Machine Learning

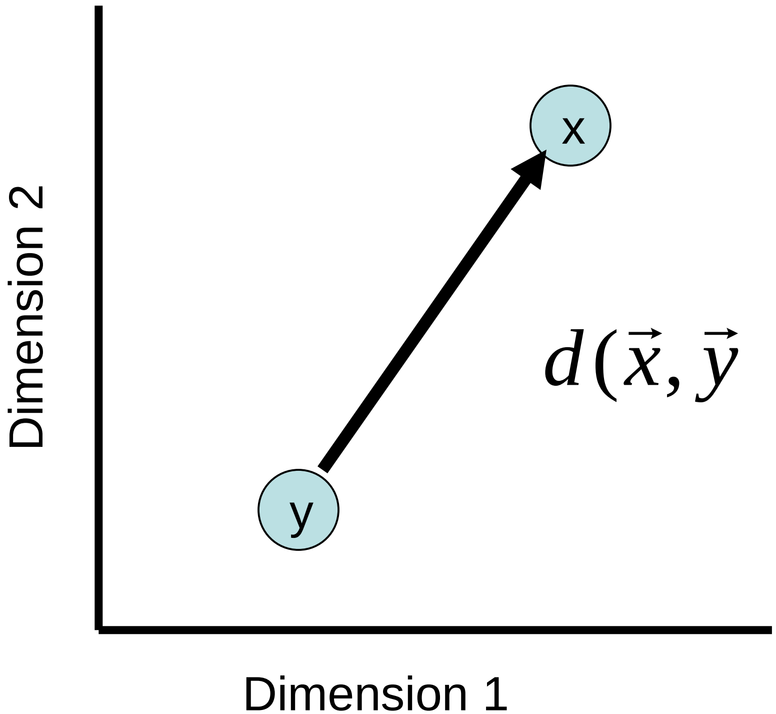
Measuring Distance

Why measure distance?

- Nearest neighbor requires a distance measure
- Also:
  - Local search methods require a measure of “locality” (Friday)
  - Clustering requires a distance measure (later)
  - Search engines require a measure of similarity, etc.

Euclidean Distance

- What people intuitively think of as “distance”



$$d(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Generalized Euclidean Distance

n = the number of dimensions

$$d(\vec{x},\vec{y}) = \left[\sum_{i=1}^{n} | x_i - y_i |^2 \right]^{1/2}$$

$$\text{where } \vec{x} =< x_1, x_2, ..., x_n >,$$

$$\vec{y} =< y_1, y_2, ..., y_n >$$

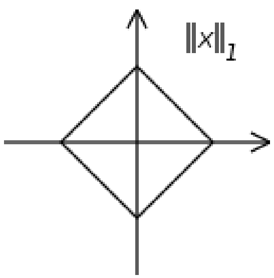
$$\text{and } \quad \forall i (x_i, y_i \in \Re)$$

L^p norms

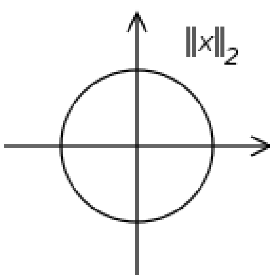
- L^p norms are all special cases of this:

$$d(\vec{x},\vec{y}) = \left[\sum_{i=1}^{n} | x_i - y_i |^p \right]^{1/p}$$

p changes the norm



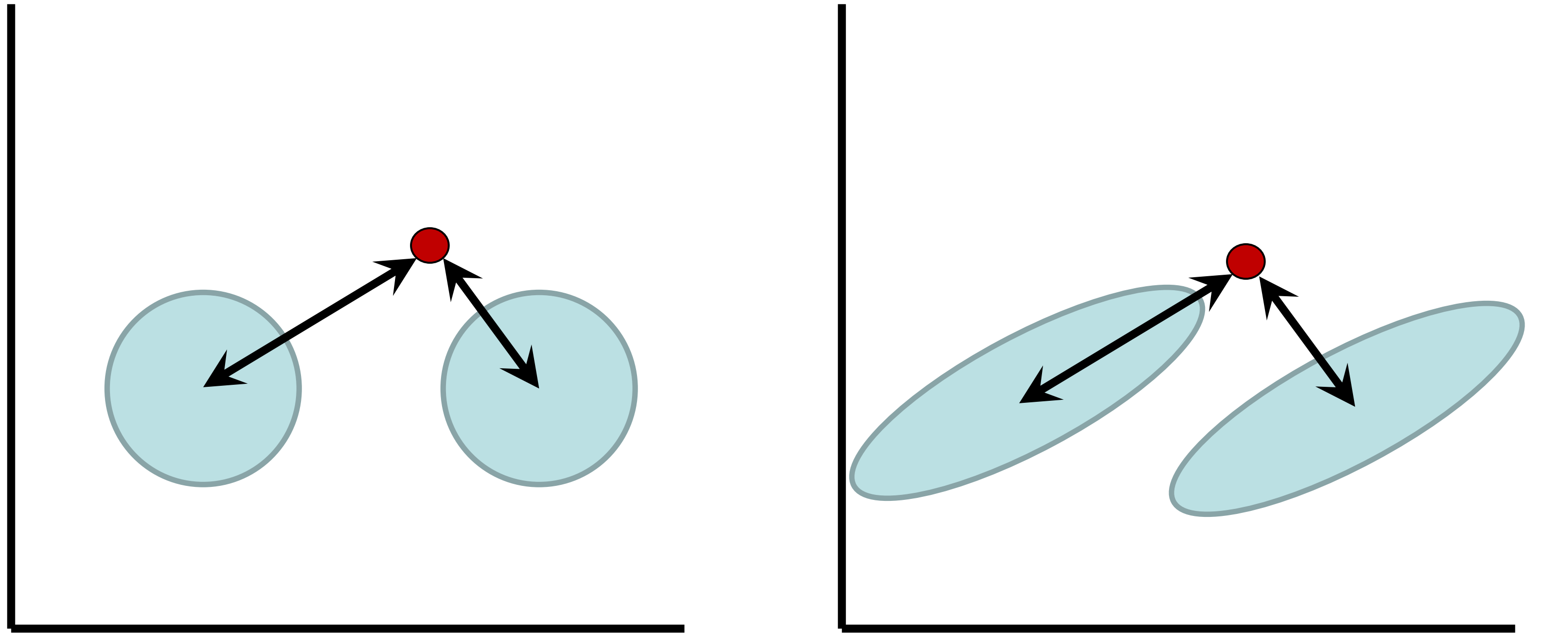
$$\|\mathrm{x}\|_1 = \mathrm{L}^1 \text{ norm} = \text{Manhattan Distance}: p = 1$$



$$\|\mathrm{x}\|_2 = \mathrm{L}^2 \text{ norm} = \text{Euclidean Distance}: p = 2$$

$$\text{Hamming Distance}: p = 1 \text{ and } x_i, y_i \in \{0,1\}$$

Weighting Dimensions



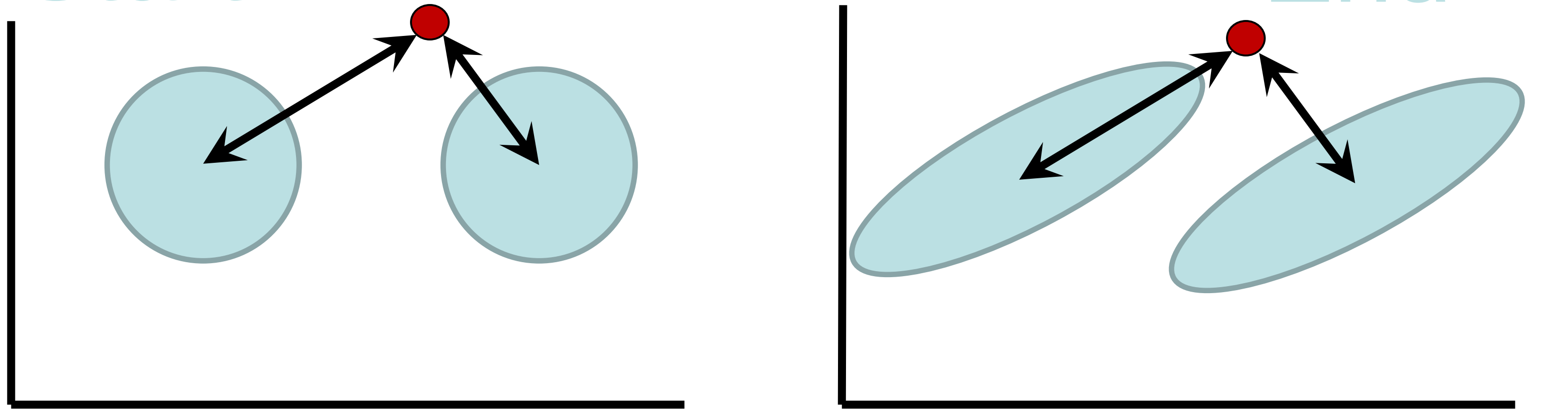
- Put point in cluster with the closest center of gravity?
- Which cluster <u>should</u> the red point go in?
- How do I measure distance in a way that gives the “right” answer for both situations?

Weighting Dimensions

Think

Start

End



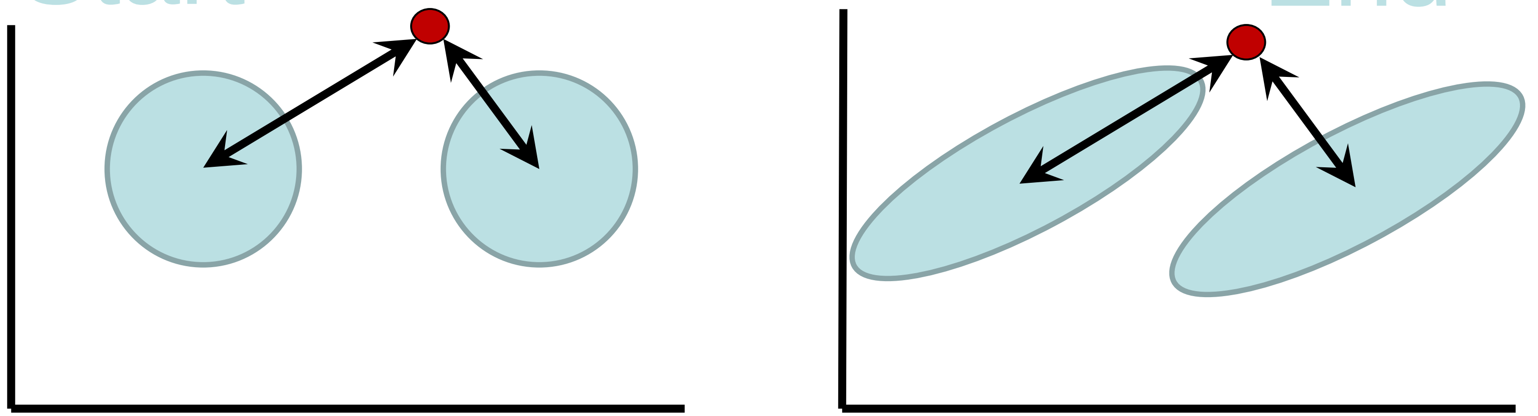
- Put point in cluster with the closest center of gravity?
- Which cluster <u>should</u> the red point go in?
- How do I measure distance in a way that gives the “right” answer for both situations?

Weighting Dimensions

Pair

Start

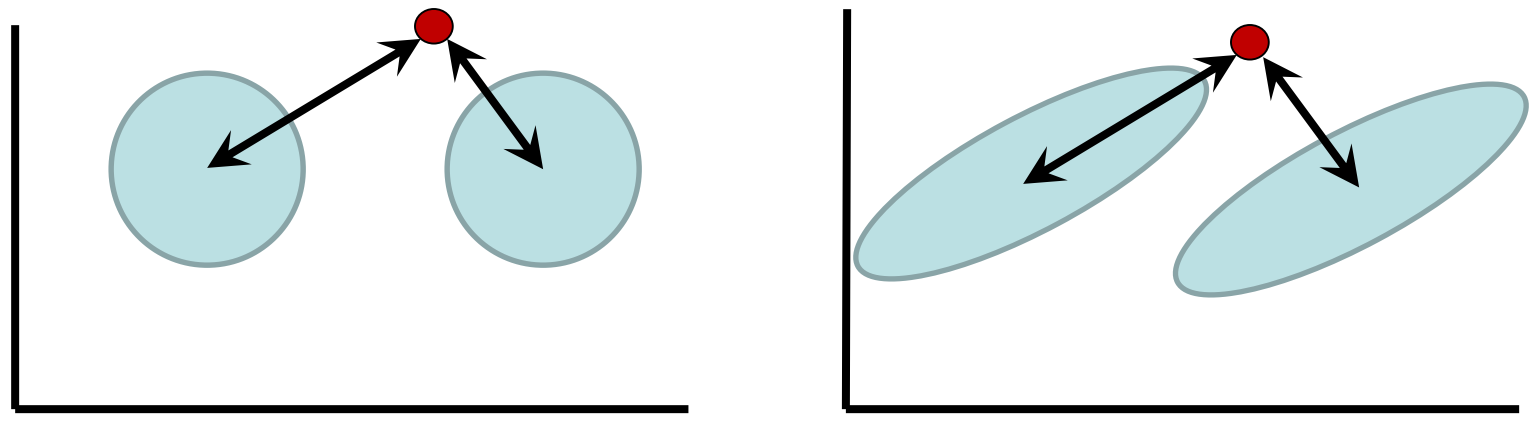
End



- Put point in cluster with the closest center of gravity?
- Which cluster should the red point go in?
- How do I measure distance in a way that gives the "right" answer for both situations?

Weighting Dimensions

Share



- Put point in cluster with the closest center of gravity?
- Which cluster <u>should</u> the red point go in?
- How do I measure distance in a way that gives the “right” answer for both situations?

Weighted Norms

- You can compensate by weighting your dimensions….

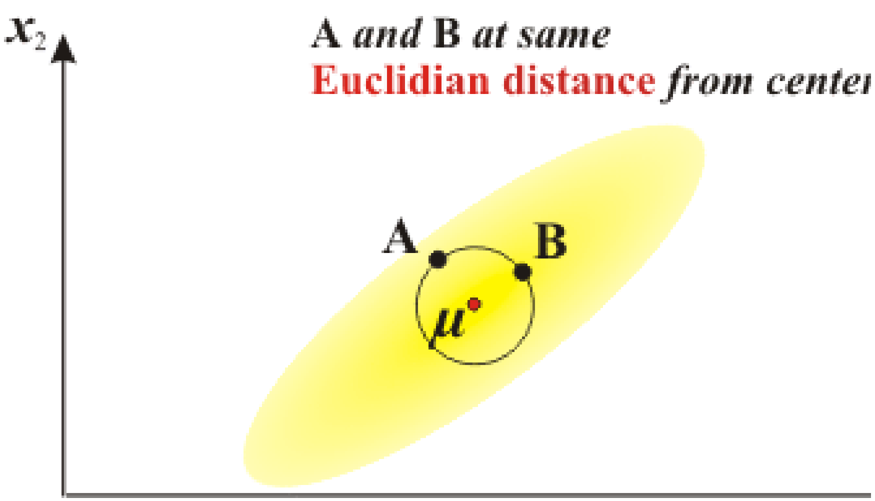
$$d(\vec{x}, \vec{y}) = \left[\sum_{i=1}^{n} w_i \, | \, x_i - y_i \, |^p \right]^{1/p}$$

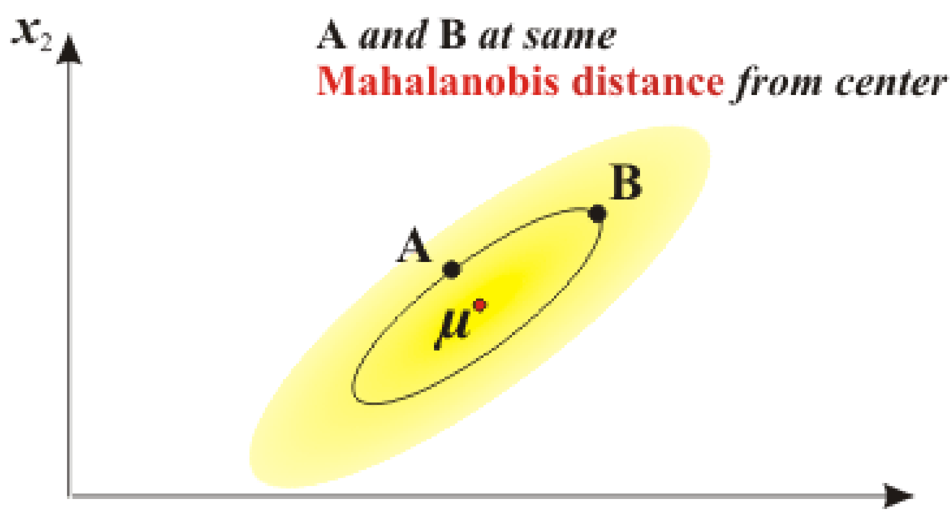
This lets you turn your circle of equal-distance into an elipse with axes parallel to the dimensions of the vectors.

Mahalanobis distance

The region of constant Mahalanobis distance around the mean of a distribution forms an ellipsoid.

The axes of this ellipsiod don't have to be parallel to the dimensions describing the vector





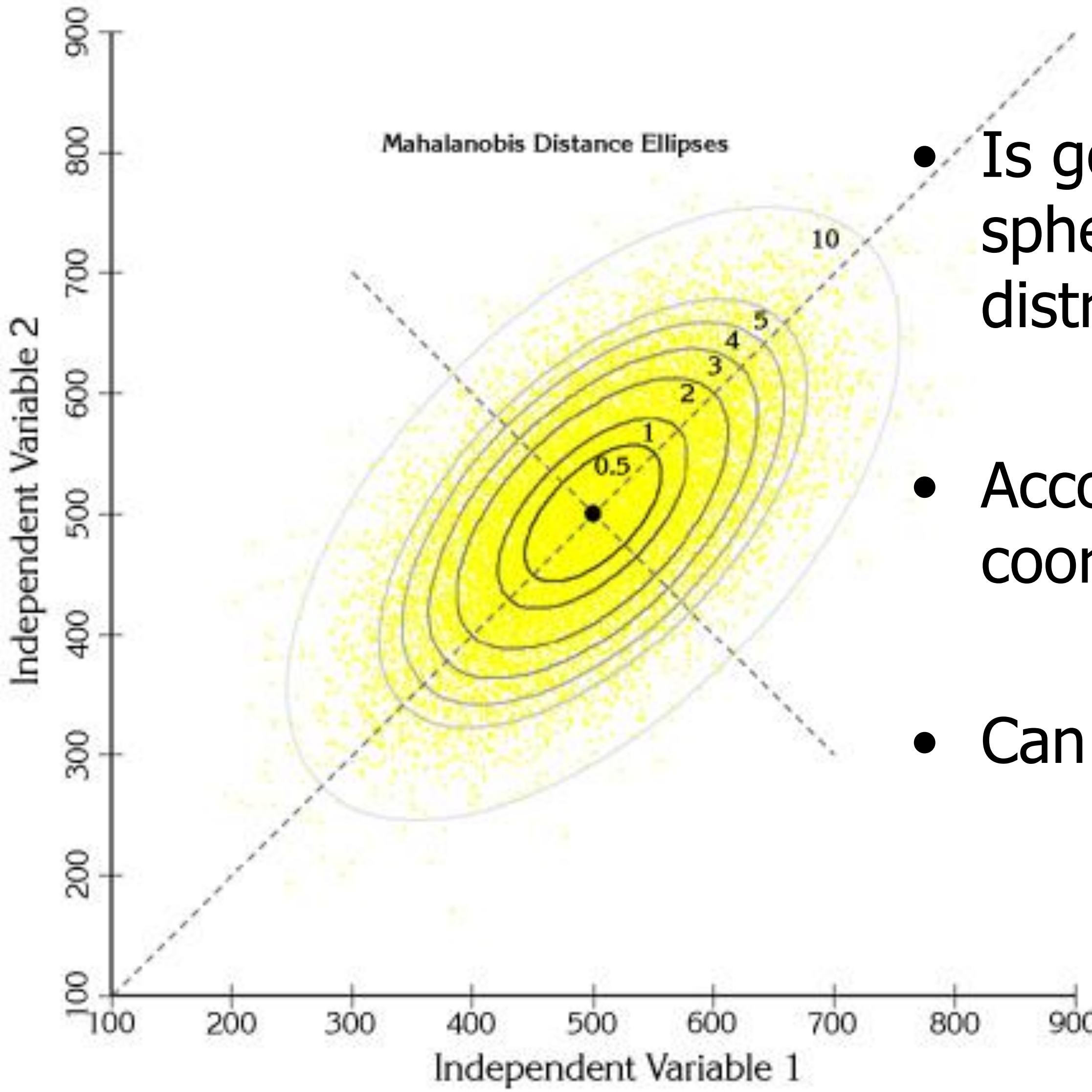
Images from: http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_mahalanobis.htm

Calculating Mahalanobis

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$$

- This matrix *S* is called the "covariance" matrix and is calculated from the data distribution

Take-away on Mahalanobis



- Is good for non-spherically symmetric distributions.
- Accounts for scaling of coordinate axes
- Can reduce to Euclidean

What is a "metric"?

- A metric has these four qualities.

$$d(x,y) = 0 \quad \text{iff} \quad x = y \qquad \text{(reflexivity)}$$

$$d(x,y) \geq 0 \qquad \text{(non-negative)}$$

$$d(x,y) = d(y,x) \qquad \text{(symmetry)}$$

$$d(x,y) + d(y,z) \geq d(x,z) \qquad \text{(triangle inequality)}$$

- …otherwise, call it a "measure"

Metric, or not?

- Driving distance with 1-way streets

- Categorical Stuff :
  - Is distance (Jazz to Blues to Rock) no less than distance (Jazz to Rock)?

Categorical Variables

- Consider feature vectors for genre & vocals:

  - Genre: {Blues, Jazz, Rock, Hip Hop}
  - Vocals: {vocals,no vocals}

s1 = {rock, vocals}

s2 = {jazz, no vocals}

s3 = { rock, no vocals}

- Which two songs are more similar?

One Solution:Hamming distance

Blues	Jazz	Rock	Hip Hop	Vocals
0	0	1	0	1
0	1	0	0	0
0	0	1	0	0

s1 = {rock, vocals}

s2 = {jazz, no_vocals}

s3 = { rock, no_vocals}

Hamming Distance = number of different bits in two binary vectors

Hamming Distance

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^{n} | x_i - y_i |$$

where $\vec{x} =< x_1, x_2, \ldots, x_n >,$

$$\vec{y} =< y_1, y_2, \ldots, y_n >$$

and $\forall i(x_i, y_i \in \{0,1\})$

Defining your own distance (an example)

How often does artist *x* quote artist *y*?

Quote Frequency

	Beethoven	Beatles	Liz Phair
Beethoven	7	0	0
Beatles	4	5	0
Liz Phair	?	1	2

Let’s build a distance measure!

Defining your own distance (an example)

	Beethoven	Beatles	Liz Phair
Beethoven	7	0	0
Beatles	4	5	0
Liz Phair	?	1	2

$$\text{Quote frequency } Q_f(x, y) = \text{value in table}$$

$$\text{Distance } d(x, y) = 1 - \frac{Q_f(x, y)}{\sum_{z \in Artists} Q_f(x, z)}$$

Missing data

- What if, for some category, on some examples, there is no value given?
- Approaches:
  - Discard all examples missing the category
  - Fill in the blanks with the mean value
  - Only use a category in the distance measure if both examples give a value

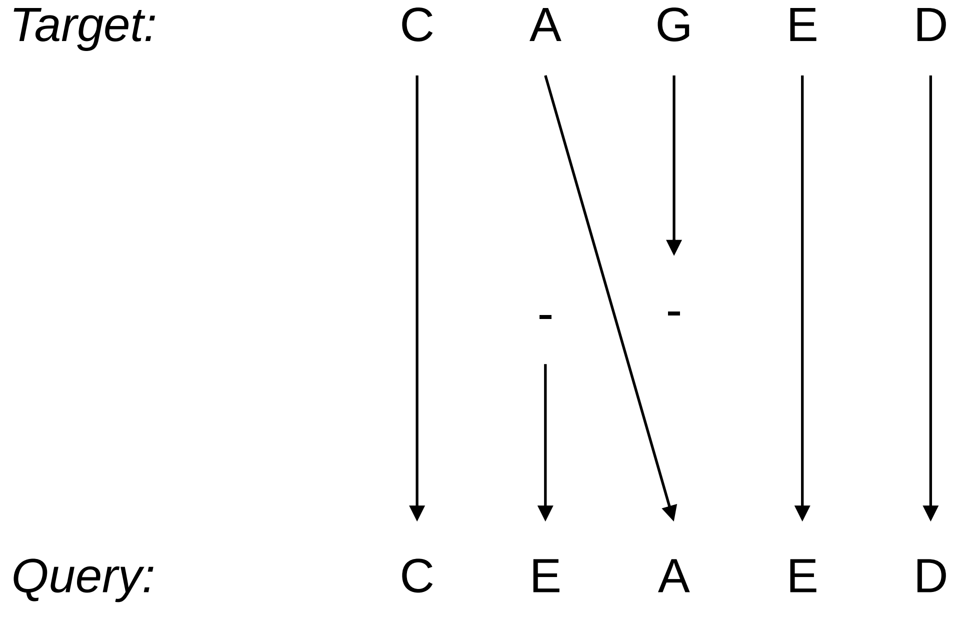
Dealing with missing data

$$w_i = \begin{cases} 1, & \text{if both } x_i \text{ and } y_i \text{ are defined} \\ 0, & \text{else} \end{cases}$$

$$d(\vec{x}, \vec{y}) = \frac{n}{n - \sum_{i=1}^{n} w_i} \left[\sum_{i=1}^{n} w_i \phi(x_i, y_i) \right]$$

Edit Distance

- Query = string from finite alphabet
- Target = string from finite alphabet
- Cost of Edits = Distance



Semantic Relatedness

d(Portland, Hippies)

<<

d(Portland, Monster trucks)

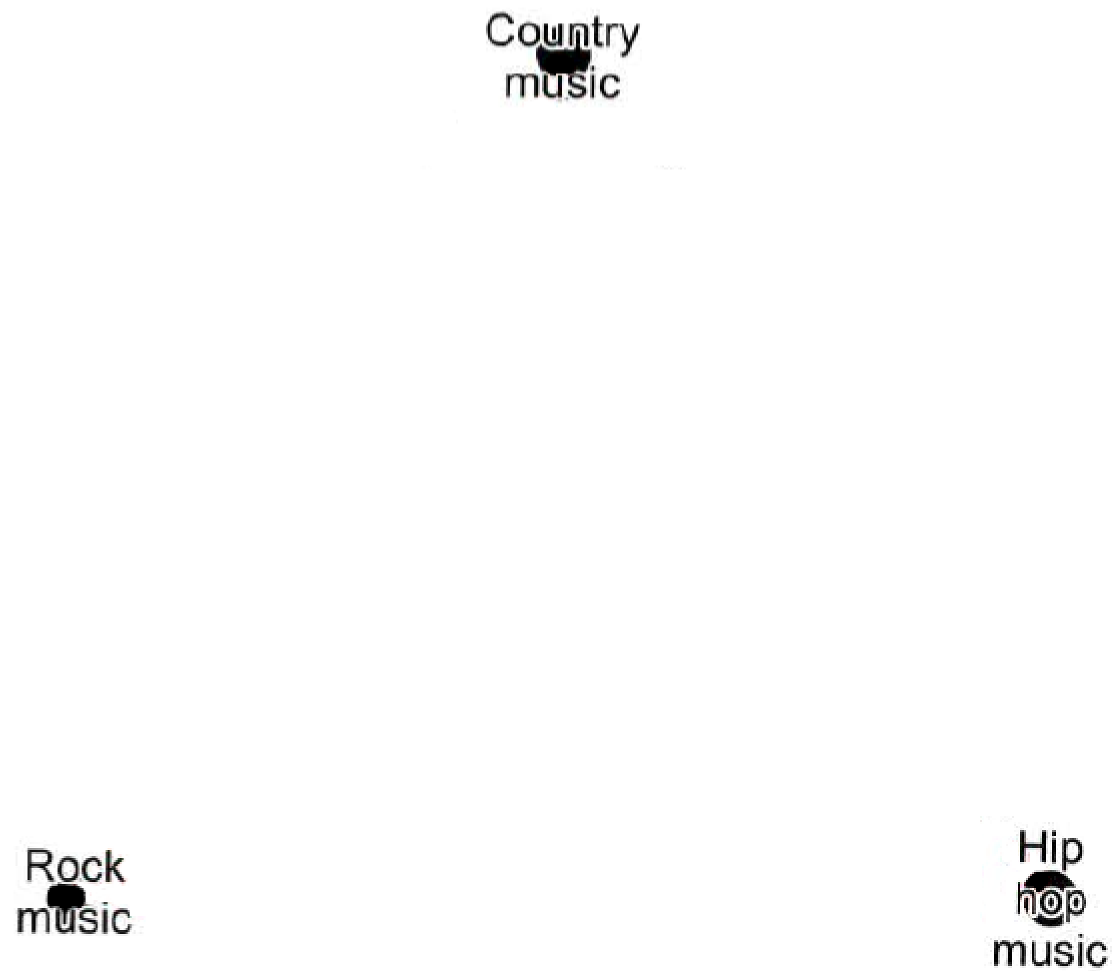
Semantic Relatedness

- Several measures have been proposed

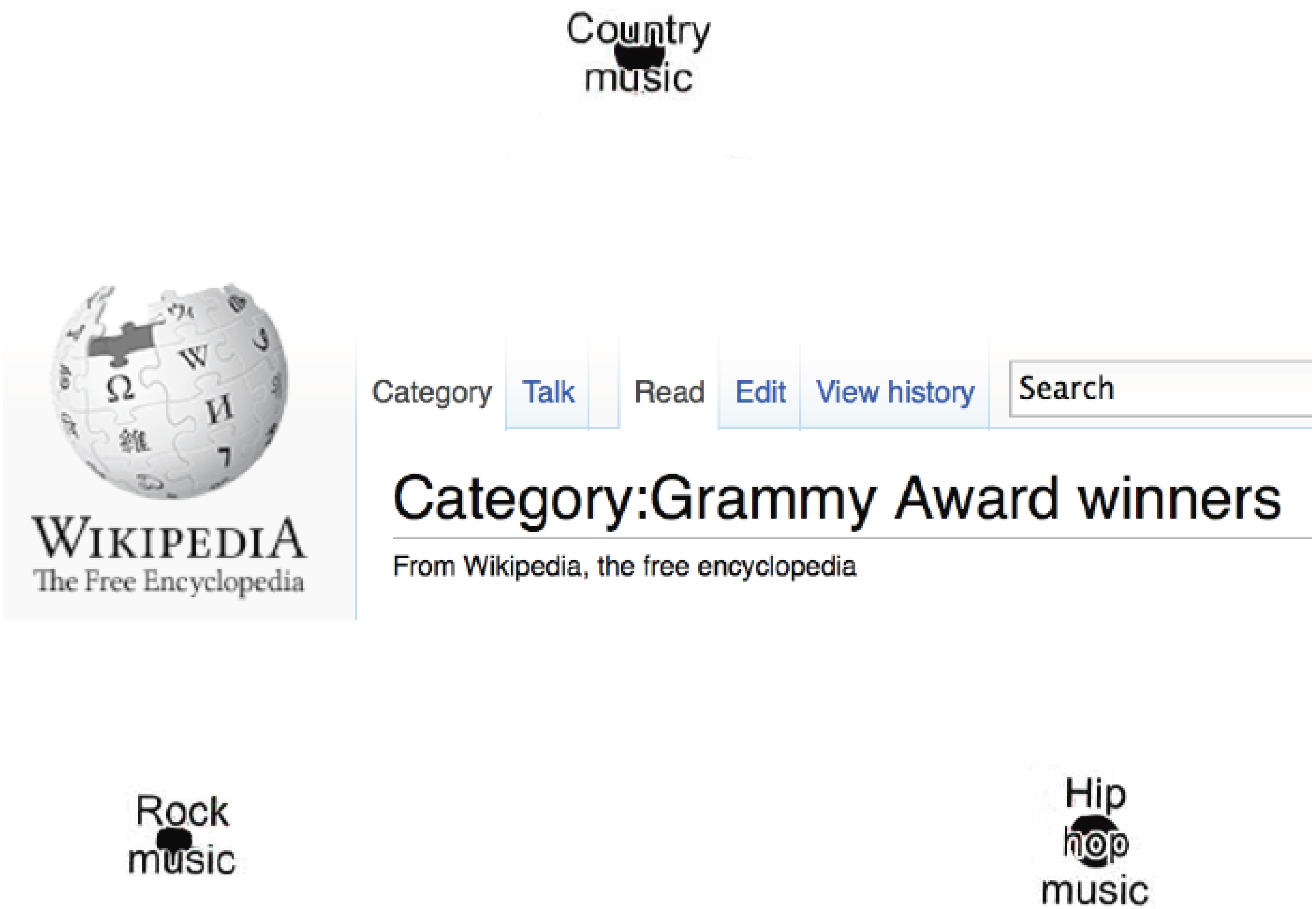
- One that works well: “Milne-Witten”

$SR_{MW}(\mathbf{x}, \mathbf{y}) \propto$ fraction of Wikipedia in-links to either **x** or **y** that link to both

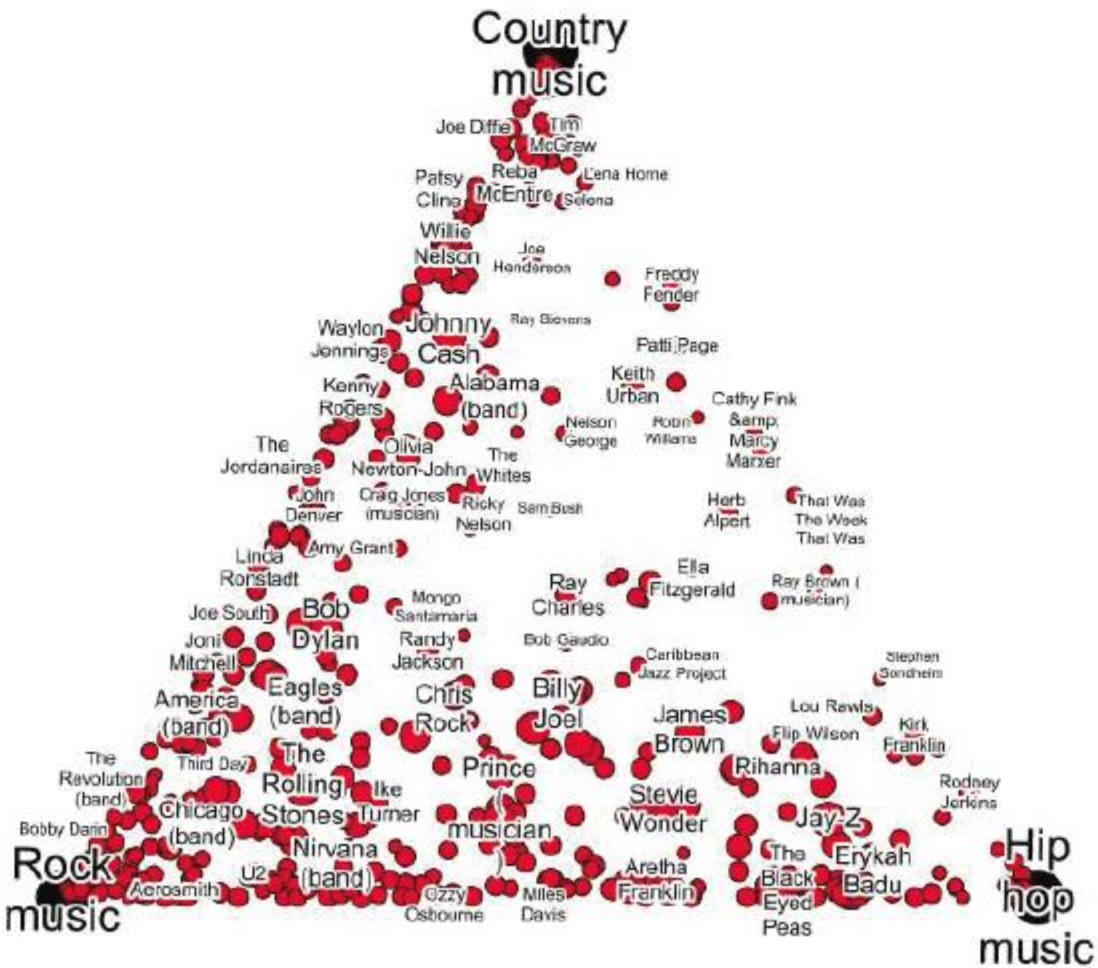
Ad-hoc Reference Systems



Ad-hoc Reference Systems

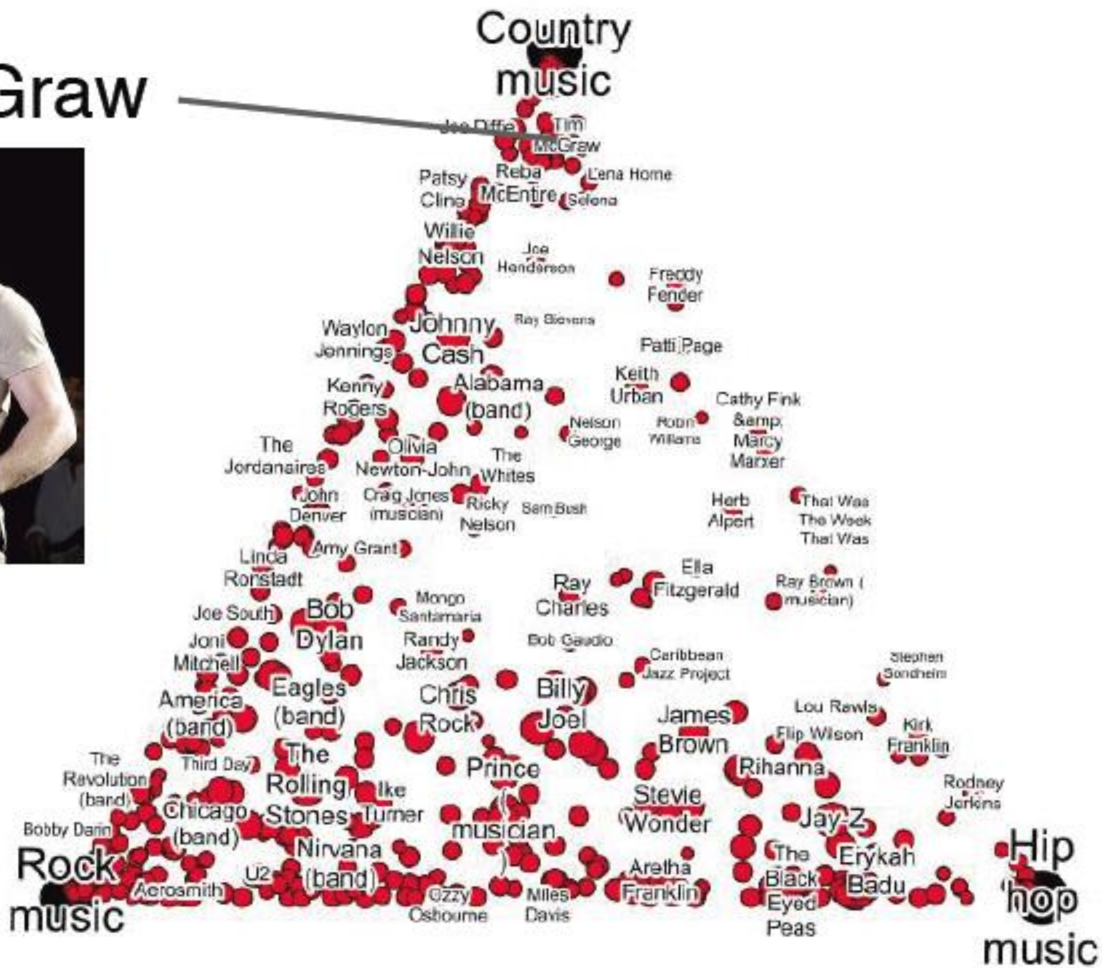


Ad-hoc Reference Systems



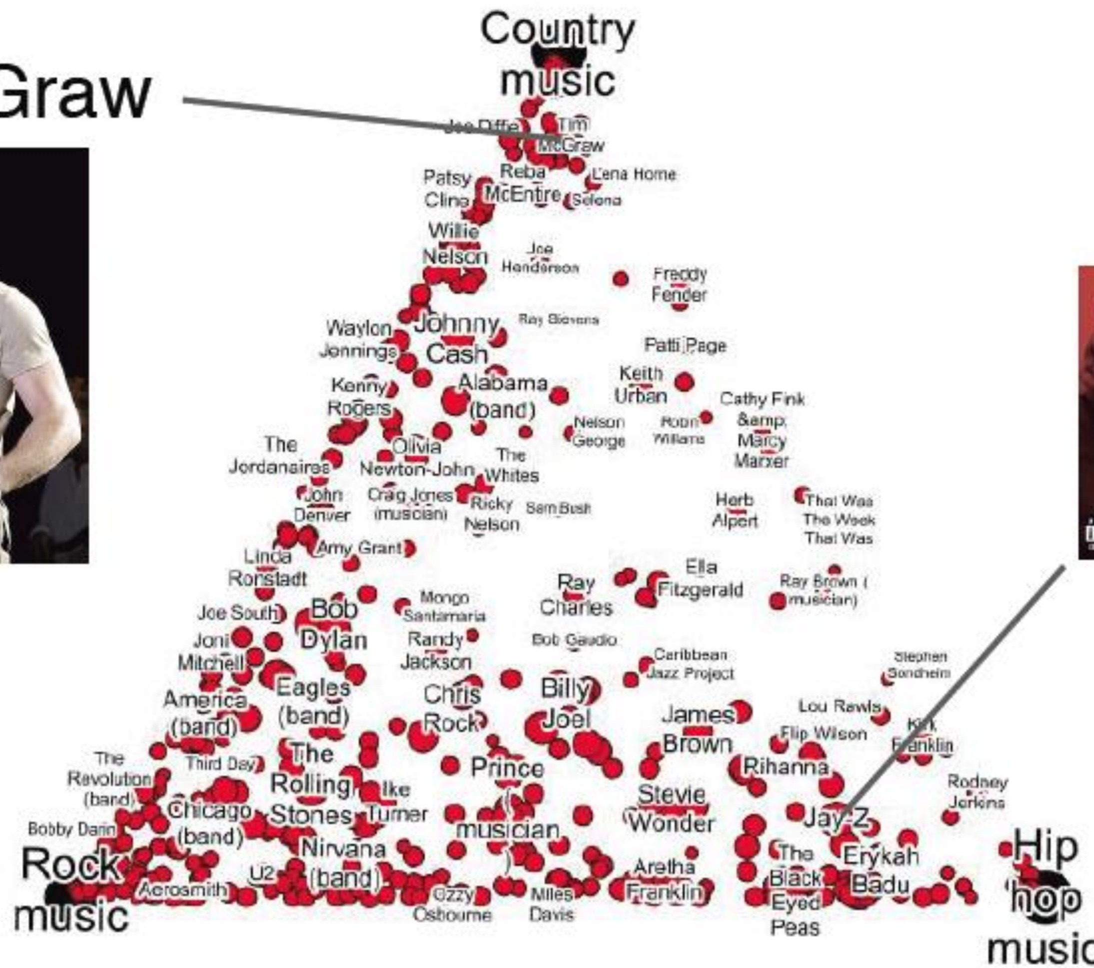
Ad-hoc Reference Systems

Tim McGraw



Ad-hoc Reference Systems

Tim McGraw



Jay-Z

One more distance measure

- Kullback–Leibler divergence
  - Related to entropy & information gain
  - not a metric, since it is not symmetric
  - Take **EECS 428:Information Theory** to find out more