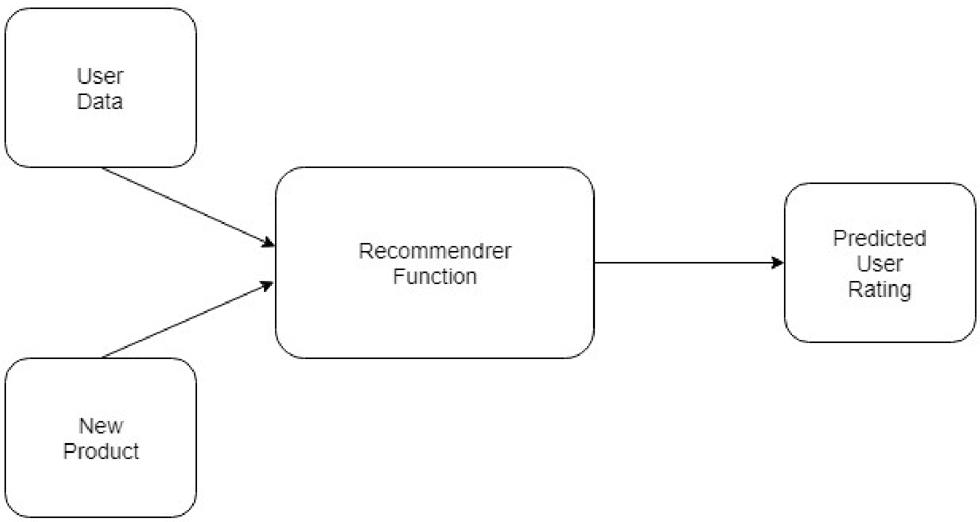
1. RECOMMENDER SYSTEMS

Recommender systems are so commonplace now that many of us use them without even knowing it. Because we can't possibly look through all the products or content on a website, a recommendation system plays an important role in helping us have a better user experience

Some examples of recommender systems in action include product recommendations on Amazon, Netflix suggestions for movies and TV shows in your feed, recommended videos on YouTube, music on Spotify, the Facebook newsfeed and Google Ads



HOW DO RECOMMENDER SYSTEMS WORK

UNDERSTANDING RELATIONSHIPS

User-Product Relationship

The user-product relationship occurs when some users have a preference towards specific products that they need. For example, a cricket player might have a preference for cricket-related items, thus the e-commerce website will build a user-product relation of player->cricket.

Product-Product Relationship

Product-product relationships occur when items are similar in nature, either by appearance or description. Some examples include books or music of the same genre, dishes from the same cuisine, or news articles from a particular event.

User-User Relationship

User-user relationships occur when some customers have similar taste with respect to a particular product or service. Examples include mutual friends, similar backgrounds, similar age, etc.

DATA & RECOMMENDER SYSTEMS

In addition to relationships, recommender systems utilize the following kinds of data:

User Behavior Data
Users behavior data is useful information about the engagement of the user on the product. It can be collected from ratings, clicks and purchase history.

User Demographic Data
User demographic information is related to the user's personal information such as age, education, income and location.

Product Attribute Data
Product attribute data is information related to the product itself such as genre in case of books, cast in case of movies, and cuisine in case of food.

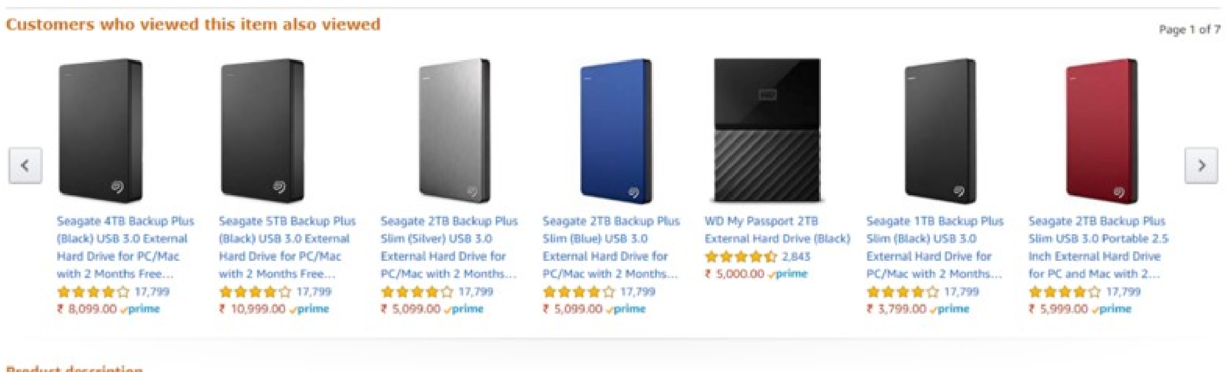
There are two particularly important methods, explicit and implicit rating.

Explicit Ratings
Explicit ratings are provided by the user. They infer the user's preference. Examples include star ratings, reviews, feedback, likes and following. Since users don't always rate products, explicit ratings can be hard to get.

Implicit Ratings
Implicit ratings are provided when users interact with the item. They infer a user's behavior and are easy to get as users are subconsciously clicking. Examples include clicks, views and purchases.

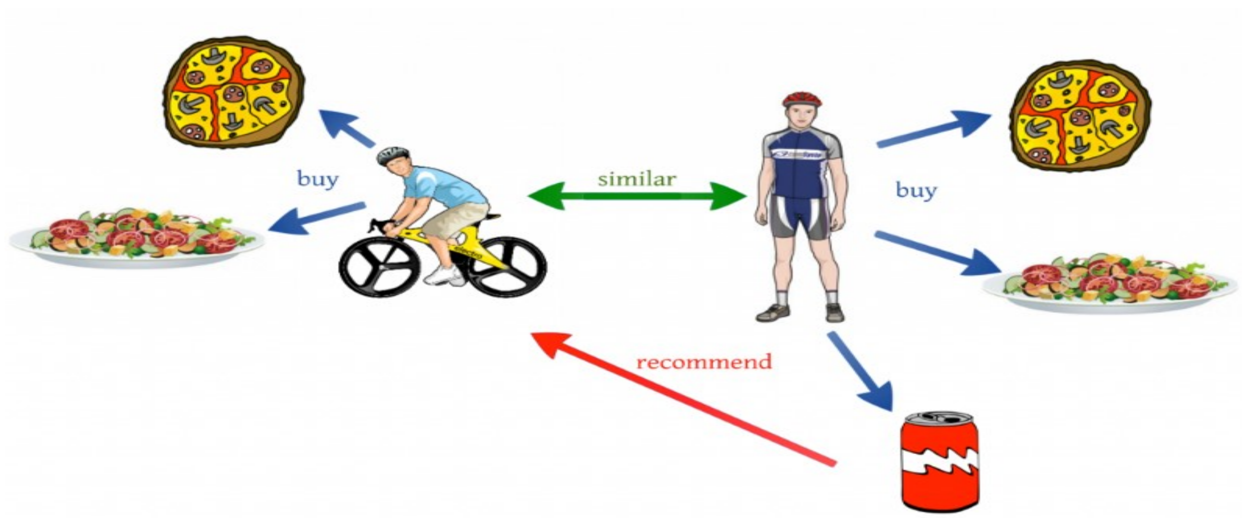
Product Similarity (Item-Item Filtering)

Product similarity is the most useful system for suggesting products based on how much the user would like the product. If the user is browsing or searching for a particular product, they can be shown similar products. Users often expect to find products they want quickly and move on if they have a hard time finding the relevant product. When the user clicks on one product we can show another similar product, or if the user buys the product we can email the user advertisements or coupons based on a similar product.



User Similarity (User-User Filtering)

User similarity is for checking the difference between the similarity of two users. If two users have similar preferences for a product we can assume they have similar interests. It's like a friend recommending a product.



Similarity Measures

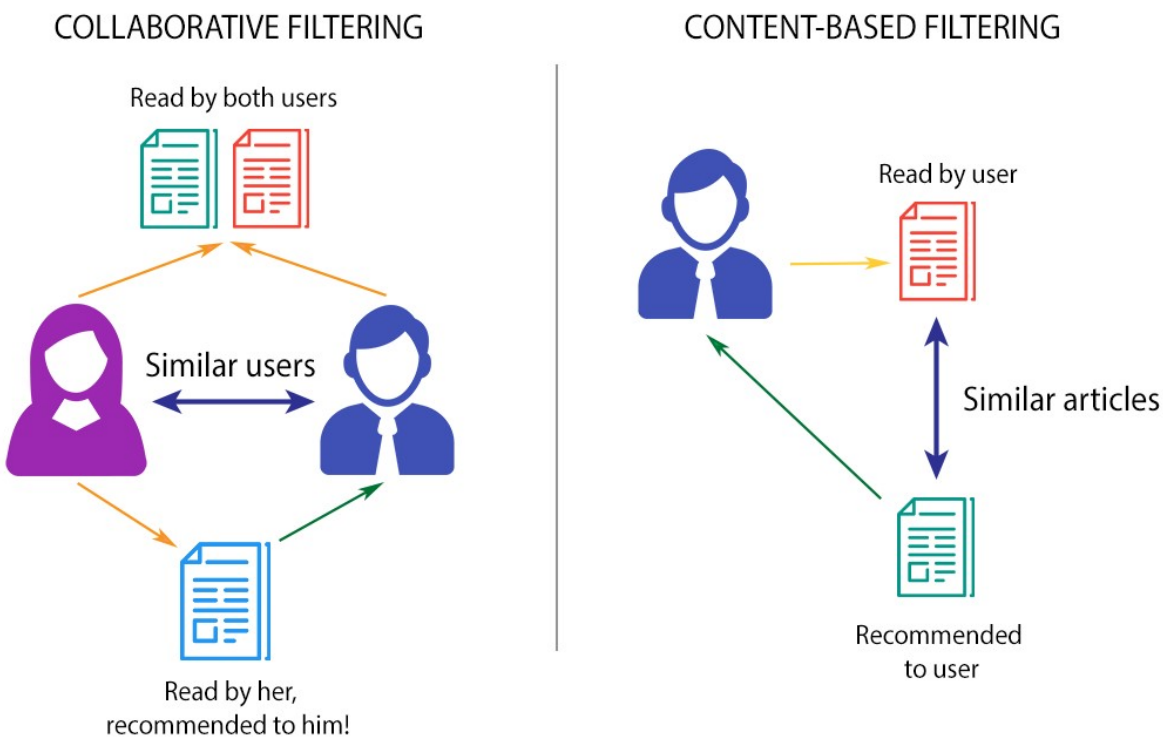
Minkowski Distance : $(\sum_{i=1}^{n} |X_i - Y_i|^p)^{1/p}$

Manhattan Distance $d = \sum_{i=1}^{n} |X_i - Y_i|$

Euclidean Distance $d = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$

Pearson Coefficient

APPROACHES TO RECOMMENDER SYSTEMS



CONTENT BASED FILTERING RECOMMENDER

Content-based recommendation systems uses their knowledge about each product to recommend new ones. Recommendations are based on attributes of the item. Content-based recommender systems work well when descriptive data on the content is provided beforehand. “Similarity” is measured against product attributes.

Suppose I watch a movie in a particular genre, then I will be recommended movies within that specific genre. The movie's attributes, like title, year of release, director and cast, are also helpful in identifying similar movie content.

Movies	Reviews Given	Rating
Mission Imposible	✓	Good
James Bond	✓	Good
Toy Story	✓	Bad

COLLABORATIVE FILTERING RECOMMENDER

Collaborative filtering recommender makes suggestions based on how users rated in the past and not based on the product themselves. It only knows how other customers rated the product. “Similarity” is measured against the similarity of users.

2. IMAGE CLASSIFICATION

Classification between objects is a fairly easy task for us, but it has proved to be a complex one for machines and therefore image classification has been an important task within the field of computervision. Image classification refers to the labeling of images into one of a number of predefined classes.

Some examples of image classification include:

- Labeling an x-ray as cancer or not (binary classification).
- Classifying a handwritten digit (multiclass classification).
- Assigning a name to a photograph of a face (multiclass classification).

The advancements in the field of autonomous driving also serve as a great example of the use of image classification in the real-world. For example, we can build an image classification model that recognizes various objects, such as other vehicles, pedestrians, traffic lights, and signposts on the road.

Structure of an Image Classification Task

- **Image Preprocessing** - The aim of this process is to improve the image data(features) by suppressing unwanted distortions and enhancement of some important image features so that our Computer Vision models can benefit from this improved data to work on.
- **Detection of an object** - Detection refers to the localization of an object which means the segmentation of the image and identifying the position of the object of interest.
- **Feature extraction and Training**- This is a crucial step wherein statistical or deep learning methods are used to identify the most interesting patterns of the image, features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes. This process where the model learns the features from the dataset is called model training.
- **Classification of the object** - This step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns.

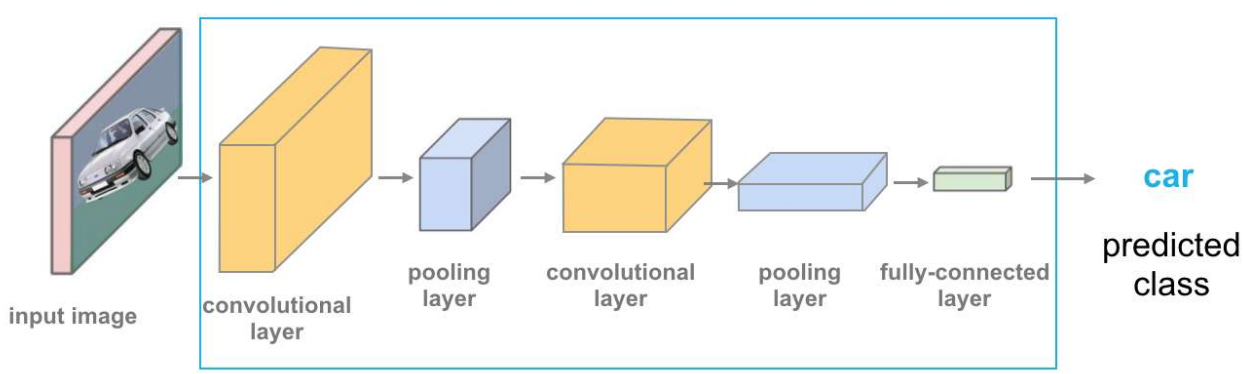


Image Classification Techniques

We will start with some statistical machine learning classifiers like Support Vector Machine and Decision Tree and then move on to deep learning architectures like Convolutional Neural Networks.

Performance evaluation

CLASSIFIER	ACCURACY	PRECISION	RECALL	ROC
SVM	85.68%	0.86	0.87	0.86
Decision Trees	84.61%	0.85	0.84	0.82
KNN	86.32%	0.86	0.86	0.88
ANN(for 100 epochs)	83.10%	0.88	0.87	0.88
CNN(for 300 epochs)	91.11%	0.93	0.89	0.97

3. SOCIAL NETWORK GRAPHS

The essential characteristics of a social network are:

There is a collection of entities that participate in the network. Typically, these entities are people

There is at least one relationship between entities of the network. On Facebook or its ilk, this relationship is called friends. Sometimes the relationship is all-or-nothing; two people are either friends or they are not. However, in other examples of social networks, the relationship has a degree.

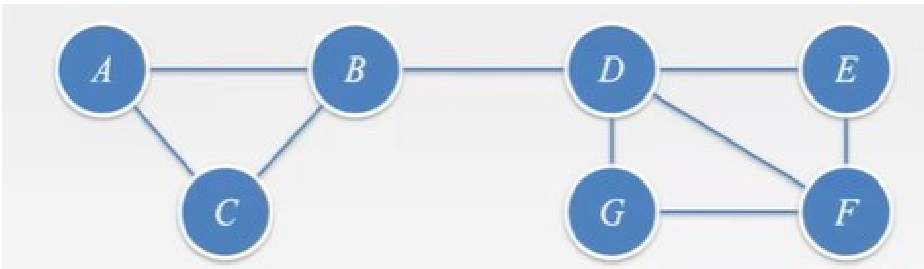
There is an assumption of nonrandomness or locality. This condition is the hardest to formalize.

That is, if entity A is related to both B and C, then there is a higher probability than average that B and C are related

Social Networks as Graphs

Social networks are naturally modeled as graphs, which we sometimes refer to as a social graph. The entities are the nodes, and an edge connects two nodes if the nodes are related by the relationship that characterizes the network. If there is a degree associated with the relationship, this degree is represented by labeling the edges.

Figure is an example of a tiny social network. The entities are the nodes A through G. The relationship, which we might think of as “friends,” is represented by the edges. For instance, B is friends with A, C, and D.



Varieties of Social Networks

Telephone Networks

Here the nodes represent phone numbers, which are really individuals. There is an edge between two nodes if a call has been placed between those phones in some fixed period of time, such as last month, or “ever.” The edges could be weighted by the number of calls made between these phones during the period.

Email Networks

The nodes represent email addresses, which are again individuals. An edge represents the fact that there was at least one email in at least one direction between the two addresses. Alternatively, we may only place an edge if there were emails in both directions. In that way, we avoid viewing spammers as “friends” with all their victims. Another approach is to label edges as weak or strong. Strong edges represent communication in both directions, while weak edges indicate that the communication was in one direction only.

Collaboration Networks

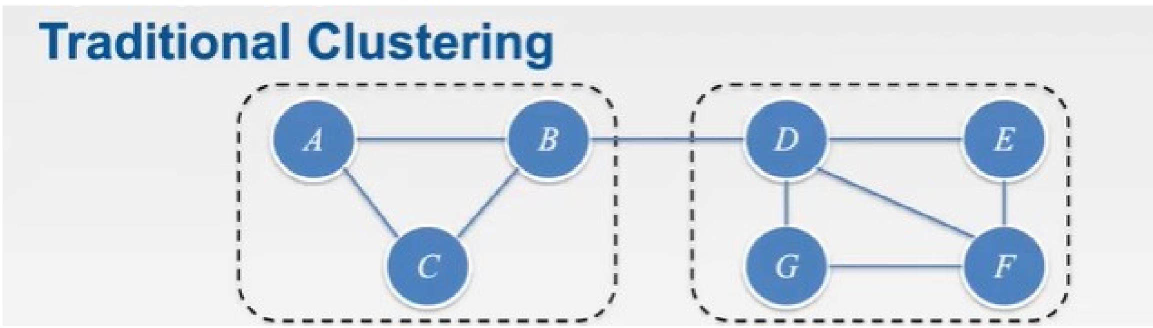
Nodes represent individuals who have published research papers. There is an edge between two individuals who published one or more papers jointly. Optionally, we can label edges by the number of joint publications. The communities in this network are authors working on a particular topic.

An alternative view of the same data is as a graph in which the nodes are papers. Two papers are connected by an edge if they have at least one author in common. Now, we form communities that are collections of papers on the same topic.

Clustering of Social-Network Graphs

Distance Measures for Social-Network Graphs

If we were to apply standard clustering techniques to a social-network graph, our first step would be to define a distance measure. combine nodes which are nearby. Repeating same process can from clusters



Traditional clustering includes two communities. Likely to put two nodes with small distance in the same cluster. Social networks graphs would have cross community edges. Severe merging of communities likely.

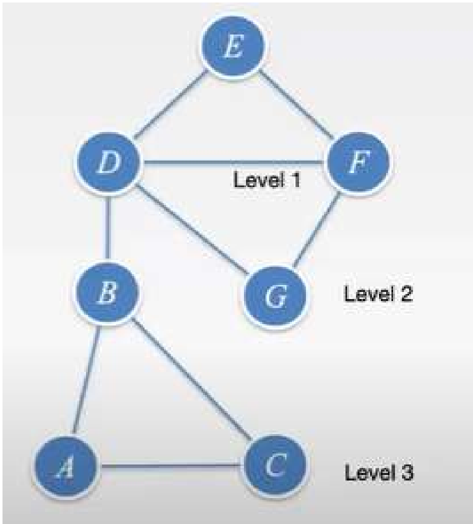
Other approach

The Girvan-Newman Algorithm

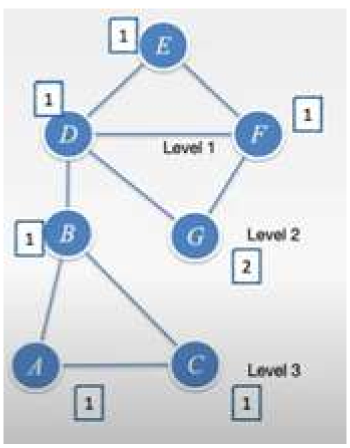
In order to exploit the betweenness of edges, we need to calculate the number of shortest paths going through each edge. We shall describe a method called the Girvan-Newman (GN) Algorithm, which visits each node X once and computes the number of shortest paths from X to each of the other nodes that go through each of the edges.

The algorithm begins by performing a breadth-first search (BFS) of the graph, starting at the node X.

The level of each node in the BFS presentation is the length of the shortest path from X to that node.



The second step of the GN algorithm is to label each node by the number of shortest paths that reach it from the root. Start by labeling the root 1.

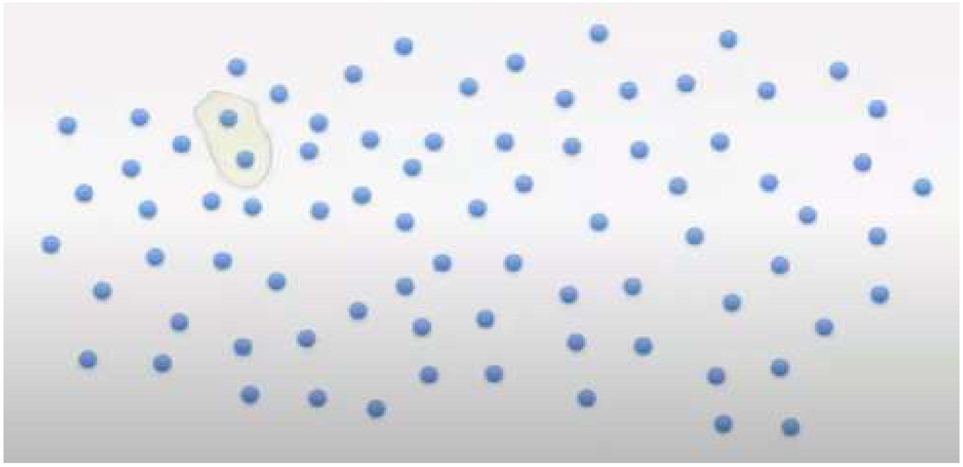


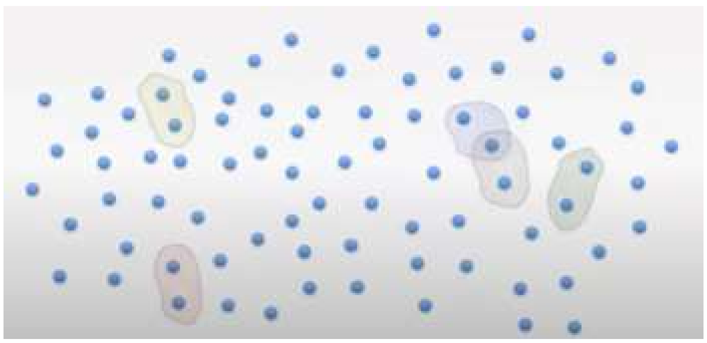
The third and final step is to calculate the credit value of node. Credit of node is calculated using shortcut method: finding the total no. of nodes that a current node is responsible to reach other nodes from root node.

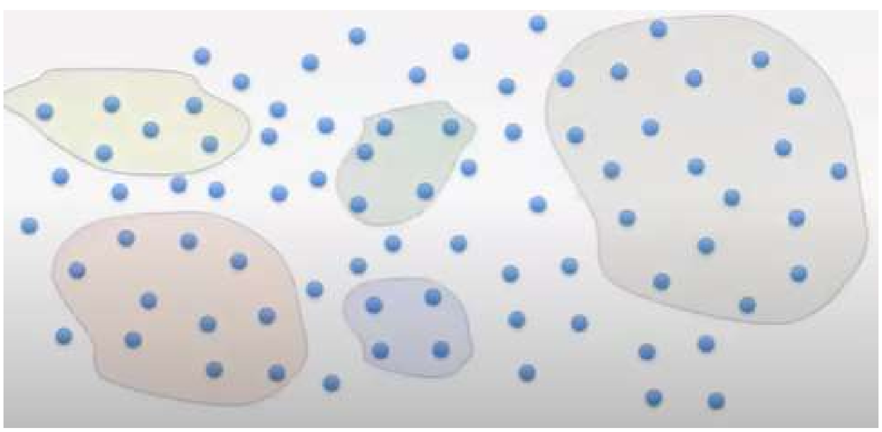
Using Betweenness to Find Communities

Bottom Up

Keep adding edges (among existing ones) starting from lowest betweenness . Graduallly join small components to build large connected components.







Top-down approach:

Start from all existing edges. The graph may look like one bid component.
Keep removing edges starting from the highest betweenness
Gradually split large components to arrive at communities
Repeat process until desired no. of clusters formed