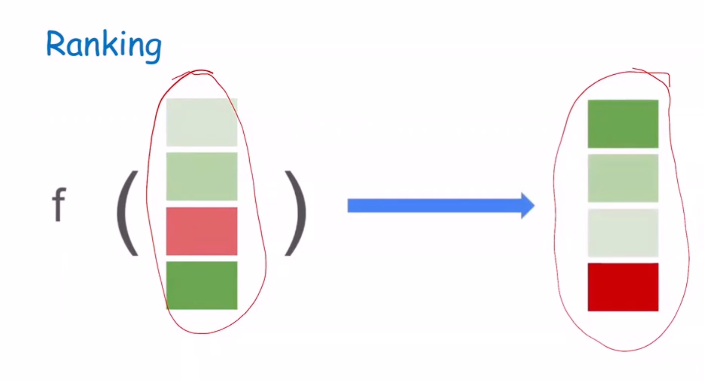
**Machine Learning : Ranking**

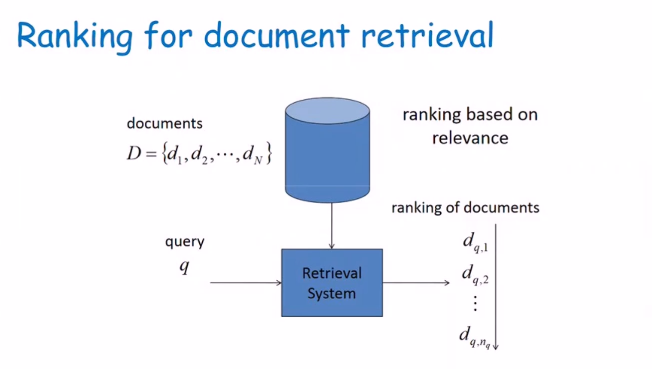
**Supervised learning**

* **Classification** : learning a mapping from features that describe data points onto discrete or categorical target values**.**
* **Regression** : learning a mapping from those features onto numeric target values.

**Ranking**

* Training models in a ranking task
* Input set of items output ordering of the set of items.
* Uses:
  + Used by a lot of companies, information retrieval
  + Natural language processing
  + Answering questions, rank order possible answers
  + Recommendations

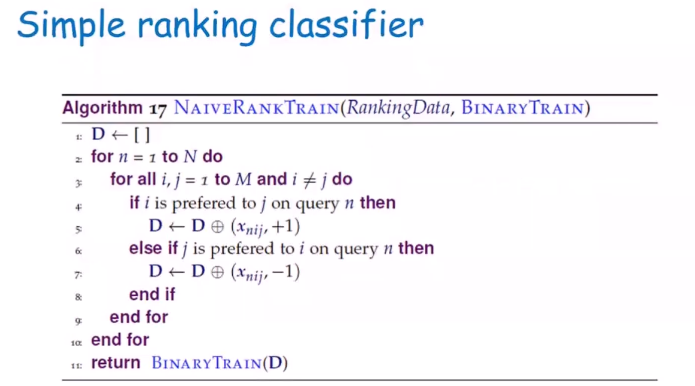




**Simple Ranking Classifier**

* Given
  + Two items
  + Query q
* Predict
  + Preference
  + With respect to q
* How do we do this?

1. Train classifier to predict pairwise preferences.
2. Turn predicted preferences into ranking.

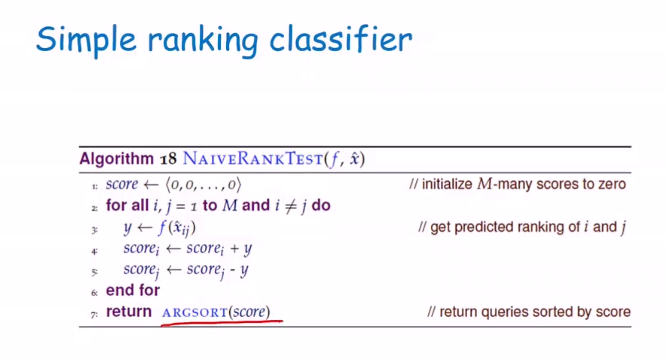


i < j

i > j

**D = Training Data (object with labels)**

**= features associated with items I, j and query n**

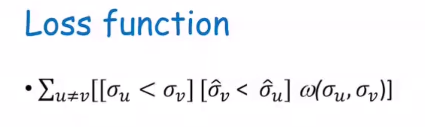


**Bipartite**

* **One item is good, one item is bad; One item is relevant, one item is not. Binary view.**

**Formalization for more general ranking**

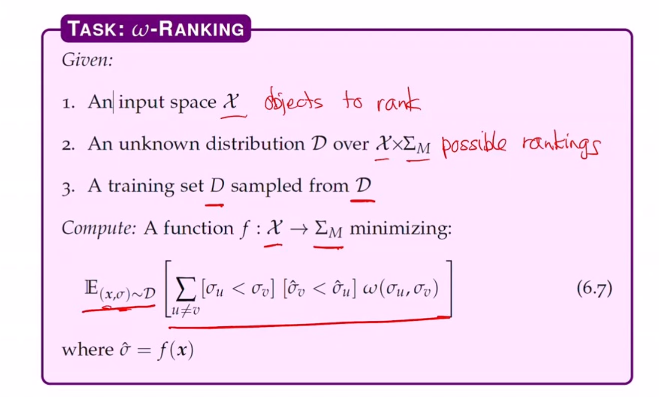
* Ranking is function.
  + Now we are looking at the overall ranking not just pairwise preferences



Actual preferred ranking

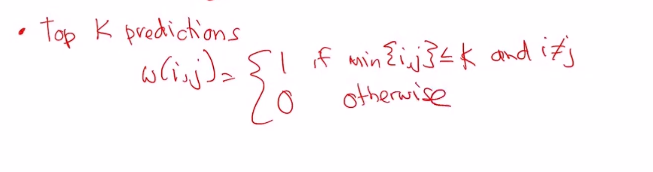
Predicted preferred ranking

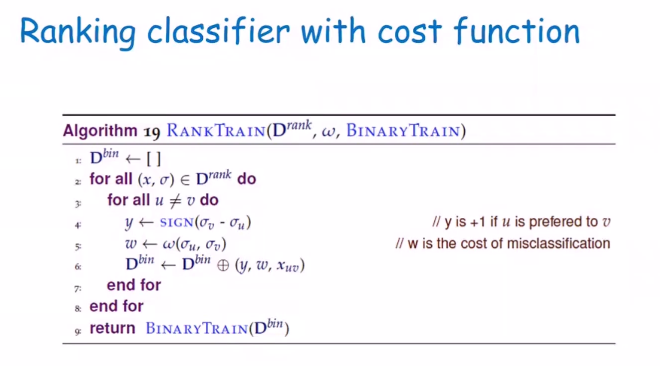
1. Symmetric
2. Monotonic
3. Triangle Inequality

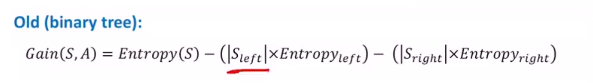
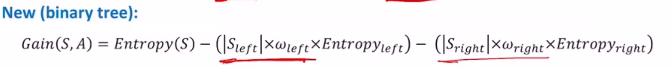
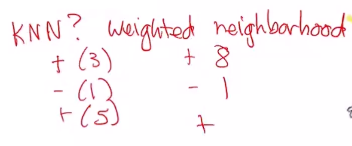


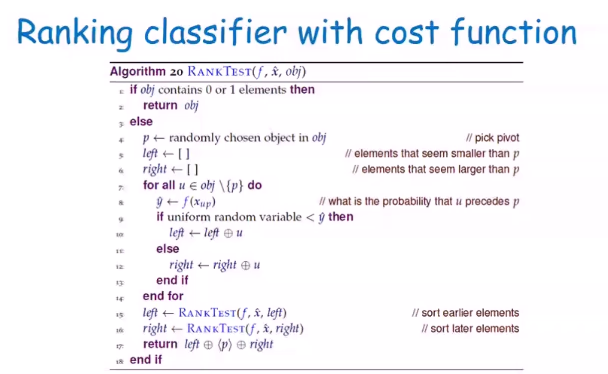
**Cost Function**

* Kemeny distance measure



**How can we use instance weights?**

* Simple : Pretend to duplicate training example proportionate to weight.
* 
* 
* KNN : weighted neighborhood
* 
* We can see that with 3 + and 5 + the majority is positive



* Rank test is like quicksort
* Function f learned for pairwise preferences for binary classifier
* X objects to rank
* Obj one particular object in the set that you are considering

