Two or three things we do (not) know about distances

Benjamin Paaßen

MiWoCI, 2017-07-26, Mittweida

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory
- **Structure:** Motivation by historic cognitive science literature (60s to 2000s)

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory
- **Structure:** Motivation by historic cognitive science literature (60s to 2000s)
- What this talk is **not** about:

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory
- **Structure:** Motivation by historic cognitive science literature (60s to 2000s)
- What this talk is **not** about:
  - Relational machine learning methods (refer to, for example, Hammer and Hasenfuss 2010; Hammer, Hofmann, et al. 2014; Schleif and Tino 2015)

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory
- **Structure:** Motivation by historic cognitive science literature (60s to 2000s)
- What this talk is **not** about:
  - Relational machine learning methods (refer to, for example, Hammer and Hasenfuss 2010; Hammer, Hofmann, et al. 2014; Schleif and Tino 2015)
  - Metric axioms and metric taxonomies (has been done very well by Nebel et al. 2017)

Introduction

- **Aim:** Discuss some advanced questions of distances for machine learning, namely
  - (Pseudo-)Euclidean embeddings
  - Indefinite Learning
  - Edit Distance Theory
- **Structure:** Motivation by historic cognitive science literature (60s to 2000s)
- What this talk is **not** about:
  - Relational machine learning methods (refer to, for example, Hammer and Hasenfuss 2010; Hammer, Hofmann, et al. 2014; Schleif and Tino 2015)
  - Metric axioms and metric taxonomies (has been done very well by Nebel et al. 2017)
  - Similarities and Kernels (you may also refer to Nebel et al. 2017; Schleif and Tino 2015)

Euclidean Embeddings

Spatial Representations of Knowledge

Shepard (1962)

The notion of nearness or **proximity**, which is objectively defined only for pairs of objects in physical space, tends to be carried over to very different situations where the **space in which entities can be closer together or further apart** is not at all evident.

Spatial Representations of Knowledge

Shepard (1962)

The notion of nearness or **proximity**, which is objectively defined only for pairs of objects in physical space, tends to be carried over to very different situations where the **space in which entities can be closer together or further apart** is not at all evident.

- Assume that objects live in an underlying, **Euclidean** space, i.e.: Given pairwise distances $\boldsymbol{D} \in \mathbb{R}^{m \times m}$, find an **Euclidean embedding** $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ $(n \leq m)$, such that $\|\vec{x}_i - \vec{x}_j\| \approx \boldsymbol{D}_{ij}$

Spatial Representations of Knowledge

Shepard (1962)

The notion of nearness or **proximity**, which is objectively defined only for pairs of objects in physical space, tends to be carried over to very different situations where the **space in which entities can be closer together or further apart** is not at all evident.

- Assume that objects live in an underlying, **Euclidean** space, i.e.: Given pairwise distances $\boldsymbol{D} \in \mathbb{R}^{m \times m}$, find an **Euclidean embedding** $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ $(n \leq m)$, such that $\|\vec{x}_i - \vec{x}_j\| \approx \boldsymbol{D}_{ij}$

$\Rightarrow$ non-metric Multi-Dimensional Scaling (MDS)

Spatial Representations of Knowledge

Shepard (1962)

The notion of nearness or **proximity**, which is objectively defined only for pairs of objects in physical space, tends to be carried over to very different situations where the **space in which entities can be closer together or further apart** is not at all evident.

- Assume that objects live in an underlying, **Euclidean** space, i.e.: Given pairwise distances $\boldsymbol{D} \in \mathbb{R}^{m \times m}$, find an **Euclidean embedding** $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ $(n \leq m)$, such that $\|\vec{x}_i - \vec{x}_j\| \approx \boldsymbol{D}_{ij}$

$\Rightarrow$ non-metric Multi-Dimensional Scaling (MDS)

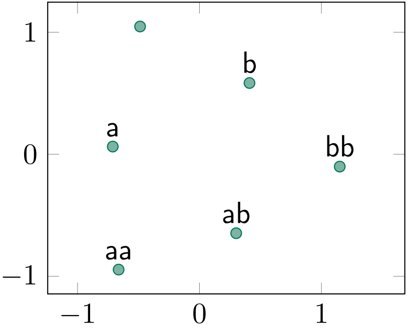
- Then use $\boldsymbol{X}$ for subsequent machine learning (implicitly or explicitly)

Embedding Example

- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance

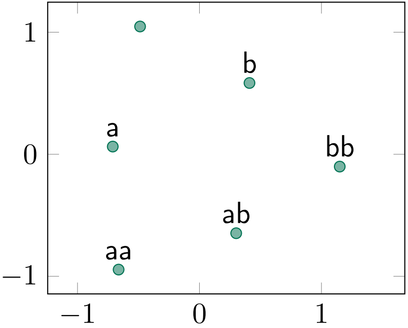
Embedding Example

- Data set: {"", "a", "b", "aa", "ab", "bb"} with Levenshtein-Distance



Embedding Example

- Data set: {"", "a", "b", "aa", "ab", "bb"} with Levenshtein-Distance



- Note: Information loss increases with higher **intrinsic** data dimensionality

Representing Prototypes (the "distance trick")

- Assume that prototypes are **affine** combinations of data points, i.e.:

$$\vec{w}_k = \sum_{i=1}^{m} \alpha_{k,i} \cdot \vec{x}_i \qquad \text{s.t.} \sum_{i=1}^{m} \alpha_{k,i} = 1 \quad \forall k \in \{1, \ldots, K\} \tag{1}$$

Representing Prototypes (the “distance trick”)

- Assume that prototypes are **affine** combinations of data points, i.e.:

$$\vec{w}_k = \sum_{i=1}^{m} \alpha_{k,i} \cdot \vec{x}_i \qquad \text{s.t.} \sum_{i=1}^{m} \alpha_{k,i} = 1 \quad \forall k \in \{1, \ldots, K\} \tag{1}$$

$$\iff \boldsymbol{W} = \boldsymbol{A} \cdot \boldsymbol{X} \qquad \text{s.t.} \sum_{i=1}^{m} \boldsymbol{A}_{k,i} = 1 \quad \forall k \in \{1, \ldots, K\} \tag{2}$$

Representing Prototypes (the "distance trick")

- Assume that prototypes are **affine** combinations of data points, i.e.:

$$\vec{w}_k = \sum_{i=1}^{m} \alpha_{k,i} \cdot \vec{x}_i \qquad \text{s.t.} \sum_{i=1}^{m} \alpha_{k,i} = 1 \quad \forall k \in \{1, \ldots, K\} \tag{1}$$

$$\iff \boldsymbol{W} = \boldsymbol{A} \cdot \boldsymbol{X} \qquad \text{s.t.} \sum_{i=1}^{m} \boldsymbol{A}_{k,i} = 1 \quad \forall k \in \{1, \ldots, K\} \tag{2}$$

- If $\boldsymbol{X}$ is an **isometric embedding**, it holds (Hammer and Hasenfuss 2010):

$$\|\vec{w}_k - \vec{x}_i\|^2 = \boldsymbol{A}_{k,:} \cdot \boldsymbol{D}_{:,i} - \frac{1}{2}\boldsymbol{A}_{k,:} \cdot \boldsymbol{D} \cdot \boldsymbol{A}_{k,:}{}^T \tag{3}$$

Indefinite Learning

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Tversky (1977)

Similarity judgments can be regarded as [...] statements of the form "*a* is like *b*."

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Tversky (1977)

Similarity judgments can be regarded as [...] statements of the form "*a* is like *b*." Such a statement is **directional**; it has a subject, *a*, and a referent, *b*, and it is **not equivalent in general** to the converse similarity statement "*b* is like *a*." [...]

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Tversky (1977)

Similarity judgments can be regarded as [...] statements of the form "*a* is like *b*." Such a statement is **directional**; it has a subject, *a*, and a referent, *b*, and it is **not equivalent in general** to the converse similarity statement "*b* is like *a*." [...] We say "the portrait resembles the person" rather than "the person resembles the portait".

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Tversky (1977)

Similarity judgments can be regarded as [...] statements of the form "*a* is like *b*." Such a statement is **directional**; it has a subject, *a*, and a referent, *b*, and it is **not equivalent in general** to the converse similarity statement "*b* is like *a*." [...] We say "the portrait resembles the person" rather than "the person resembles the portait".

- **all** metric axioms may be violated in practical cases, especially the triangular inequality (Schleif and Tino 2015)

Metric properties and non-Euclidean distances

- Distances have to conform to the **minimality**, **symmetry**, and **triangle inequality** axioms

Tversky (1977)

Similarity judgments can be regarded as [...] statements of the form "*a* is like *b*." Such a statement is **directional**; it has a subject, *a*, and a referent, *b*, and it is **not equivalent in general** to the converse similarity statement "*b* is like *a*." [...] We say "the portrait resembles the person" rather than "the person resembles the portait".

- **all** metric axioms may be violated in practical cases, especially the triangular inequality (Schleif and Tino 2015)
- Even if all axioms hold, a distance may be **non-Euclidean**

Pseudo-Euclidean Embeddings

Pękalska and Duin (2005, p. 122)

Let $\boldsymbol{D} \in \mathbb{R}^{m \times m}$ be non-negative, symmetric, and reflexive (i.e.: zero diagonal).

Pseudo-Euclidean Embeddings

Pękalska and Duin (2005, p. 122)

Let $\boldsymbol{D} \in \mathbb{R}^{m \times m}$ be non-negative, symmetric, and reflexive (i.e.: zero diagonal). Then, there exists **two** embeddings $\boldsymbol{X}^+ \in \mathbb{R}^{m \times n^+}$ and $\boldsymbol{X}^- \in \mathbb{R}^{m \times n^-}$ with $n^+ + n^- \leq m$, such that for all $i, j \in \{1, \ldots, m\}$:

Pseudo-Euclidean Embeddings

Pękalska and Duin (2005, p. 122)

Let $\boldsymbol{D} \in \mathbb{R}^{m \times m}$ be non-negative, symmetric, and reflexive (i.e.: zero diagonal). Then, there exists **two** embeddings $\boldsymbol{X}^+ \in \mathbb{R}^{m \times n^+}$ and $\boldsymbol{X}^- \in \mathbb{R}^{m \times n^-}$ with $n^+ + n^- \leq m$, such that for all $i, j \in \{1, \ldots, m\}$:

$$\boldsymbol{D}_{ij} = \sqrt{\|\vec{x^+}_i - \vec{x^+}_j\|^2 - \|\vec{x^-}_i - \vec{x^-}_j\|^2} \tag{4}$$

where $\vec{x^+}_i$ is the ith row of $\boldsymbol{X}^+$ and $\vec{x^-}_i$ is the ith row of $\boldsymbol{X}^-$.

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$
- Compute corresponding similarities via **double-centering**:

$$\boldsymbol{S} = -\frac{1}{2} \cdot \Big(\boldsymbol{D} - \underbrace{\frac{1}{m} \cdot \mathbb{1} \cdot \boldsymbol{D}}_{\text{column means}} - \underbrace{\frac{1}{m} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{row means}} + \underbrace{\frac{1}{m^2} \cdot \mathbb{1} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{matrix mean}} \Big) \tag{5}$$

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$
- Compute corresponding similarities via **double-centering**:

$$\boldsymbol{S} = -\frac{1}{2} \cdot \Big(\boldsymbol{D} - \underbrace{\frac{1}{m} \cdot \mathbb{1} \cdot \boldsymbol{D}}_{\text{column means}} - \underbrace{\frac{1}{m} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{row means}} + \underbrace{\frac{1}{m^2} \cdot \mathbb{1} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{matrix mean}} \Big) \tag{5}$$

- Eigenvalue decomposition: $\boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^T = \text{eig}(\boldsymbol{S})$

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$
- Compute corresponding similarities via **double-centering**:

$$\boldsymbol{S} = -\frac{1}{2} \cdot \Big(\boldsymbol{D} - \underbrace{\frac{1}{m} \cdot \mathbb{1} \cdot \boldsymbol{D}}_{\text{column means}} - \underbrace{\frac{1}{m} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{row means}} + \underbrace{\frac{1}{m^2} \cdot \mathbb{1} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{matrix mean}} \Big) \tag{5}$$

- Eigenvalue decomposition: $\boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^T = \text{eig}(\boldsymbol{S})$
- Select positive and negative Eigenvalues: $\boldsymbol{\Lambda}^+ = \boldsymbol{\Lambda}[:, \vec{\lambda} > 0]$ and $\boldsymbol{\Lambda}^- = \boldsymbol{\Lambda}[:, \vec{\lambda} < 0]$, where $\vec{\lambda} = \text{diag}(\boldsymbol{\Lambda})$

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$
- Compute corresponding similarities via **double-centering**:

$$\boldsymbol{S} = -\frac{1}{2} \cdot \left(\boldsymbol{D} - \underbrace{\frac{1}{m} \cdot \mathbb{1} \cdot \boldsymbol{D}}_{\text{column means}} - \underbrace{\frac{1}{m} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{row means}} + \underbrace{\frac{1}{m^2} \cdot \mathbb{1} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{matrix mean}} \right) \tag{5}$$

- Eigenvalue decomposition: $\boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^T = \text{eig}(\boldsymbol{S})$
- Select positive and negative Eigenvalues: $\boldsymbol{\Lambda}^+ = \boldsymbol{\Lambda}[:, \vec{\lambda} > 0]$ and $\boldsymbol{\Lambda}^- = \boldsymbol{\Lambda}[:, \vec{\lambda} < 0]$, where $\vec{\lambda} = \text{diag}(\boldsymbol{\Lambda})$
- Generate embeddings: $\boldsymbol{X}^+ = \boldsymbol{U} \cdot \sqrt{\boldsymbol{\Lambda}^+}$, and $\boldsymbol{X}^- = \boldsymbol{U} \cdot \sqrt{-\boldsymbol{\Lambda}^-}$

Pseudo-Euclidean Embeddings (contd.)

- Input: squared (!) matrix $\boldsymbol{D}$
- Compute corresponding similarities via **double-centering**:

$$\boldsymbol{S} = -\frac{1}{2} \cdot \left(\boldsymbol{D} - \underbrace{\frac{1}{m} \cdot \mathbb{1} \cdot \boldsymbol{D}}_{\text{column means}} - \underbrace{\frac{1}{m} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{row means}} + \underbrace{\frac{1}{m^2} \cdot \mathbb{1} \cdot \boldsymbol{D} \cdot \mathbb{1}}_{\text{matrix mean}} \right) \quad (5)$$

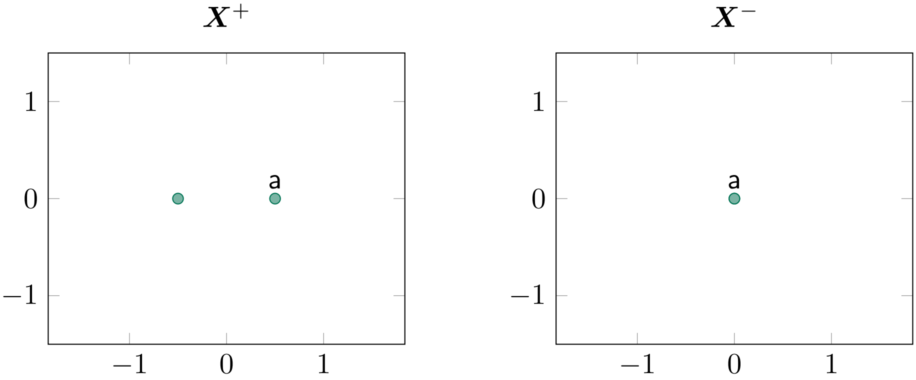
- Eigenvalue decomposition: $\boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^T = \text{eig}(\boldsymbol{S})$
- Select positive and negative Eigenvalues: $\boldsymbol{\Lambda}^+ = \boldsymbol{\Lambda}[:, \vec{\lambda} > 0]$ and $\boldsymbol{\Lambda}^- = \boldsymbol{\Lambda}[:, \vec{\lambda} < 0]$, where $\vec{\lambda} = \text{diag}(\boldsymbol{\Lambda})$
- Generate embeddings: $\boldsymbol{X}^+ = \boldsymbol{U} \cdot \sqrt{\boldsymbol{\Lambda}^+}$, and $\boldsymbol{X}^- = \boldsymbol{U} \cdot \sqrt{-\boldsymbol{\Lambda}^-}$
- A distance is **Euclidean** iff there are no negative Eigenvalues

Embedding Example

- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance

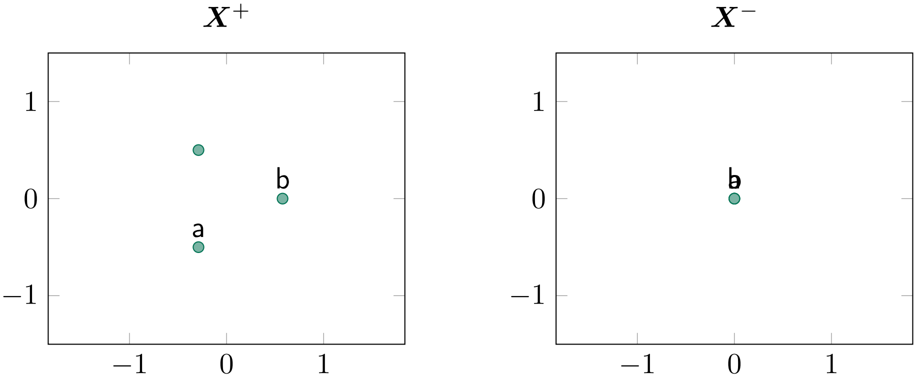
Embedding Example

- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance



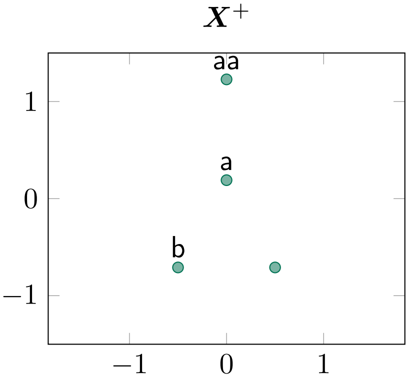
Embedding Example

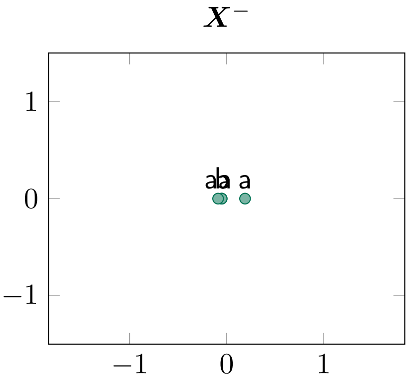
- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance



Embedding Example

- Data set: {"", "a", "b", "aa", "ab", "bb"} with Levenshtein-Distance

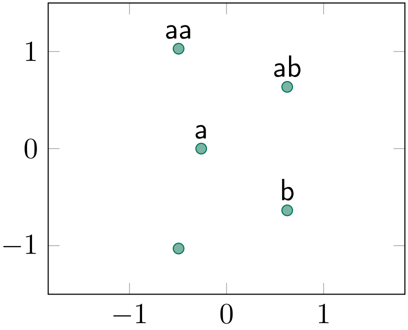




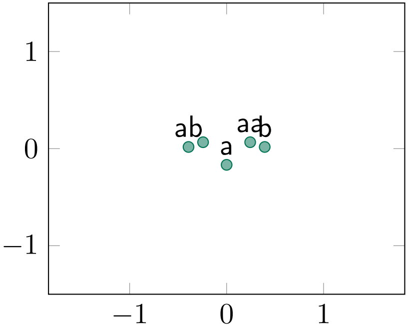
Embedding Example

- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance

$\boldsymbol{X}^+$

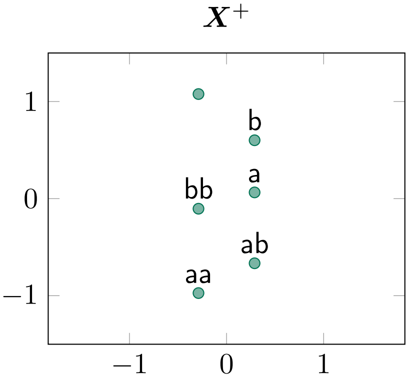


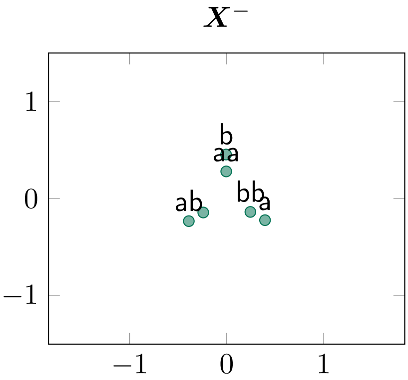
$\boldsymbol{X}^-$



Embedding Example

- Data set: {“”, “a”, “b”, “aa”, “ab”, “bb”} with Levenshtein-Distance





Challenges in non-Euclidean distances

- Distance trick still works

Challenges in non-Euclidean distances

- Distance trick still works, but prototype-datapoints-distances can become **negative**

Challenges in non-Euclidean distances

- Distance trick still works, but prototype-datapoints-distances can become **negative**
- Eigenvalue correction may distort the distances significantly (Nebel et al. 2017; Schleif and Tino 2015)

Challenges in non-Euclidean distances

- Distance trick still works, but prototype-datapoints-distances can become **negative**
- Eigenvalue correction may distort the distances significantly (Nebel et al. 2017; Schleif and Tino 2015)
- Optimization problems become (even more) non-convex

Challenges in non-Euclidean distances

- Distance trick still works, but prototype-datapoints-distances can become **negative**
- Eigenvalue correction may distort the distances significantly (Nebel et al. 2017; Schleif and Tino 2015)
- Optimization problems become (even more) non-convex

$\Rightarrow$ Lack of methods which can deal with non-Euclidean distances explicitly, without information loss (?) (Schleif and Tino 2015)

Edit Distances

Structural Alignment and Representational Distortion

Hodgetts, Hahn, and Chater (2009)

Nevertheless, both accounts appear to have **fundamental limitations**. [...]

Structural Alignment and Representational Distortion

Hodgetts, Hahn, and Chater (2009)

Nevertheless, both accounts appear to have **fundamental limitations**. [...] Real-world objects are not merely represented as lists of features or dimensions but **represented in a structured way** that considers not only the composite elements but the relations between these different elements.

Structural Alignment and Representational Distortion

Hodgetts, Hahn, and Chater (2009)

Nevertheless, both accounts appear to have **fundamental limitations**. [...] Real-world objects are not merely represented as lists of features or dimensions but **represented in a structured way** that considers not only the composite elements but the relations between these different elements.

- Measure **distances between structures** via **edit** or **alignment** distances (Paassen, Mokbel, and Hammer 2016)

Alignment distance

- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ

Alignment distance

- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ
- Assign a cost C to each pair in $(\Sigma \cup \{-\}) \times (\Sigma \cup \{-\})$

Alignment distance

- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ
- Assign a cost C to each pair in $(\Sigma \cup \{-\}) \times (\Sigma \cup \{-\})$
- Alignment distance $d(x, y)$: Cost of minimum assignment of nodes in x to nodes in y under some constraints to the assignment.

Alignment distance

- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ
- Assign a cost C to each pair in $(\Sigma \cup \{-\}) \times (\Sigma \cup \{-\})$
- Alignment distance $d(x, y)$: Cost of minimum assignment of nodes in x to nodes in y under some constraints to the assignment.

$\Rightarrow$ Equivalent to edit distance for sequences

Alignment distance

- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ
- Assign a cost C to each pair in $(\Sigma \cup \{-\}) \times (\Sigma \cup \{-\})$
- Alignment distance $d(x, y)$: Cost of minimum assignment of nodes in x to nodes in y under some constraints to the assignment.

$\Rightarrow$ Equivalent to edit distance for sequences

$\Rightarrow$ Triangular inequality in general hard to verify

Alignment distance

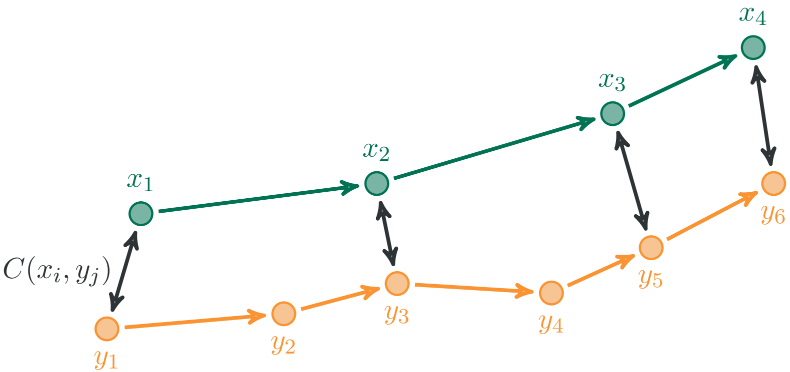
- Assume a set of **graphs** $x \in \mathcal{X}$ over a **node alphabet** Σ
- Assign a cost C to each pair in $(\Sigma \cup \{-\}) \times (\Sigma \cup \{-\})$
- Alignment distance $d(x, y)$: Cost of minimum assignment of nodes in x to nodes in y under some constraints to the assignment.

$\Rightarrow$ Equivalent to edit distance for sequences

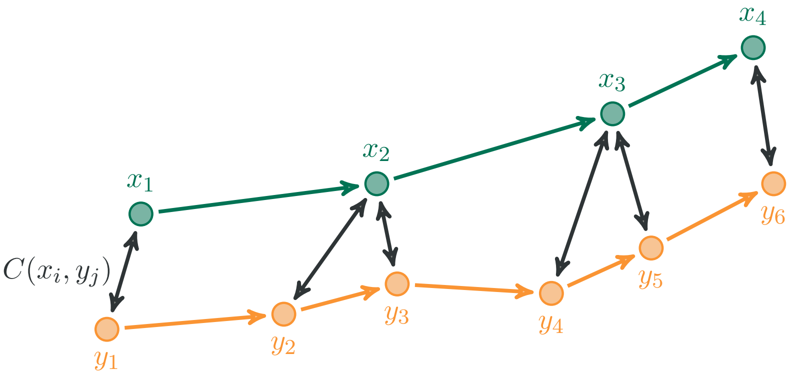
$\Rightarrow$ Triangular inequality in general hard to verify

$\Rightarrow$ Link to minimum bipartite matching problems

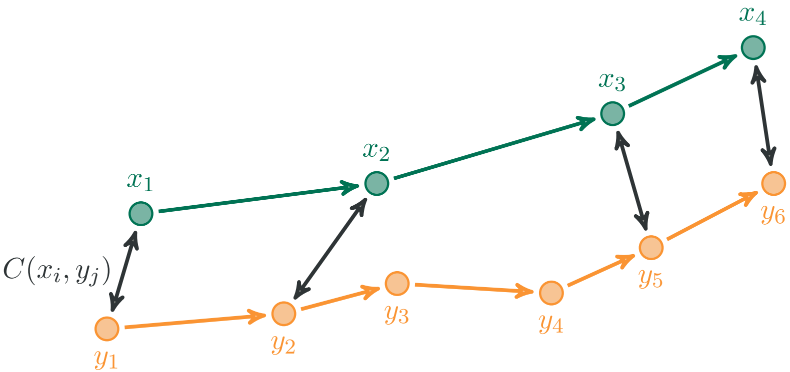
Alignment Distance Example



Alignment Distance Example



Alignment Distance Example



Edit Distance

(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$

Edit Distance

(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$
- Create a **legal move graph** $G = (\mathcal{X}, E)$ where $E = \{(x, y) | \delta \in \Delta : \delta(x) = y\}$

Edit Distance

(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$
- Create a **legal move graph** $G = (\mathcal{X}, E)$ where $E = \{(x, y) | \delta \in \Delta : \delta(x) = y\}$
- Assign a **cost** C to each edge in G

Edit Distance

(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$
- Create a **legal move graph** $G = (\mathcal{X}, E)$ where $E = \{(x, y) | \delta \in \Delta : \delta(x) = y\}$
- Assign a **cost** C to each edge in G
- Edit distance $d(x, y)$: length of shortest path from x to y in G

Edit Distance

(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$
- Create a **legal move graph** $G = (\mathcal{X}, E)$ where $E = \{(x, y) | \delta \in \Delta : \delta(x) = y\}$
- Assign a **cost** C to each edge in G
- Edit distance $d(x, y)$: length of shortest path from x to y in G

$\Rightarrow$ Metric properties (but **not** Euclideanicity!) achievable under moderate constraints to Δ and C

Edit Distance

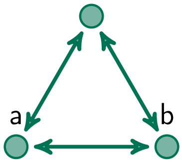
(Paaßen et al., 2017, work in progress ...)

- Assume a set of possible **edits** $\delta : \mathcal{X} \to \mathcal{X}$ for some set of structured data $\mathcal{X}$
- Create a **legal move graph** $G = (\mathcal{X}, E)$ where $E = \{(x, y) | \delta \in \Delta : \delta(x) = y\}$
- Assign a **cost** C to each edge in G
- Edit distance $d(x, y)$: length of shortest path from x to y in G

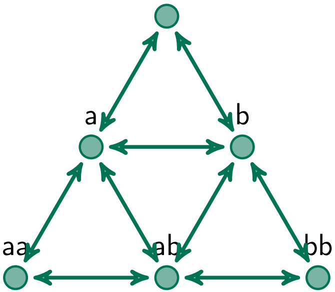
$\Rightarrow$ Metric properties (but **not** Euclideanicity!) achievable under moderate constraints to Δ and C

$\Rightarrow$ Supports search for **pre-images** (e.g. object that corresponds to prototype)

Edit Distance Example



Edit Distance Example



Conclusion

- quasi-metrics (distances without triangular inequality) permit a minimal (w.r.t. dimensionality), isometric **embedding**

Conclusion

- quasi-metrics (distances without triangular inequality) permit a minimal (w.r.t. dimensionality), isometric **embedding**
- Can be used **implicitly** via distance trick

Conclusion

- quasi-metrics (distances without triangular inequality) permit a minimal (w.r.t. dimensionality), isometric **embedding**
- Can be used **implicitly** via distance trick
- Distances may be **non-Euclidean** (can your method deal with that? Can you afford to enforce Euclideanicity?)

Conclusion

- quasi-metrics (distances without triangular inequality) permit a minimal (w.r.t. dimensionality), isometric **embedding**
- Can be used **implicitly** via distance trick
- Distances may be **non-Euclidean** (can your method deal with that? Can you afford to enforce Euclideanicity?)
- edit/alignment distances offer a general view on structured data

Conclusion

- quasi-metrics (distances without triangular inequality) permit a minimal (w.r.t. dimensionality), isometric **embedding**
- Can be used **implicitly** via distance trick
- Distances may be **non-Euclidean** (can your method deal with that? Can you afford to enforce Euclideanicity?)
- edit/alignment distances offer a general view on structured data

Thank you for your attention!

Literature I

Hammer, Barbara and Alexander Hasenfuss (2010). "Topographic Mapping of Large Dissimilarity Data Sets". In: *Neural Computation* 22.9, pp. 2229–2284. DOI: 10.1162/NECO_a_00012.

Hammer, Barbara, Daniela Hofmann, et al. (2014). "Learning vector quantization for (dis-)similarities". English. In: *Neurocomputing* 131, pp. 43–51.

Hodgetts, Carl J., Ulrike Hahn, and Nick Chater (2009). "Transformation and alignment in similarity". In: *Cognition* 113.1, pp. 62–79. DOI: 10.1016/j.cognition.2009.07.010.

Nebel, D. et al. (2017). "Types of (dis-)similarities and adaptive mixtures thereof for improved classification learning". In: *Neurocomputing*. in press. DOI: 10.1016/j.neucom.2016.12.091.

Literature II

Paassen, Benjamin, Bassam Mokbel, and Barbara Hammer (2016). "Adaptive structure metrics for automated feedback provision in intelligent tutoring systems". In: *Neurocomputing* 192, pp. 3–13. DOI: 10.1016/j.neucom.2015.12.108.

Pękalska, Elżbieta and Robert P. W. Duin (2005). *The dissimilarity representation for pattern recognition: foundations and applications*. Ed. by H. Bunke and P. S. P. Wang. Vol. 64. Series in Machine Perception and Artificial Intelligence. Singapore. ISBN: 981-256-530-2.

Schleif, Frank-Michael and Peter Tino (2015). "Indefinite Proximity Learning: A Review". In: *Neural Computation* 27.10, pp. 2039–2096. DOI: 10.1162/NECO_a_00770.

Shepard, Roger N. (1962). "The analysis of proximities: Multidimensional scaling with an unknown distance function. I." In: *Psychometrika* 27.2, pp. 125–140. DOI: 10.1007/BF02289630.

Literature III

Tversky, Amos (1977). "Features of similarity". In: *Psychological Review* 84.4, pp. 327–352. DOI: 10.1037/0033-295X.84.4.327.