Introduction to Data Mining

Distances & Similarities

CPSC/AMTH 445a/545a

Guy Wolf
guy.wolf@yale.edu

Yale University
Fall 2016

Outline

Consider a dataset X as an arbitrary collection of data points

Distance metric

A distance metric is a function $d : X \times X \to [0, \infty)$ that satisfies three conditions for any $x, y, z \in X$:

1. $d(x, y) = 0 \Leftrightarrow x = y$
2. $d(x, y) = d(y, x)$
3. $d(x, y) \leq d(x, z) + d(z, y)$

The set X of data points together with an appropriate distance metric $d(\cdot, \cdot)$ is called a metric space.

When $X \subset \mathbb{R}^n$ we can consider Euclidean distances:

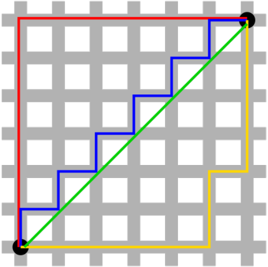
Euclidean distance

The distance between $x, y \in X$ is defined by
$\|x - y\|^2 = \sum_{i=1}^{n}(x[i] - y[i])^2$

- One of the classic most common distance metrics
- Often inappropriate in realistic settings without proper preprocessing & feature extraction
- Also used for *least mean square error* optimizations
- Proximity requires all attributes to have equally small differences

Manhattan distance

The Manhattan distance between $x, y \in X$ is defined by $\|x - y\|_1 = \sum_{i=1}^{n} |x[i] - y[i]|$. This distance is also called taxicab or cityblock distance



Taken from Wikipedia

Minkowski distance

The Minkowski distance between $x, y \in X \subset \mathbb{R}^n$ is defined by

$$\|x - y\|_p^p = \sum_{i=1}^{n} |x[i] - y[i]|^p$$

for some $p > 0$. This is also called the ℓ_p distance.

Three popular Minkowski distances are:

$p = 1$ Manhattan distance: $\|x - y\|_1 = \sum_{i=1}^{n} |x[i] - y[i]|$

$p = 2$ Euclidean distance: $\|x - y\|_2 = \sum_{i=1}^{n} |x[i] - y[i]|^2$

$p = \infty$ Supremum/$\ell_{\max}$ distance: $\|x - y\|_\infty = \sup_{1 \le i \le n} |x[i] - y[i]|$

Distance metrics

Normalization & standardization

Minkowski distances require normalization to deal with varying magnitudes, scaling, distribution or measurement units.

Min-max normalization

$\text{minmax}(x)[i] = \frac{x[i]-m_i}{r_i}$, where m_i and r_i are the min value and range of attribute i.

Z-score standardization

$\text{zscore}(x)[i] = \frac{x[i]-\mu_i}{\sigma_i}$, where μ_i and σ_i are the mean and STD of attribute i.

log attenuation

$\text{logatt}(x)[i] = \text{sgn}(x[i]) \log(|x[i]| + 1)$

Mahalanobis distances

The Mahalanobis distance is defined by

$$\mathrm{mahal}(x, y) = \sqrt{(x - y)\Sigma^{-1}(x - y)^T}$$

where Σ is the covariance matrix of the data and data points are represented as row vectors.

Mahalanobis distances

The Mahalanobis distance is defined by

$$\mathrm{mahal}(x, y) = \sqrt{(x - y)\Sigma^{-1}(x - y)^T}$$

where Σ is the covariance matrix of the data and data points are represented as row vectors.

When all attributes are independent with unit standard deviation (e.g., z-scored) then $\Sigma = \mathrm{Id}$ and we get the Euclidean distance.

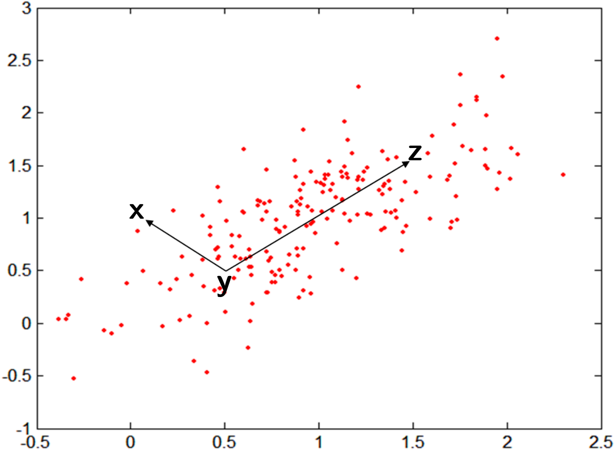
Mahalanobis distances

The Mahalanobis distance is defined by

$$\text{mahal}(x, y) = \sqrt{(x - y)\Sigma^{-1}(x - y)^T}$$

where Σ is the covariance matrix of the data and data points are represented as row vectors.

When all attributes are independent with variances σ_i^2 then $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ and we get $\text{mahal}(x, y) = \sqrt{\sum_{i=1}^{n}(\frac{x[i]-y[i]}{\sigma_i})^2}$, which is the Euclidean distance between z-scored data points.



$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

$$\begin{aligned} x &= (0,1) \\ y &= (0.5,0.5) \\ z &= (1.5,1.5) \end{aligned}$$

$$\begin{aligned} d(x,y) &= 5 \\ d(y,z) &= 4 \end{aligned}$$

Distance metrics
Hamming distance

When the data contains nominal values, we can use Hamming distances:

Hamming distances

The hamming distance is defined as $\text{hamm}(x, y) = \sum_{i=1}^{n} x[i] \neq y[i]$ for data points x, y that contain n nominal attributes.

This distance is equivalent to ℓ_1 distance with binary flag representation.

Example

If $x = (\text{'big'}, \text{'black'}, \text{'cat'})$, $y = (\text{'small'}, \text{'black'}, \text{'rat'})$, and $z = (\text{'big'}, \text{'blue'}, \text{'bulldog'})$ then $hamm(x, y) = d(x, z) = 2$ and $hamm(y, z) = 3$.

Similarities or affinities quantify whether, or how much, data points are *similar*.

Similarity/affinity measure

We will consider a similarity or affinity measure as a function $a : X \times X \to [0, 1]$ such that for every $x, y \in X$

- $a(x, x) = a(y, y) = 1$
- $a(x, y) = a(y, x)$

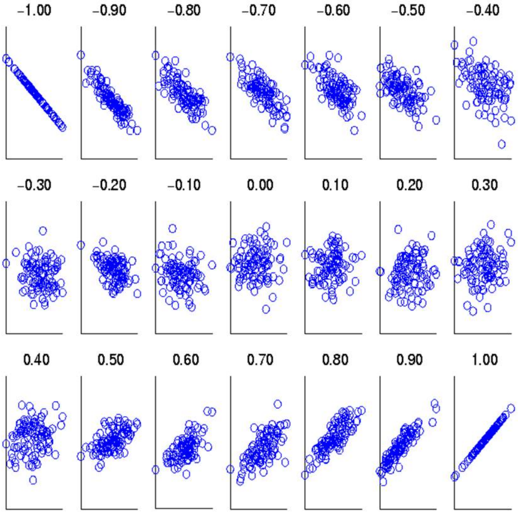
Dissimilarities quantify the opposite notion, and typically take values in $[0, \infty)$, although they are sometimes normalized to finite ranges.

Distances can serve as a way to measure dissimilarities.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Similarities and dissimilarities

Correlation



Given a distance metric $d(x, y)$, we can use it to formulate Guassian affinities

Gaussian affinities

Gaussian affinities are defined as
$k(x, y) = \exp(-\frac{d(x,y)^2}{2\varepsilon})$
given a distance metric d.

Essentially, data points are similar if they are within the same spherical neighborhoods w.r.t. the distance metric, whose radius is determined by ε.

For Euclidean distances they are also known as RBF (radial basis function) affinities.

Another similarity metric in Euclidean space is based on the inner product (i.e., dot product) $\langle x, y \rangle = \|x\| \, \|y\| \cos(\angle xy)$

Cosine similarities

The cosine similarity between $x, y \in X \subset \mathbb{R}^n$ is defined as

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \, \|y\|}$$

Similarities and dissimilarities

Cosine similarities

Another similarity metric in Euclidean space is based on the inner product (i.e., dot product) $\langle x, y \rangle = \|x\| \, \|y\| \cos(\angle xy)$

Cosine similarities

The cosine similarity between $x, y \in X \subset \mathbb{R}^n$ is defined as

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \, \|y\|}$$

Similarities and dissimilarities

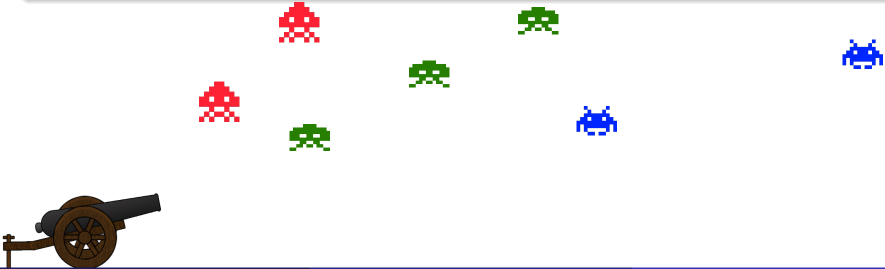
Cosine similarities

Another similarity metric in Euclidean space is based on the inner product (i.e., dot product) $\langle x, y \rangle = \|x\| \, \|y\| \cos(\angle xy)$

Cosine similarities

The cosine similarity between $x, y \in X \subset \mathbb{R}^n$ is defined as

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \, \|y\|}$$



Similarities and dissimilarities

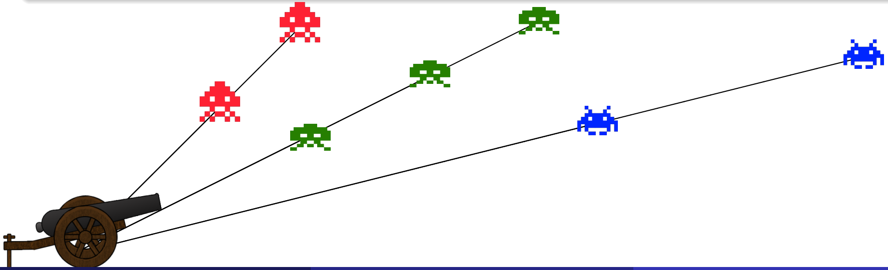
Cosine similarities

Another similarity metric in Euclidean space is based on the inner product (i.e., dot product) $\langle x, y \rangle = \|x\| \, \|y\| \cos(\angle xy)$

Cosine similarities

The cosine similarity between $x, y \in X \subset \mathbb{R}^n$ is defined as

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \, \|y\|}$$



Similarities and dissimilarities

Jaccard index

For data with n binary attributes we consider two similarity metrics:

Simple matching coefficient

$$SMC(x,y) = \frac{\sum_{i=1}^{n} x[i] \wedge y[i] + \sum_{i=1}^{n} \neg x[i] \wedge \neg y[i]}{n}$$

Jaccard coefficient

$$J(x,y) = \frac{\sum_{i=1}^{n} x[i] \wedge y[i]}{\sum_{i=1}^{n} x[i] \vee y[i]}$$

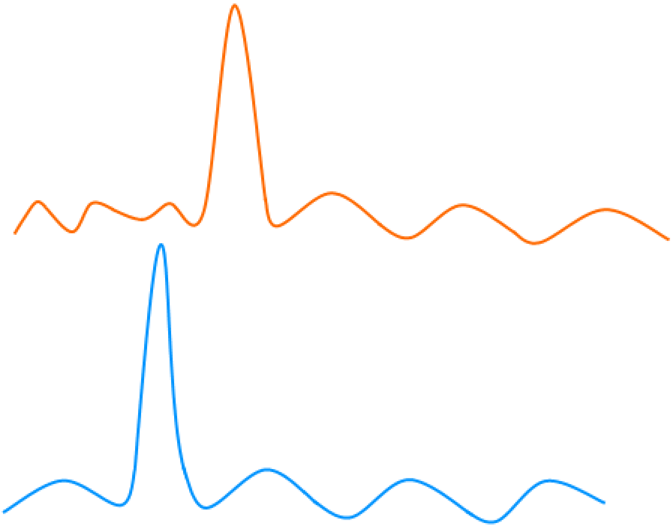
The Jaccard coefficient can be extended to continuous attributes:

Tanimoto (extended Jaccard) coefficient

$$T(x,y) = \frac{\langle x,y \rangle}{\|x\|^2 + \|y\|^2 - \langle x,y \rangle}$$

Dynamic time-warp

Comparing misaligned signals



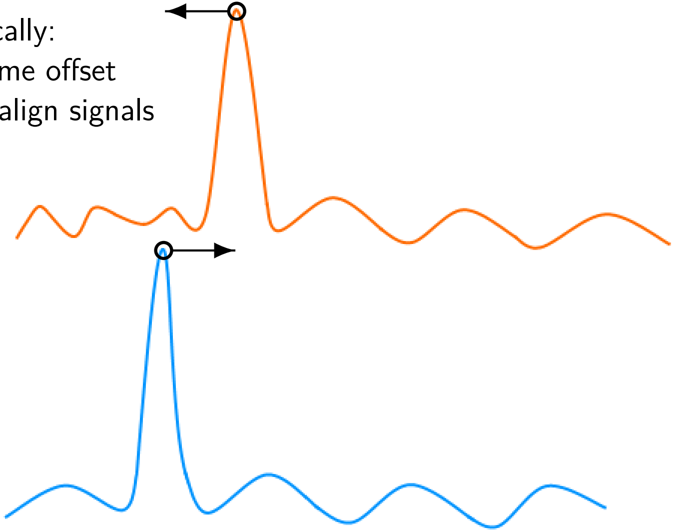
Dynamic time-warp

Comparing misaligned signals

Theoretically:

Use time offset

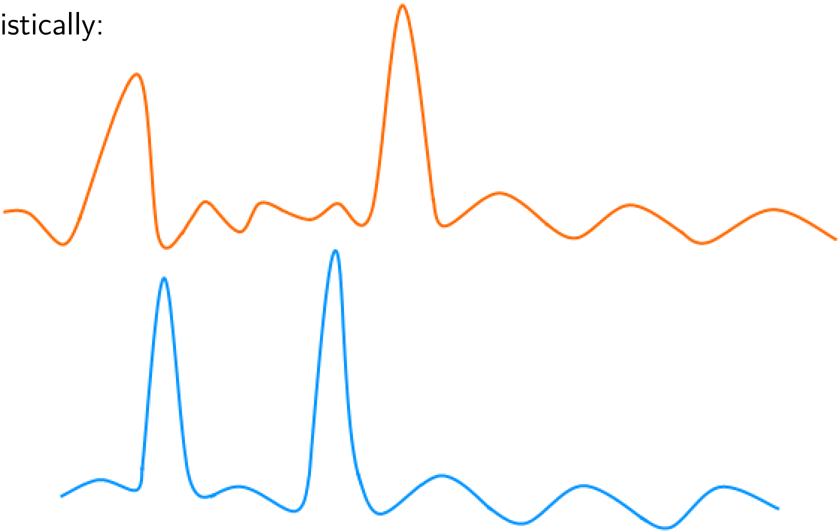
to align signals



Dynamic time-warp

Comparing misaligned signals

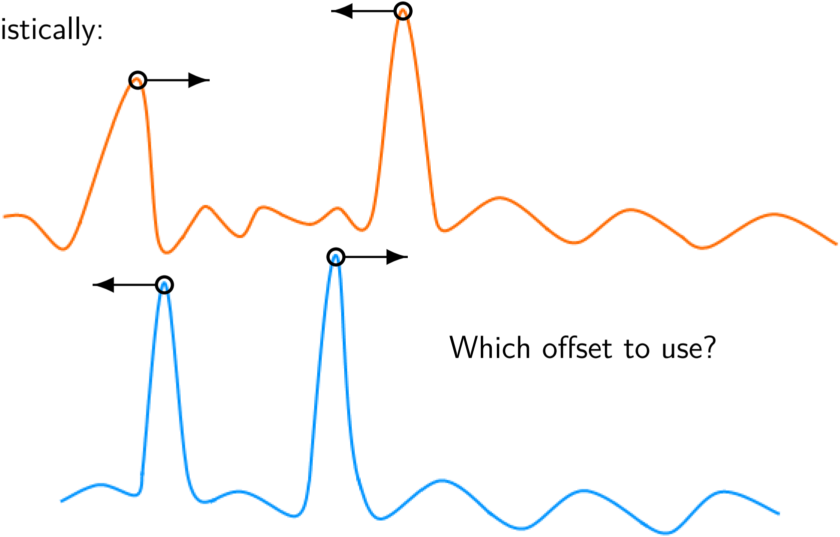
Realistically:



Dynamic time-warp

Comparing misaligned signals

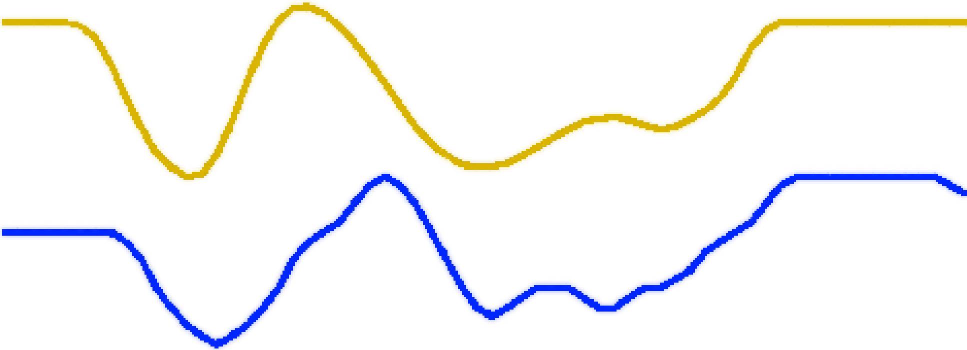
Realistically:



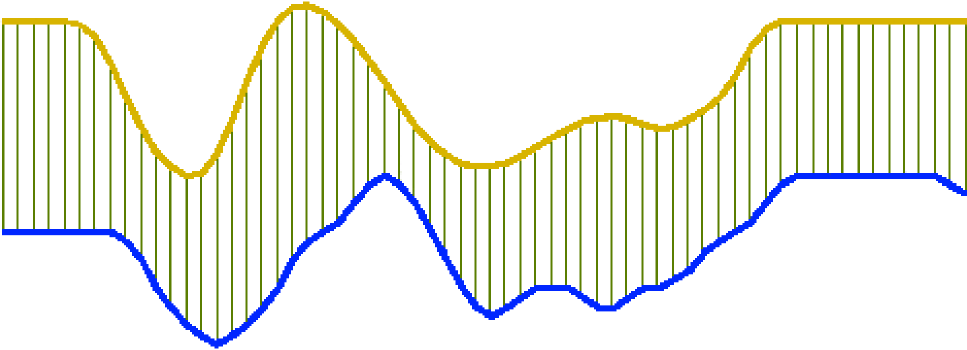
Which offset to use?

Dynamic time-warp

Adaptive alignment

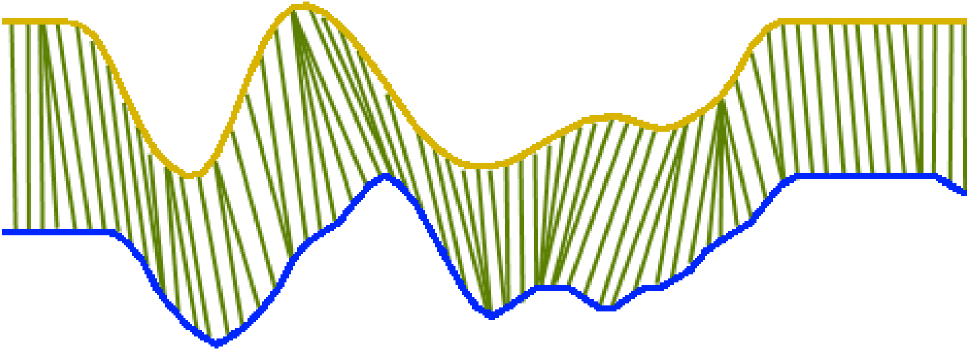


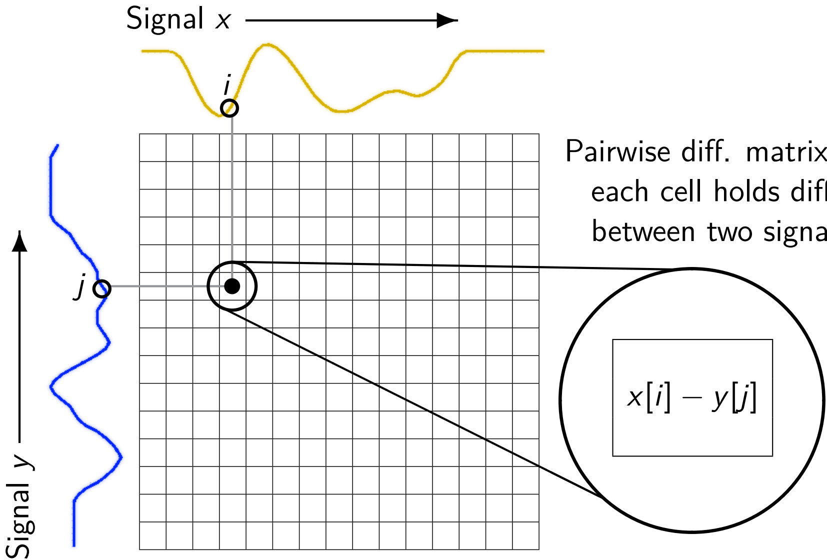
Dynamic time-warp
Adaptive alignment



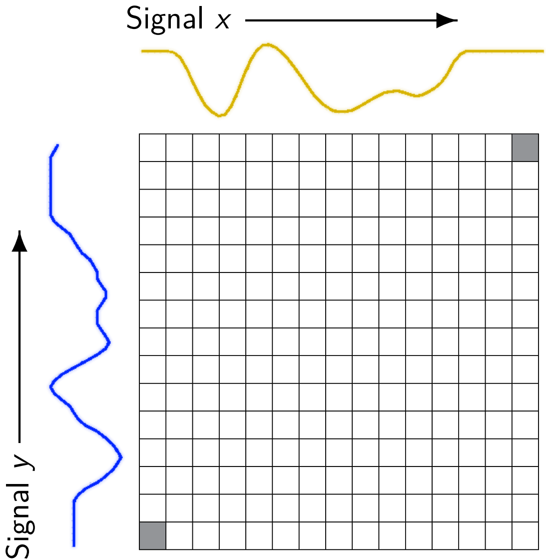
Dynamic time-warp

Adaptive alignment

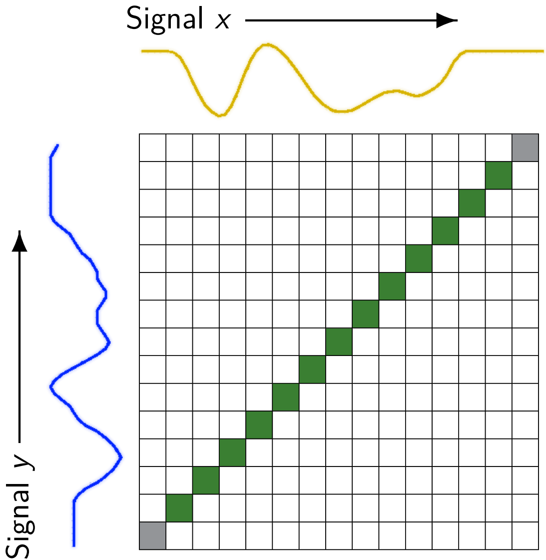




Pairwise diff. matrix:
each cell holds difference between two signal entries



Alignment path:
get from start to end
of both signals



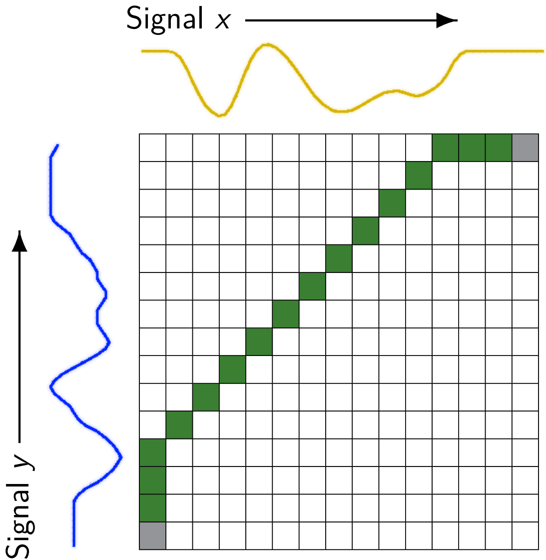
1:1 alignment:
trivial - nothing modified by the alignment

Aligned distance:

$$\sum \left(\blacksquare\right)^2 = \|x - y\|^2$$

Dynamic time-warp

Computing DTW dissimilarity



Time offset:

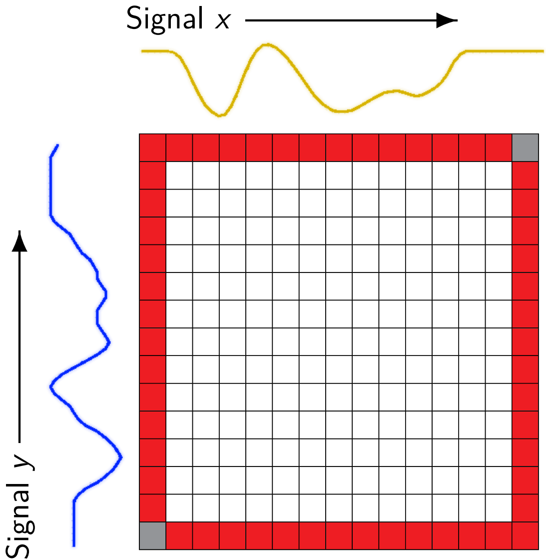
works sometimes, but not always optimal

Aligned distance:

$$\sum \left(\blacksquare \right)^2 = ?$$

Dynamic time-warp

Computing DTW dissimilarity



Extreme offset:

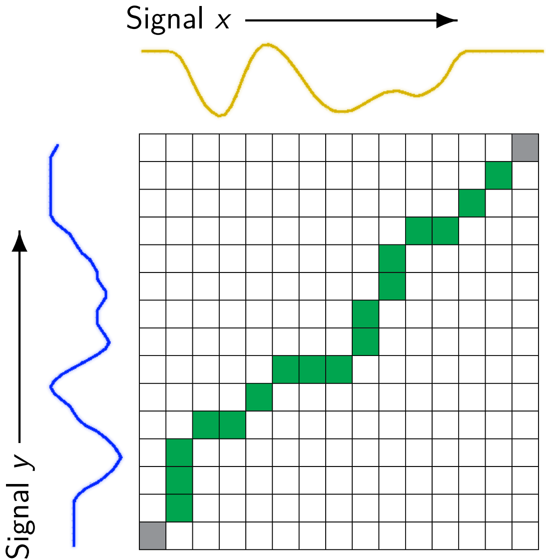
complete misalignment - worst alignment alternative

Aligned distance:

$$\sum \left(\blacksquare\right)^2 = \|x\|^2 + \|y\|^2$$

Dynamic time-warp

Computing DTW dissimilarity



Optimal alignment:

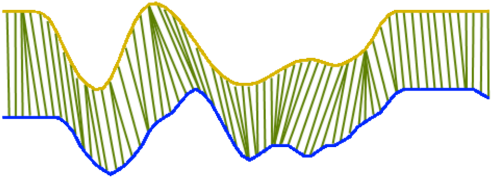
Optimize alignment by minimizing aligned distance

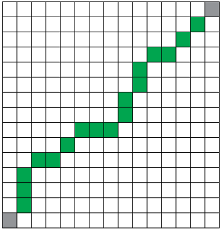
Aligned distance:

$$\sum \left(\blacksquare \right)^2 = \min$$

Dynamic time-warp

Dynamic programming algorithm





Dynamic Programming

- A method for solving complex problems by breaking them down into simpler subproblems.
- Applicable to problems exhibiting the properties of overlapping subproblems and optimal substructure.
- Better performances than naive methods that do not utilize the subproblem overlap.

Dynamic time-warp

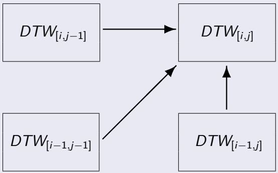
Dynamic programming algorithm

DTW Algorithm:

For each signal-time i and for each signal-time j:

- Set $cost \leftarrow (x[i] - y[j])^2$
- Set the optimal distance at stage $[i, j]$ to:

$$DTW_{[i,j]} \leftarrow cost + \min \left\{ \begin{array}{c} DTW_{[i,j-1]} \\ DTW_{[i-1,j-1]} \\ DTW_{[i-1,j]} \end{array} \right\}$$



Optimal distance: $DTW_{[m,n]}$ (where m & n are lengths of signals).

Optimal alignment: backtracking the path leading to $DTW_{[m,n]}$ via min-cost choices of the algorithm

Dynamic time-warp

Remark about earth-mover distances (EMD)

What is the cost of transforming one distribution to another?

$$EMD_p^p(x, y) = \min\{\sum_{i=1}^{n}\sum_{j=1}^{n} |i-j|^p\,\Omega_{ij} \ :$$
$$\sum_{j=1}^{n}\Omega_{ij} = x[i] \wedge \sum_{i=1}^{n}\Omega_{ij} = y[j]\}$$

where Ω is a moving strategy (transferring Ω_{ij} mass from i to j).

Can be solved with the Hungarian algorithm, but more efficient methods exist and rely on wavelets and mathematical analysis.

Combining similarities

To combine similarities of different attributes we can consider several alternatives:

1. Transform all the attributes to conform to the same similarity/distance metric
2. Use weighted average to combine similarities $a(x, y) = \sum_{i=1}^{n} w_i a_i(x, y)$ or distances $d^2(x, y) = \sum_{i=1}^{n} w_i d_i^2(x, y)$ with $\sum_{i=1}^{n} w_i = 1$.
3. Consider asymmetric attributes by defining binary flags $\delta_i(x, y) \in \{0, 1\}$ that mark whether two data points share comparable information in affinity i and then combine only comparable information by $a(x, y) = \frac{\sum_{i=1}^{n} w_i \delta_i(x,y) a_i(x,y)}{\sum_{i=1}^{n} \delta_i(x,y)}$.

Summary

To compare data points we can either

1. quantify how similar they are with a similarity or affinity metric, or
2. quantify how different they are with a dissimilarity or a distance metric.

There are many possible metrics (e.g., Euclidean, Mahalanobis, Hamming, Gaussian, Cosine, Jaccard), and the choice of which one to use depends on both the task and the input data.

It is sometimes useful to consider several different metrics and then combine them together. Alternatively, data preprocessing can be done to transform all the data to conform with a single metric.