X-Informatics Case Study: e-Commerce and Life Style Informatics: Recommender Systems III: Item-based Collaborative Filtering and its Technologies

June 18 2013

Geoffrey Fox

gcf@indiana.edu

http://www.infomall.org/X-InformaticsSpring2013/index.html

Associate Dean for Research, School of Informatics and Computing

Indiana University Bloomington

2013

Big Data Ecosystem in One Sentence

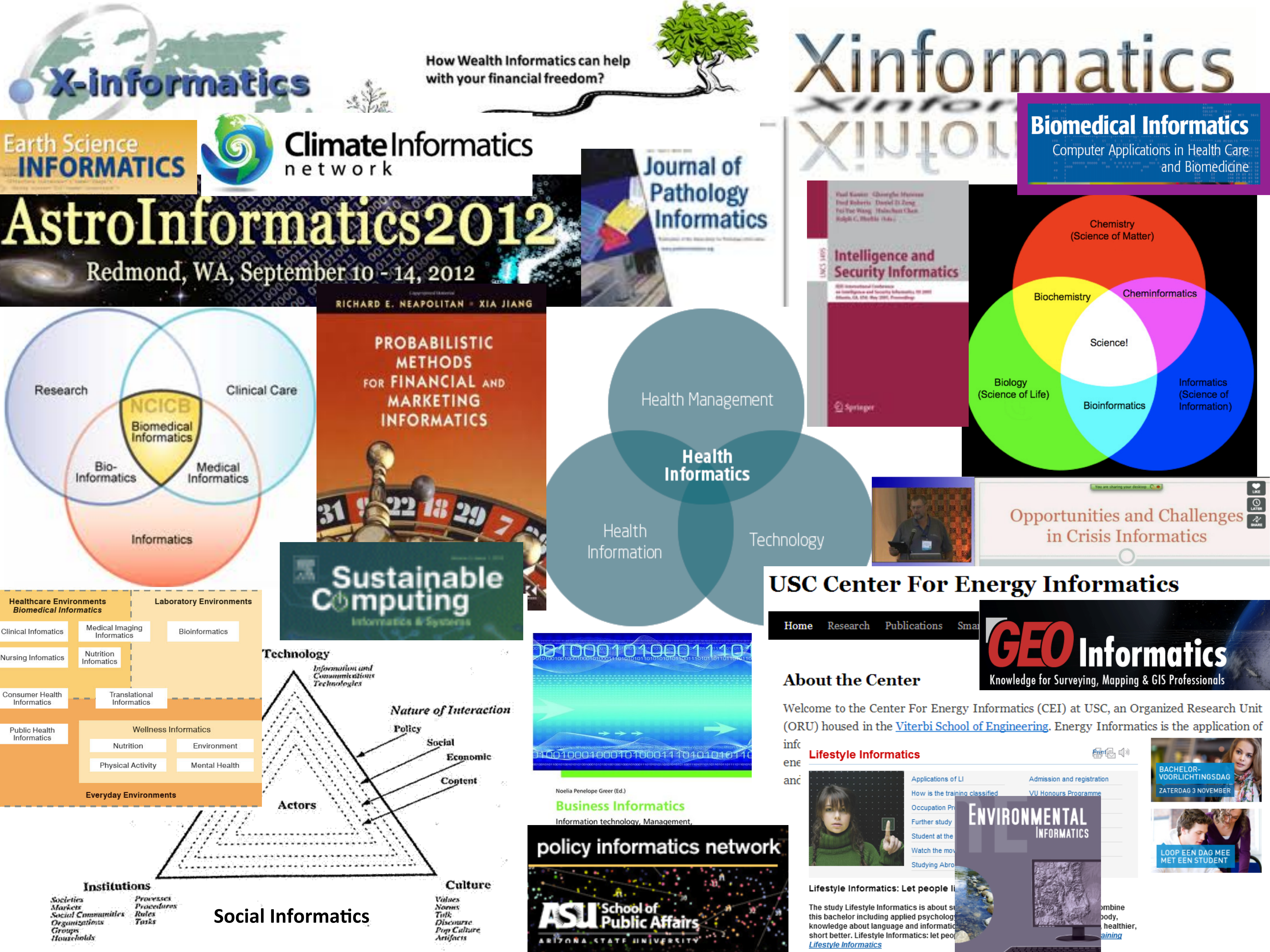
Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics (or e-X)

X = Astronomy, Biology, Biomedicine, Business, Chemistry, Climate, Crisis, Earth Science, Energy, Environment, Finance, Health, Intelligence, Lifestyle, Marketing, Medicine, Pathology, Policy, Radar, Security, Sensor, Social, Sustainability, Wealth and Wellness with more fields (physics) defined implicitly

Spans Industry and Science (research)

Education: Data Science see recent New York Times articles

http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/


X-informatics
How Wealth Informatics can help with your financial freedom?
Xinformatics
Earth Science INFORMATICS
Climate Informatics network
Journal of Pathology Informatics
Biomedical Informatics
Computer Applications in Health Care and Biomedicine
AstroInformatics2012
Redmond, WA, September 10 - 14, 2012
Intelligence and Security Informatics
Chemistry (Science of Matter)
Biochemistry
Cheminformatics
Science!
Biology (Science of Life)
Bioinformatics
Informatics (Science of Information)
Research
Clinical Care
NCICB
Biomedical Informatics
Bio-Informatics
Medical Informatics
Informatics
RICHARD E. NEAPOLITAN · XIA JIANG
PROBABILISTIC METHODS FOR FINANCIAL AND MARKETING INFORMATICS
Health Management
Health Informatics
Health Information
Technology
Opportunities and Challenges in Crisis Informatics
Sustainable Computing
Informatics & Systems
USC Center For Energy Informatics
Home
Research
Publications
GEO Informatics
Knowledge for Surveying, Mapping & GIS Professionals
About the Center
Welcome to the Center For Energy Informatics (CEI) at USC, an Organized Research Unit (ORU) housed in the Viterbi School of Engineering. Energy Informatics is the application of
Healthcare Environments
Biomedical Informatics
Laboratory Environments
Clinical Infomatics
Medical Imaging Informatics
Bioinformatics
Nursing Infomatics
Nutrition Infomatics
Consumer Health Informatics
Translational Informatics
Public Health Informatics
Wellness Informatics
Nutrition
Environment
Physical Activity
Mental Health
Everyday Environments
Technology
Information and Communications Technologies
Nature of Interaction
Policy
Social
Economic
Content
Actors
Institutions
Culture
Social Informatics
Noelia Penelope Greer (Ed.)
Business Informatics
Information technology, Management,
policy informatics network
ASU School of Public Affairs
ARIZONA STATE UNIVERSITY
Lifestyle Informatics
Applications of LI
Admission and registration
How is the training classified
VU Honours Programme
Lifestyle Informatics: Let people li
ENVIRONMENTAL INFORMATICS
BACHELOR-VOORLICHTINGSDAG
ZATERDAG 3 NOVEMBER
LOOP EEN DAG MEE MET EEN STUDENT

Item-based Collaborative Filtering

Memory-based and model-based approaches

- **User-based CF is said to be "memory-based“ (Real-Time)**
  - the rating matrix is directly used to find neighbors / make predictions
  - does not scale for most real-world scenarios
  - large e-commerce sites have tens of millions of customers and millions of items
- **Model-based approaches (Batch or Off Line)**
  - based on an offline pre-processing or "model-learning" phase
  - at run-time, only the learned model is used to make predictions
  - models are updated / re-trained periodically
  - large variety of techniques used
  - model-building and updating can be computationally expensive
  - *item*-based CF is an example for model-based approaches

Item-based collaborative filtering

- **Basic idea:**
  – Use the similarity between items (and not users) to make predictions
- **Example:**
  – Look for items that are similar to Item5
  – Take Alice's ratings for these items to predict the rating for Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

The cosine similarity measure

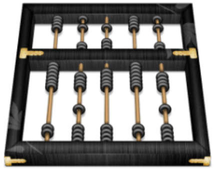
Item based collaborative filtering

- Produces better results in item-to-item filtering
- Ratings are seen as vector in M-dimensional space

M is number of users

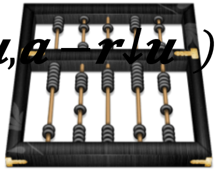
- Similarity is calculated based on the angle between the vectors

$$sim(a,b) = \vec{a} \cdot \vec{b} / |\vec{a}| * |\vec{b}|$$



- Adjusted cosine similarity
  - take average user ratings into account, transform the original ratings
  - U: set of users who have rated both items a and b

$$sim(a,b) = \sum_{u \in U} (r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u) / \sqrt{\sum_{u \in U} (r_{u,a} - \bar{r}_u)^2}$$

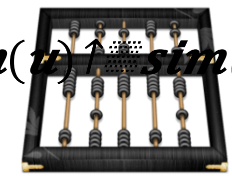


Making predictions

Item based collaborative filtering

- A common prediction function:

$$pred(u,p) = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)}$$



- Neighborhood size is typically also limited to a specific size
- Not all neighbors are taken into account for the prediction
- An analysis of the MovieLens dataset indicates that "in most real-world situations, a neighborhood of 20 to 50 neighbors seems reasonable" (Herlocker et al. 2002)

Pre-processing for item-based filtering

- **Item-based filtering does not solve the scalability problem itself**
- **Pre-processing approach by Amazon.com (in 2003)**
  - Calculate all pair-wise item similarities in advance
  - The neighborhood to be used at run-time is typically rather small, because only items are taken into account which the user has rated
  - Item similarities are supposed to be more stable than user similarities
- **Memory requirements**
  - Up to N^2 pair-wise similarities to be memorized (N = number of items) in theory
  - In practice, this is significantly lower (items with no co-ratings)
  - Further reductions possible
    - Minimum threshold for co-ratings
    - Limit the neighborhood size (might affect recommendation accuracy)

More on ratings – Explicit ratings

- Probably the most precise ratings
- Most commonly used (1 to 5, 1 to 7 Likert response scales)
- Research topics
  - Optimal granularity of scale; indication that 10-point scale is better accepted in movie dom.
  - An even more fine-grained scale was chosen in the joke recommender discussed by Goldberg et al. (2001), where a continuous scale (from −10 to +10) and a graphical input bar were used
    - No precision loss from the discretization
    - User preferences can be captured at a finer granularity
    - Users actually "like" the graphical interaction method
  - Multidimensional ratings (multiple ratings per movie such as ratings for actors and sound)
- Main problems
  - Users not always willing to rate many items
    - number of available ratings could be too small → sparse rating matrices → poor recommendation quality
  - How to stimulate users to rate more items?

More on ratings – Implicit ratings

- Typically collected by the web shop or application in which the recommender system is embedded
- When a customer buys an item, for instance, many recommender systems interpret this behavior as a positive rating
- Clicks, page views, time spent on some page, demo downloads …
- Implicit ratings can be collected constantly and do not require additional efforts from the side of the user
- Main problem
  - One cannot be sure whether the user behavior is correctly interpreted
  - For example, a user might not like all the books he or she has bought; the user also might have bought a book for someone else
- Implicit ratings can be used in addition to explicit ones; question of correctness of interpretation

Data sparsity problems

- **Cold start problem**
  - How to recommend new items? What to recommend to new users?
- **Straightforward approaches**
  - Ask/force users to rate a set of items
  - Use another method (e.g., content-based, demographic or simply non-personalized) in the initial phase
  - Default voting: assign default values to items that only one of the two users to be compared has rated (Breese et al. 1998)
- **Alternatives**
  - Use better algorithms (beyond nearest-neighbor approaches)
  - Example:
    - In nearest-neighbor approaches, the set of sufficiently similar neighbors might be too small to make good predictions
    - Assume "transitivity" of neighborhoods

If A near B and B near C
Then A near C even if not rated by common users

Example algorithms for sparse datasets

Example of Transitivity

- **Recursive CF** (Zhang and Pu 2007)
  - Assume there is a very close user neighbor n of u who however has not rated the target item i yet.
  - Idea:
    - Apply CF-method recursively and predict a rating for item i for the neighbor
    - Use this predicted rating instead of the rating of a more distant direct neighbor

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	?
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

sim = 0.85

Predict rating for User1

Technologies:
k Nearest Neighbors and High Dimensional Spaces

k Nearest Neighbors

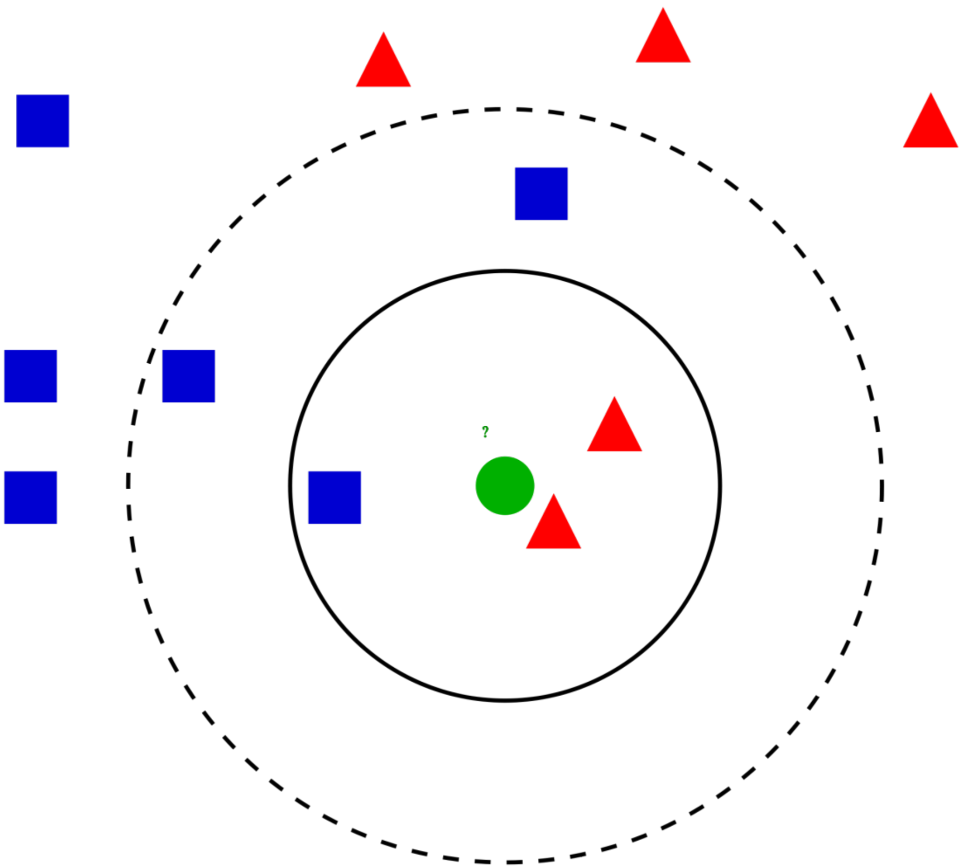
- The basic user and item based recommendation systems built around k Nearest Neighbor method
- Set k = 3…20 and find the k nearest neighbors
- If k too small then sensitive to error (maybe one of the k is somehow screwed up)
- The algorithm can be trivial
- Calculate all the distances (user i – Alice) where all users including Alice have rated some cutoff number of identical items
- Sort distances lowest to highest
- Take the lowest k
- See Python code

Some Issues

- Users and Items each live in **abstract spaces**
- Users in space of items with rankings as coefficients
- Items in space of users with rankings as coefficient
- OR in content-based, items live in space defined by their content
  - Words
  - Clusters or “latent factors” or “mixtures” or topics
  - Semantic information as in Music Genome
- How do we easily find nearest neighbors?
- Take user-based case
- The trivial algorithm needs O(M) distance computations for M customers and then O(M_{real}(Alice)log{M_{real}(Alice)}) where M_{real}(Alice) users are considered as have rated enough items in common.
- Sorting P numbers takes time P logP
- If k small, complexity is k M_{real}(Alice)
- Note M_{real}(Alice) << M

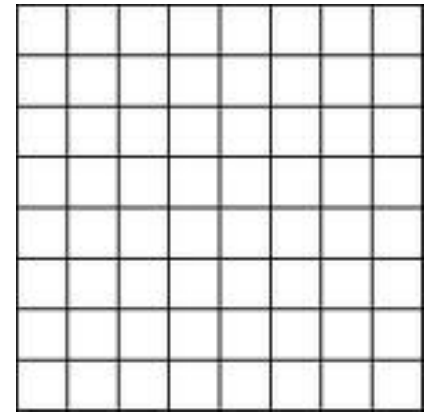
Wikipedia:
Example of *k*-NN classification

- The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If *k = 3*(solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If *k = 5*(dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

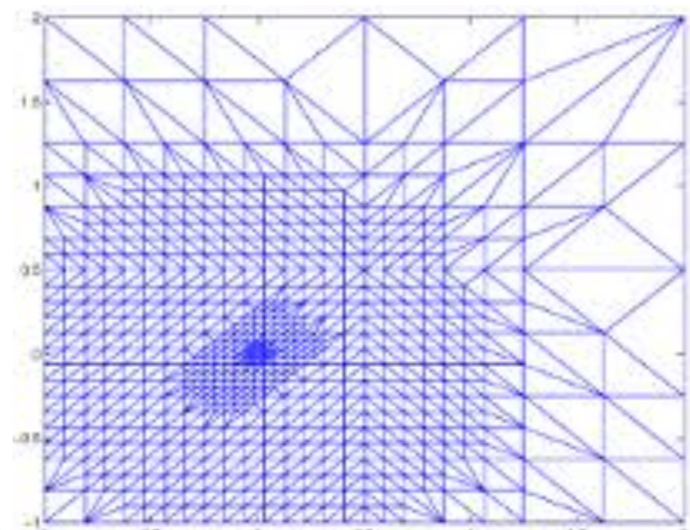


The Curse of Dimensionality

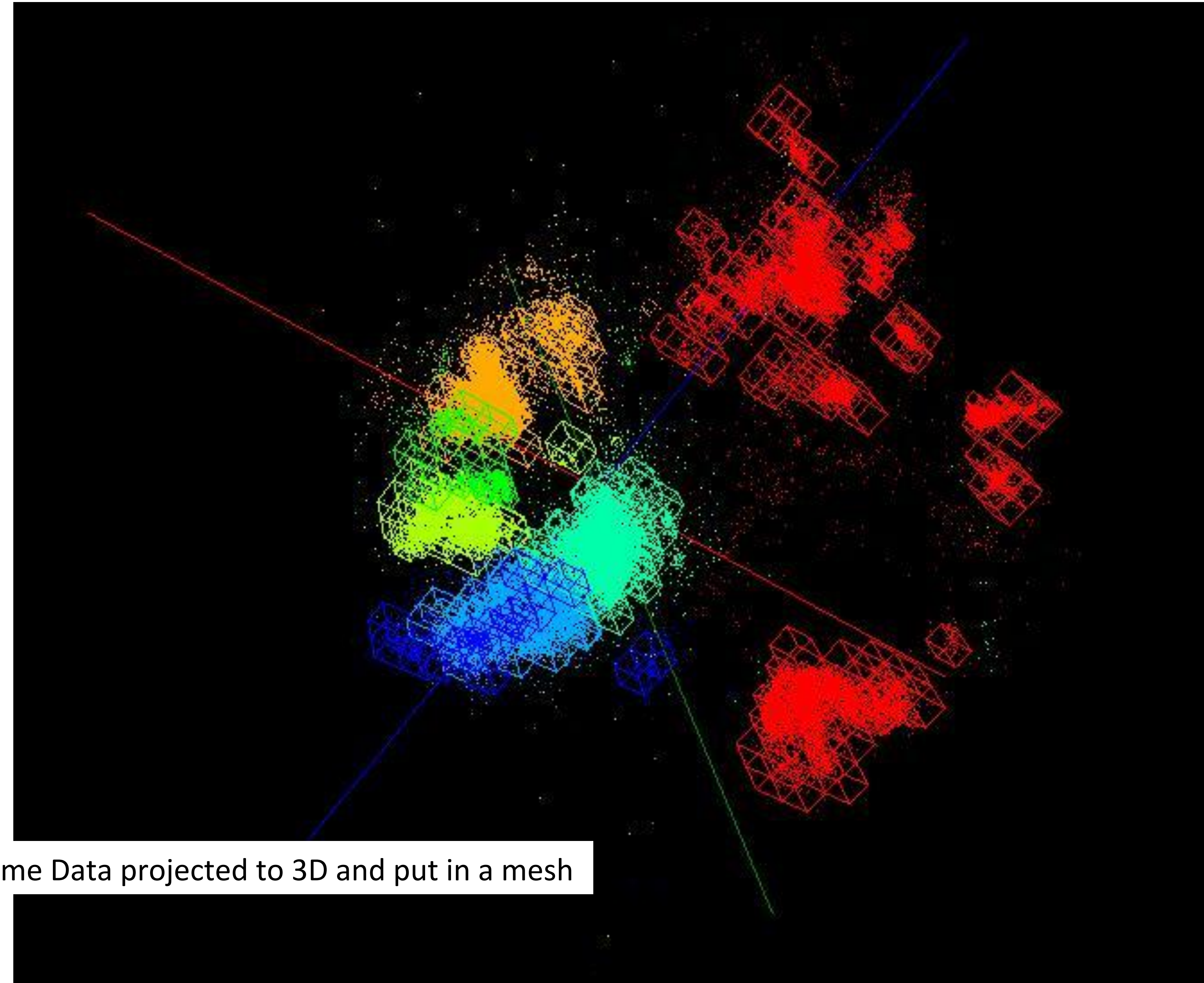
- In a low dimension space it is very fast to find nearest neighbors



- Divide space into a grid
- Assign points to cells of grid
- The mesh can be non uniform
- Just look at same cell point is in or its neighbors to find k nearest neighbors



- Doesn't work for "abstract" or high dimension spaces as too many cells
- Factor analysis or clustering or dimension reduction addresses this



Genome Data projected to 3D and put in a mesh