# Structured comparing

### Comparing two maps

d1=(ten1, diachi1)

d2=(ten2, diachi2)

|  |
| --- |
| **public** **static** **void** main(String[] args)  {  Map<String, Object> m1 = *buildMap*("ten", "Hoang Anh", "dc", "120 Ha noi 2");  System.***out***.println(m1);    Map<String, Object> m2 = *buildMap*("ten", "Duc Anh", "dc", "450 Ha nam ninh");  System.***out***.println(m2);    JaccardEngineMap e = **new** JaccardEngineMap();  System.***out***.println( e.jaccardIndex(m1, m2) );  } |
| {ten=Hoang Anh, dc=120 Ha noi 2}  {ten=Duc Anh, dc=450 Ha nam ninh}  dc -- 0.14285714285714285  ten -- 0.3333333333333333  0.23809523809523808 |

### Comparing two generic objects

+ Using java reflection

+ Using annotation to override the field function

# Derived distance

### Derived distance / similarity

d(x, y) give values from [0, +inf)   
>> how to make D(x, y) from d(x, y) with values in [0, 1)

We can define D(x, y) = 1 - exp( -d(x, y) )   
>> if d(x, y) = 0, D(x, y) = 0  
>> if d(x, y) = +inf, D(x, y) = 1

D(x, y) <= D(x, z) + D(y, z)  
1 - exp( -d(x, y) ) <= 1 - exp( -d(x, z) ) + 1 - exp( -d(y, z) ) ???

We can defin D(x, y) = d(x, y) / (1 + d(x, y) )  
>> d(x, y) = 0 then D(x, y) = 0  
>> d(x, y) = +inf then D(x, y) = 1

d(x, y) / (1 + d(x, y) ) <= d(x, z) / (1 + d(x, z) ) + d(z, y) / (1 + d(z, y) ) ???

### Why [0, 1) instead of [0, +inf)?

D(x, y) = d1(x, y) + d2(x, y)

+ we need to normalize d1 and d2 within [0, 1) so that they have the same **order of importance**.

### what is a distance function d(x, y)?

https://en.wikipedia.org/wiki/Metric\_(mathematics)

d(x, y) >= 0 all x, all y

d(x, y) = d(y, x) all x, all y

d(x, y) <= d(x, z) + d(y, z) all x, all y, all z

# Similarity-based Machine Learning

### Machine learning is ???

+ we give examples >> computer learn model from examples >> computer complete tasks automatically

+ compter I1 -> O1 = T(I1)   
+ compter I1 -> O2 = T(I1, E)

### SBML is ???

+ if we have a similarity function, we can have the learning system

+ classification -> kNN classification with similarity function

+ clustering -> kernel kmeans with similarity function

|  |  |  |
| --- | --- | --- |
| movie similarity function | movie classification  movie clustering  movie recommendation (imdb, youtube)  matching two parties |  |
| song similarity function | song classification  song clustering  song recommendation (youtube, amazon)  matching two parties |  |

What can we do if we can define a similarity function?

# Crawling

### Catalog page

+ the list of similar items

### Content page

+ parsing algorithm