Distance selection

Victor Kitov

v.v.kitov@yandex.ru

Yandex
School of Data Analysis.

Table of Contents

Distance metric selection[1]

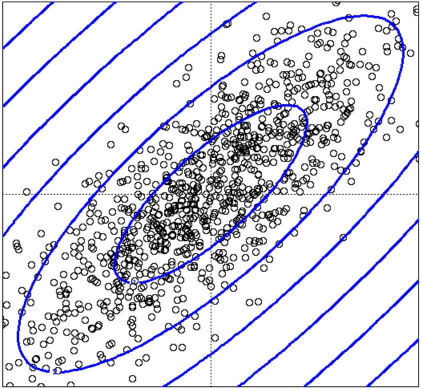
Metric	$d(x,z)$
Euclidean	$\sqrt{\sum_{i=1}^{D}(x^i - z^i)^2}$
L_p	$\sqrt[p]{\sum_{i=1}^{D}(x^i - z^i)^p}$
L_∞	$\max_{i=1,2,...D} \lvert x^i - z^i \rvert$
L_1	$\sum_{i=1}^{D} \lvert x^i - z^i \rvert$
Canberra	$\frac{1}{D}\sum_{i=1}^{D} \frac{\lvert x^i - z^i \rvert}{\lvert x^i + z^i \rvert}$
Lance-Williams	$\frac{\sum_{i=1}^{D} \lvert x^i - z^i \rvert}{\sum_{i=1}^{D} \lvert x^i + z^i \rvert}$

Comments:

- prone to curse of dimensionality
- performance↓ as we have more irrelevant features.

[1] Plot iso-lines for L_1, L_2, L_∞ metrics.

Correlated variables



- Objects along $y = x$ line are more similar, than along $y = -x$.
- How to measure similarity?

Whitening transformation

- $x \sim F(\mu, \Sigma)$, $\mu = \mathbb{E}[\mu]$, $\Sigma = cov(x, x)$, $\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{DxD}$
- Whitening transformation:

$$z = \Sigma^{-1/2}(x - \mu)$$

- Properties[2]:

$$Ez = \mathbf{0},\ cov[z, z] = I.$$

[2] Prove them.

Distance between normalized feature vectors

- Distance between normalized x and x' is equal to Euclidean distance between $z=\Sigma^{-1/2}(x-\mu)$ and $z'=\Sigma^{-1/2}(x'-\mu)$:

$$\begin{aligned}
\rho_M(x,x') &= \rho_E(z,z')=\sqrt{(z-z')^T(z-z')}= \\
&= \sqrt{\left(\Sigma^{-1/2}(x-x')\right)^T\Sigma^{-1/2}(x-x')} \\
&= \sqrt{(x-x')^T\Sigma^{-1/2}\Sigma^{-1/2}(x-x')} \\
&= \sqrt{(x-x')^T\Sigma^{-1}(x-x')}
\end{aligned}$$

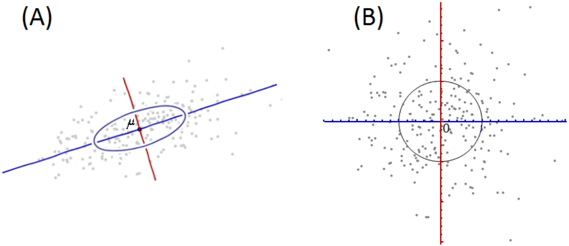
- This is known as *Mahalonobis distance*[3].

[3] How will Mahalanobis distance look like when features are uncorrelated? Interpret the result.

Distance between whitened objects (Mahalanobis distance)

(A): correlated feature space: objects and unit sphere $\{x : \rho_M(x,\mu)^2 = (x-\mu)^T \Sigma^{-1}(x-\mu) = 1\}$[4].

(B): whitened feature space: objects and unit sphere $\{z : \rho_E(z,0)^2 = 1\}$.



[4] Prove that this would be an ellipse. Hint: use spectral factorization and change of basis to eigenvectors if Σ.

Table of Contents

Distance between categorical vectors

- Suppose $x \in \mathbb{R}^D$ consists of D categorical features, $\rho(x, z)-?$
- We can transform x to one-hot encoding, but $0 = 0$ will overweight $0 \neq 1$ and $1 \neq 0$.
- Similarity functions are calculated as sums:

$$sim(x, z) = \sum_{d=1}^{D} sim(x^d, z^d)$$

- Possibility:

$$sim(x^d, z^d) = \mathbb{I}[x^d = z^d]$$

- Drawback: equal treatment of common and specific categories.
    - e.g. $x^d = illness$, $95\% : x^d =$ 'healthy', $1\% : x^d =$ 'flu', $x^d =$ 'pneumonia', etc.
        - healthy patients are not as similar as ill with pneumonia

Distance between categorical vectors

$$sim(x,z)=\sum_{d=1}^{D} sim(x^d,z^d)$$

- Solution:

$$sim(x^d,z^d)=\begin{cases}0, & x^d\neq z^d\\ K(p(x^d)) & x^d=z^d,\ \text{for some }\ \downarrow K(u)\end{cases}$$

Common choices: $K(p(x^d))=\frac{1}{p(x^d)^2}$, $K(p(x^d))=1-p(x^d)^2$.

Mixture of numeric and categorical features

- Suppose $x = (x_{num}, x_{cat})$, where
  - x_{num}: vector of numeric features
  - x_{cat}: vector of categorical features
- Similarity:

$$sim(x, z) = \lambda sim_{num}(x_{num}, z_{num}) + (1 - \lambda) sim_{cat}(x_{cat}, z_{cat})$$

- $\lambda \in (0, 1)$ - hyperparameter, measuring relative importance of numeric features.
  - by default λ=fraction of numeric features.
- $sim(x_{num}, z_{num}) = F(\rho(x_{num}, z_{num}))$ for some $\downarrow F(u)$, e.g.

$$sim_{num}(x_{num}, z_{num}) = \frac{1}{1 + \rho(x_{num}, z_{num})}$$

Mixture of numeric and categorical features

- Important to convert similarities to equal scale:

$$sim_{num}(x_{num}, z_{num}) = \frac{sim_{num}(x_{num}, z_{num})}{\sigma_{num}}$$
$$sim_{cat}(x_{cat}, z_{cat}) = \frac{sim_{cat}(x_{cat}, z_{cat})}{\sigma_{cat}}$$

- σ_{num}, σ_{cat} - standard deviations of $sim_{num}(\cdot,\cdot)$, $sim_{cat}(\cdot,\cdot)$ for random subsamples of objects.

Similarity between sets / binary vectors

- Jaccard similarity for sets A, B:

$$sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

- Jaccard similarity for binary vectors a, b:

$$sim(A,B) = \frac{\sum_{d=1}^{D} a^d b^d}{\sum_{d=1}^{D} a^d + b^d - a^d b^d}$$

- Possible use cases:
  - purchase - basket of goods
  - document-set of words
  - user profile - set of preferences

Table of Contents

Text representation

- Suppose there are D unique words in the language $w_1, ...w_D$.
- Object=document, having word w_i occurring n_i times, $i = 1, 2, ...D$.

$$x = (\mathbb{I}[n_1 > 0], ...\mathbb{I}[n_D > 0])$$

$$x = \left(\frac{n_1}{n}, ...\frac{n_D}{n}\right), \quad n = \sum_{i=1}^{D} n_i\text{-document length}$$

- Word counts representation:

$$x = (n_1, ...n_D)$$

Text representation

- Give higher weight to rare words:

$$x = \left(n_1 \frac{N}{N_1}, ... n_D \frac{N}{N_D}\right)$$

  - N - total number of documents in the collection,
  - N_k - number of documents, containing word w_k.
- Decrease impact of too frequent words inside document

$$x = \left(\ln(1+n_1)\frac{N}{N_1}, ... \ln(1+n_D)\frac{N}{N_D}\right)$$

- Decrease impact of too rare words in the collection:

$$x = \left(\ln(1+n_1)\ln\frac{N}{N_1}, ... \ln(1+n_D)\ln\frac{N}{N_D}\right)$$

  - may use $\sqrt{\cdot}$ instead of $\ln(\cdot)$ as shrinking transformation.

Cosine similarity

- *Cosine similarity* is most popular for documents comparison:

$$sim(x,z)=\frac{x^T z}{\|x\|\,\|z\|}=\frac{\sum_{i=1}^{D}x^i z^i}{\sqrt{\sum_{i=1}^{D}(x^i)^2}\sqrt{\sum_{i=1}^{D}(z^i)^2}}$$

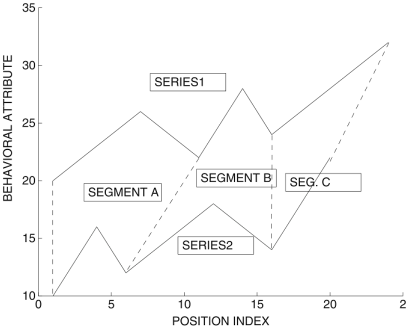
  - $\langle x,z\rangle = x^T z = \|x\|\,\|z\|\,cos(\alpha)$, where α - is the angle between x and z.
  - so cosine similarity is invariant to document length, because it depends on $\angle(x,z)$, not $\|x\|, \|z\|$.

Table of Contents

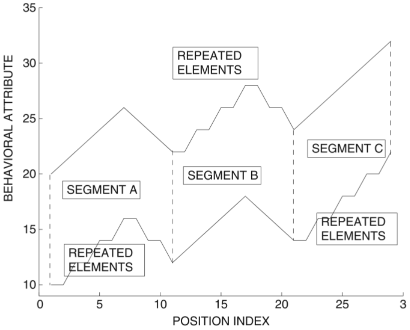
Comparing time series

- Consider time series f_t.
- May require standardization $\frac{f_t - \mathbb{E}f_t}{\sigma(f_t)}$
  - e.g. stock prices may vary similarly but around different mean values and with different magnitude.
- May require time standardization $f_t \to f_{at}$, $a > 0$
  - e.g. speech recognition, sounds can be pronounced slowly or fast
- Time standardization can be variable $f_t \to f_{a(t)}$ for some monotonous function $a(t)$.
  - called dynamic time warping
  - efficient polynomial time algorithm exists

Dynamic time warping (DTW) distance



(a) Original series



(b) Warped series

Table of Contents

5 Comparing strings
- Edit distance
- Longest common subsequence

Edit distance

- Sequences: of letters (words), of words (phrases), of nucleotides (DNA sequences), etc.
- Minimum edit distance between two strings - the minimum number of editing operations (insertion, deletion, substitution) needed to transform one string into another.
    - each editing operation has cost 1
    - however we may assign different costs
- Applications:
    - error correction:
        - e.g. graffe <-> giraffe
    - named entity recognition
        - e.g. Stanford President John Hennessy <-> Stanford University President John Hennessy

Example

Distance from [intention] to [execution] is 5.

- Optimal (minimum loss) conversion path:

```
i n t e n t i o n
                    <-- delete i
n t e n t i o n
                    <-- substitute n by e
e t e n t i o n
                    <-- substitute t by x
e x e n t i o n
                    <-- insert u
e x e n u t i o n
                    <-- substitute n by c
e x e c u t i o n
```

5 Comparing strings
- Edit distance
- Longest common subsequence

Longest common subsequence

- Common subsequence: matching elements in x and z neen not be contiguous (come immediately one after another).
- Common subsequence(abcde, xbyzcdw) = bcd
- Application example: how large were changes to original file after modification?

Table of Contents

Metric learning

Equal treatment of features:

$$\rho(x,z)=\sqrt{(x^1-z^1)^2+...+(x^D-z^D)^2}$$

- But for different applications different features should be more important!
  - e.g. image analysys: pose detection and identity recognition.

Metric learning

Equal treatment of features:

$$\rho(x,z)=\sqrt{(x^1-z^1)^2+...+(x^D-z^D)^2}$$

- But for different applications different features should be more important!
  - e.g. image analysys: pose detection and identity recognition.

Custom treatment of features with weights $w_1,...w_D$:

$$\rho(x,z|w)=\sqrt{w_1(x^1-z^1)^2+...+w_D(x^D-z^D)^2}$$

How to find weights?

Metric learning

- Define

$$S = \{(i,j): \; x_i \text{ is similar to } x_j\} \qquad (\text{e.g. } y_i = y_j)$$

$$D = \{(i,j): \; x_i \text{ is dissimilar to } x_j\} \qquad (\text{e.g. } y_i \neq y_j)$$

- We may solve:

$$w = \arg\min_w \left\{ \sum_{(i,j)\in S} (\rho(x_i, x_j|w) - 0)^2 + \sum_{(i,j)\in D} (\rho(x_i, x_j|w) - 1)^2 \right\}$$

- Any parametrized metric $\rho(x, z|w)$ can be used.
- Other approaches exist.

Summary

- Selecting proper distance is important
  - more important than tuning ML algorithm
- Each data type has its own distance functions:
  - numeric vectors
  - categorical vectors
  - time series
  - sequences
- Distance can be tuned using supervised information.