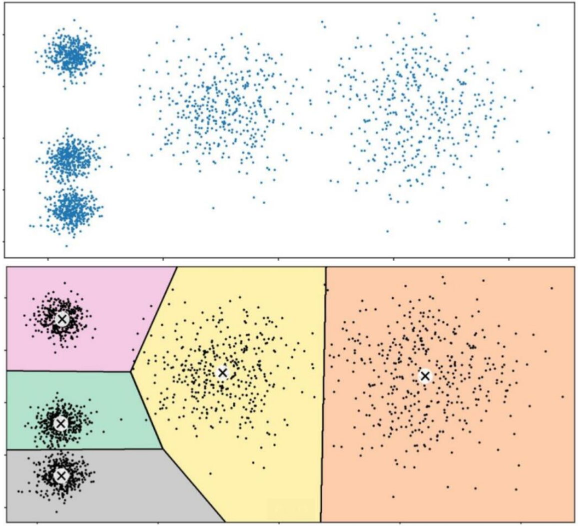
- Supervised learning builds models to reconstruct "known" patterns given by historical data
- Unsupervised learning tries to identify patterns without guidance

Customer segmentation

- Different types of newspaper readers
- Age vs product profile of retail shop customers
- Viewer recommendations on video platform

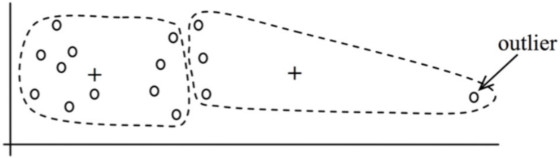
Clustering

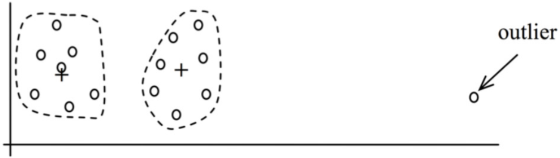
- Organize data into "similar" groups — clusters
- Define a similarity measure, or distance function
- Clusters are groups of data items that are "close together"



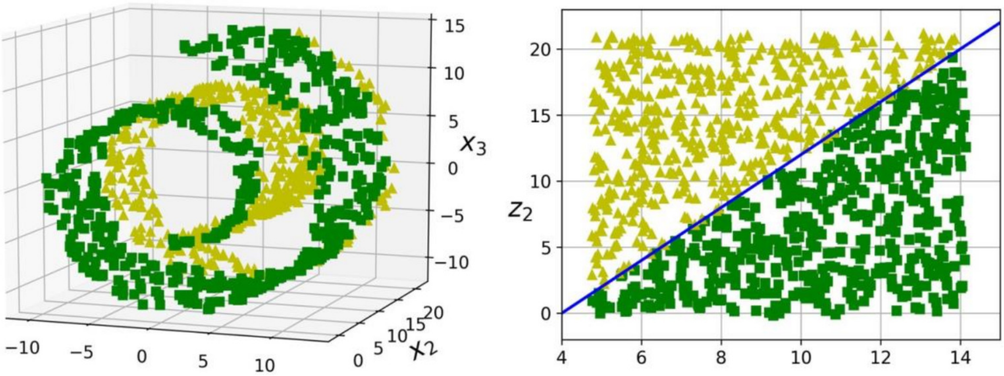
Outliers

- Outliers are anomalous values
  - Net worth of Jeff Bezos, Mukesh Ambani
- Outliers distort clustering and other analysis
- How can we identify outliers?



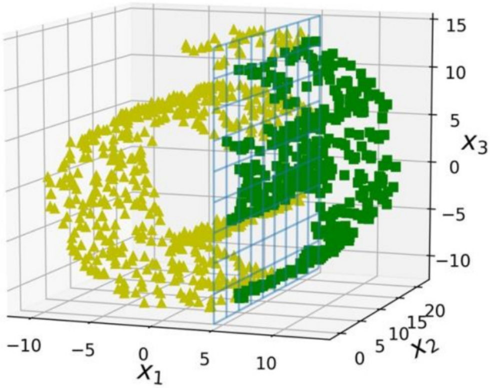


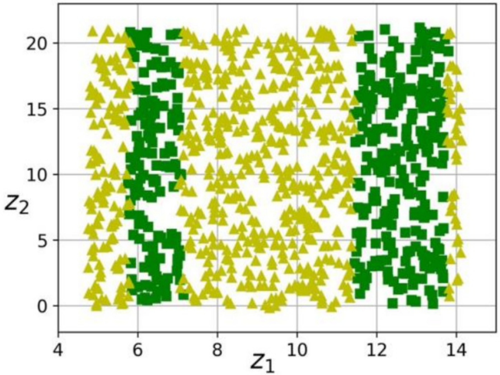
Dimensionality reduction



Preprocessing for supervised learning

Need not be a good idea — perils of working blind!





Summary

Machine Learning

- Supervised learning
  - Build predictive models from historical data
- Unsupervised learning
  - Search for structure
  - Clustering, outlier detection, dimensionality reduction

If intelligence were a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, . . .

Yann Le Cun, ACM Turing Award 2018

Market-Basket Analysis

- People who buy X also tend to buy Y
- Rearrange products on display based on customer patterns
  - The diapers and beer legend
  - The true story, `http://www.dssresources.com/newsletters/66.php`
- Applies in more abstract settings
  - Items are concepts, basket is a set of concepts in which a student does badly
    - Students with difficulties in concept A also tend to misunderstand concept B
  - Items are words, transactions are documents

Formal setting

- Set of items $I = \{i_1, i_2, \ldots, i_N\}$
- A transaction is a set $t \subseteq I$ of items
- Set of transactions $T = \{t_1, t_2, \ldots, t_M\}$
- Identify association rules $X \to Y$
  - $X, Y \subseteq I$, $X \cap Y = \emptyset$
  - If $X \subseteq t_j$ then it is likely that $Y \subseteq t_j$
- Two thresholds
  - How frequently does $X \subseteq t_j$ imply $Y \subseteq t_j$?
  - How significant is this pattern overall?

Setting thresholds

- For $Z \subseteq I$, $Z.\text{count} = |\{t_j \mid Z \subseteq t_j\}|$
- How frequently does $X \subseteq t_j$ imply $Y \subseteq t_j$?
  - Fix a confidence level χ
  - Want $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
- How significant is this pattern overall?
  - Fix a support level σ
  - Want $\dfrac{(X \cup Y).count}{M} \geq \sigma$
- Given sets of items I and transactions T, with confidence χ and support σ, find all valid association rules $X \to Y$

Frequent itemsets

- $X \rightarrow Y$ is interesting only if $(X \cup Y).\text{count} \geq \sigma \cdot M$
- First identify all frequent itemsets
  - $Z \subseteq I$ such that $Z.\text{count} \geq \sigma \cdot M$
- Naïve strategy: maintain a counter for each Z
  - For each $t_j \in T$
    For each $Z \subseteq t_j$
    Increment the counter for Z
  - After scanning all transactions, keep Z with $Z.\text{count} \geq \sigma \cdot M$
- Need to maintain $2^{|I|}$ counters
  - Infeasible amount of memory
  - Can we do better?

Sample calculation

- Let's assume a bound on each $t_i \in T$
  - No transacation has more than 10 items
- Say $N = |I| = 10^6$, $M = |T| = 10^9$, $\sigma = 0.01$
  - Number of possible subsets to count is $\displaystyle\sum_{i=1}^{10} \binom{10^6}{i}$
- A singleton subset that is frequent is an item that appears in at least 10^7 transactions
- Totally, T contains at most 10^{10} items
- At most $10^{10}/10^7 = 1000$ items are frequent!
- How can we exploit this?

Apriori

- Clearly, if Z is frequent, so is every subset $Y \subseteq Z$
- We exploit the contrapositive

Apriori observation

If Z is not a frequent itemset, no superset $Y \supseteq Z$ can be frequent

- For instance, in our earlier example, every frequent itemset must be built from the 1000 frequent items
- In particular, for any frequent pair $\{x, y\}$, both $\{x\}$ and $\{y\}$ must be frequent
- Build frequent itemsets bottom up, size 1,2,...

Apriori algorithm

- F_i : frequent itemsets of size i — Level i
- F_1: Scan T, maintain a counter for each $x \in I$
- $C_2 = \{\{x, y\} \mid x, y \in F_1\}$: Candidates in level 2
- F_2: Scan T, maintain a counter for each $X \in C_2$
- $C_3 = \{\{x, y, z\} \mid \{x, y\}, \{x, z\}, \{y, z\} \in F_2\}$
- F_3: Scan T, maintain a counter for each $X \in C_3$
- ...
- C_k = subsets of size k, every $(k-1)$-subset is in F_{k-1}
- F_k: Scan T, maintain a counter for each $X \in C_k$
- ...