Distance Metric Learning from Pairwise Proximities

Brian McFee
bmcfee@cs.ucsd.edu

UCSD CSE

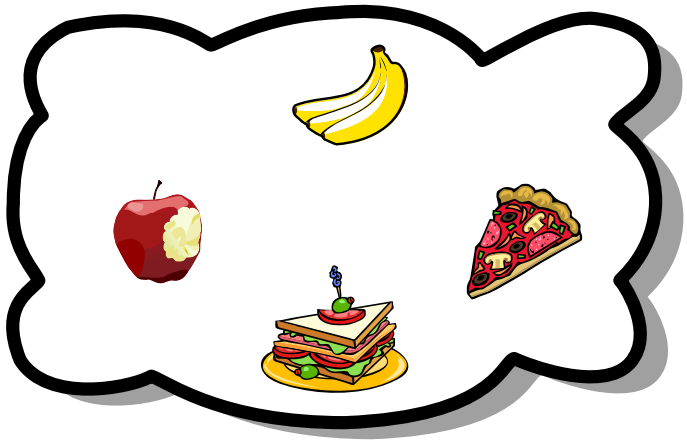
August 12, 2008

What is metric learning?

- Given a set of objects $\mathcal{X}$, learn a mapping of $\mathcal{X}$ into a metric space.
- $\mathcal{X}$ may either contain vectors in $\mathbb{R}^D$, or arbitrary objects.
- Distance after embedding should reflect similarity in $\mathcal{X}$.

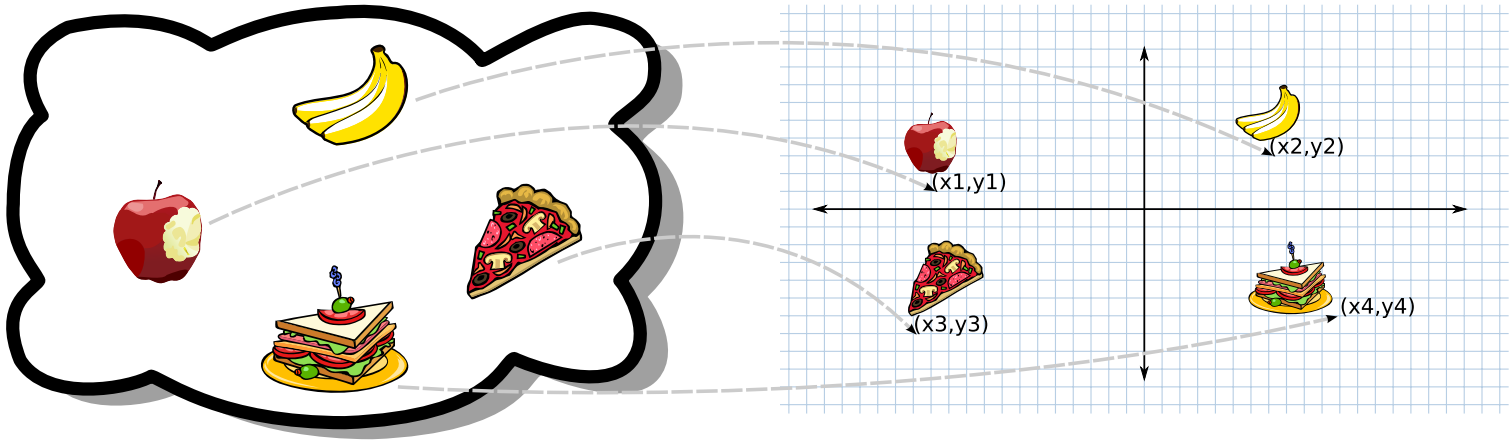
What is metric learning?

- Given a set of objects $\mathcal{X}$, learn a mapping of $\mathcal{X}$ into a metric space.
- $\mathcal{X}$ may either contain vectors in $\mathbb{R}^D$, or arbitrary objects.
- Distance after embedding should reflect similarity in $\mathcal{X}$.



What is metric learning?

- Given a set of objects $\mathcal{X}$, learn a mapping of $\mathcal{X}$ into a metric space.
- $\mathcal{X}$ may either contain vectors in $\mathbb{R}^D$, or arbitrary objects.
- Distance after embedding should reflect similarity in $\mathcal{X}$.



How can we use metric learning?

- We can analyze non-vectorial data.
- After embedding $\mathcal{X}$, we can apply standard algorithms for:
  - clustering, classification, neighbor search, visualization, etc.
- We can model subjective similarity.
  - This is important for multimedia data!

Euclidean embedding

- We'll focus on embeddings into Euclidean space $\mathbb{R}^d$:

$$d(x, y)^2 = (x - y)^\mathsf{T}(x - y).$$

- Euclidean space is familiar, natural for visualization ($d = 2$ or 3).
- Euclidean space is mathematically convenient.

Proximity constraints

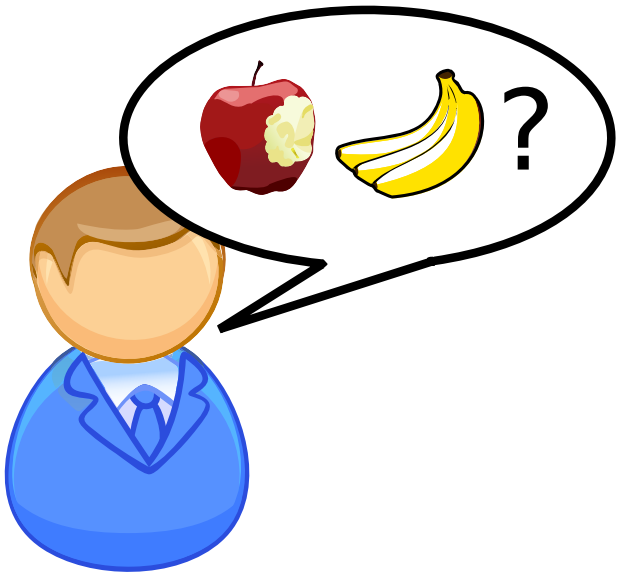
- How should we describe similarity for a pair $x, y \in \mathcal{X}$?
- It depends on the application.
- Algorithm design depends on the type of similarity.
- *Proximity* is a catch-all term for descriptions of distance, similarity or dissimilarity.

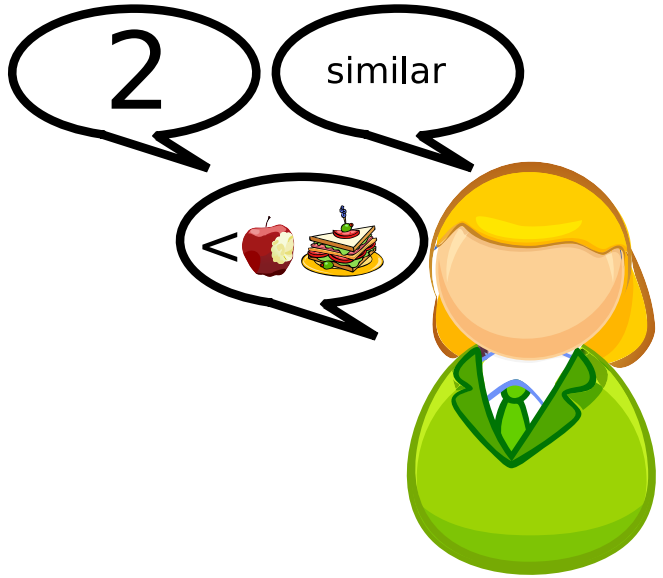
Notions of proximity

Quantitative	distance from x to y
Discrete proximity	$\{x, y\}$ are similar, $\{w, z\}$ are dissimilar
Partial order	$\{x, y\}$ are more similar than $\{w, z\}$

Acquiring constraints

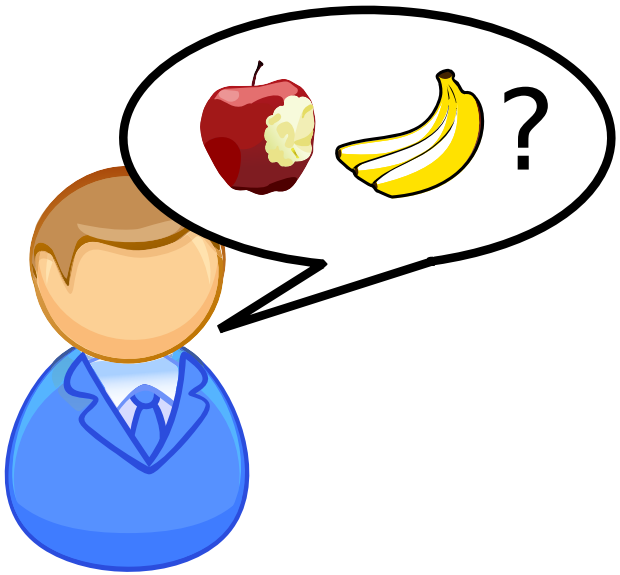
- Where do proximity constraints come from? Humans!
- For a music task, we might ask:
  - How far is Song A from Song B?
  - Is Song A similar to Song B?
  - Is Song A more similar to Song B or to Song C?

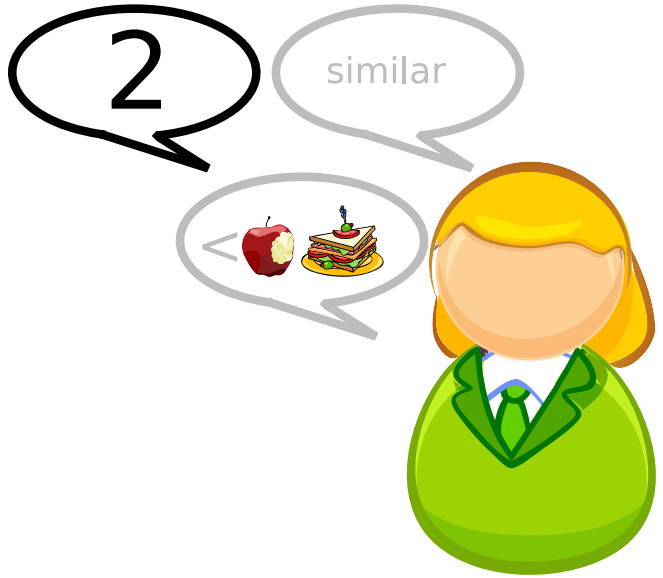




Acquiring constraints

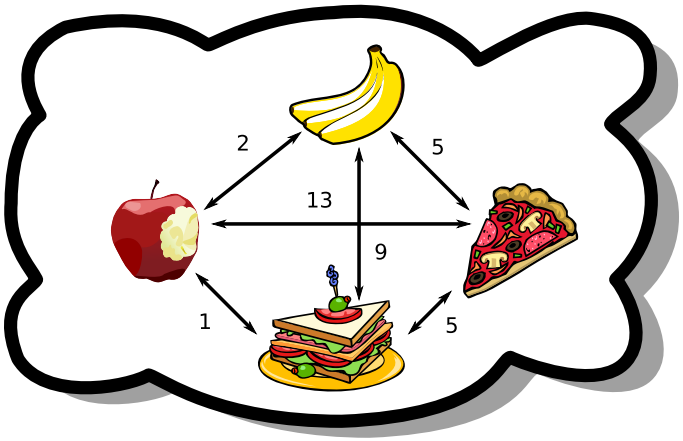
- Where do proximity constraints come from? Humans!
- For a music task, we might ask:
  - How far is Song A from Song B?
  - Is Song A similar to Song B?
  - Is Song A more similar to Song B or to Song C?





Multidimensional scaling

- *MDS*: umbrella term for embedding from numerical dissimilarity.
- Input:
  - $\Delta \in \mathbb{R}_+^{n \times n}$ hollow and symmetric, squared "distances"
    target dimension d: $0 < d \leq n$
- Output:
  - $X \in \mathbb{R}^{n \times d}$



	A	B	P	S
A	0	2	13	1
B	2	0	5	9
P	13	5	0	5
S	1	9	5	0

Δ

Linear algebra review

Definition

A symmetric matrix A has a *spectral decomposition* $A = V\Lambda V^{\mathsf{T}}$.
The columns of V are the eigenvectors of A.
Λ is diagonal, eigenvalues λ_i of A.

Linear algebra review

Definition

A symmetric matrix A has a *spectral decomposition* $A = V\Lambda V^{\mathsf{T}}$.
The columns of V are the eigenvectors of A.
Λ is diagonal, eigenvalues λ_i of A.

Definition

A symmetric matrix A is *positive semi-definite* (PSD) if $\forall_i \lambda_i \geq 0$.
Notation: $A \succeq 0$

EDM and PSD

Theorem

Δ is a Euclidean distance matrix (EDM) if and only if $A = -\frac{1}{2} H_n \Delta H_n \succeq 0$, where $H_n = I_n - \frac{1}{n} \mathbf{1} \cdot \mathbf{1}^\mathsf{T}$.

- If X is a configuration that generates Δ, then $A = -\frac{1}{2} H_n \Delta H_n = X X^\mathsf{T}$.
- Recover X by spectral decomposition:

$$A = V \Lambda V^\mathsf{T} = \left(V \Lambda^{1/2} \right) \left(\Lambda^{1/2} V^\mathsf{T} \right) = X X^\mathsf{T}.$$

- If Δ is not Euclidean, project A onto PSD before factoring. (Zero out all $\lambda_i < 0$.)

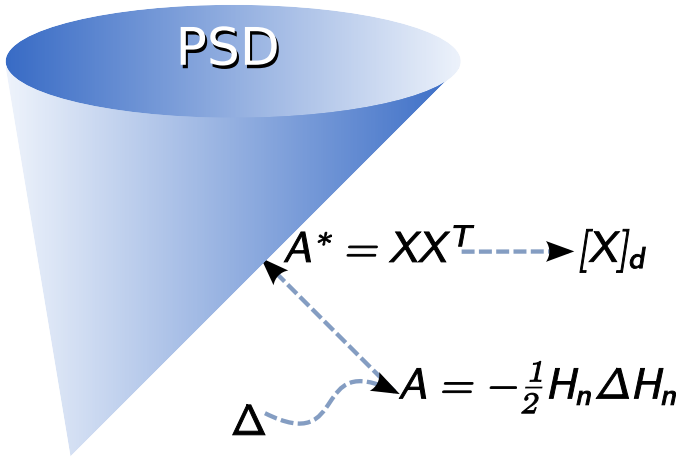
Classical MDS

Input

Δ: dissimilarity matrix
d: target dimension

Output

$X \in \mathbb{R}^{n \times d}$



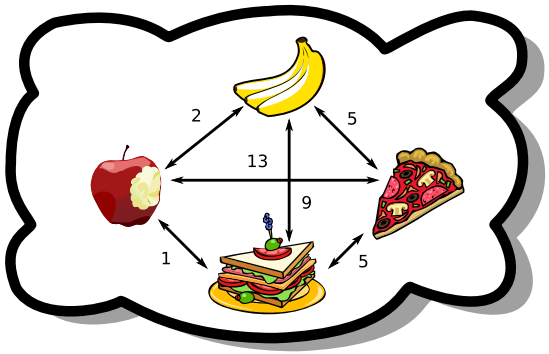
Algorithm

CMDS(Δ, d)

1. $A \leftarrow -\frac{1}{2} H_n \Delta H_n$.
2. *Spectral decomposition:* $A = V \Lambda V^{\mathsf{T}}$.
3. *Zero all* $\lambda_i < 0$.
4. *return the first* d *columns of* $V \Lambda^{1/2}$.

(Torgerson [Tor52] and Gower [Gow66])

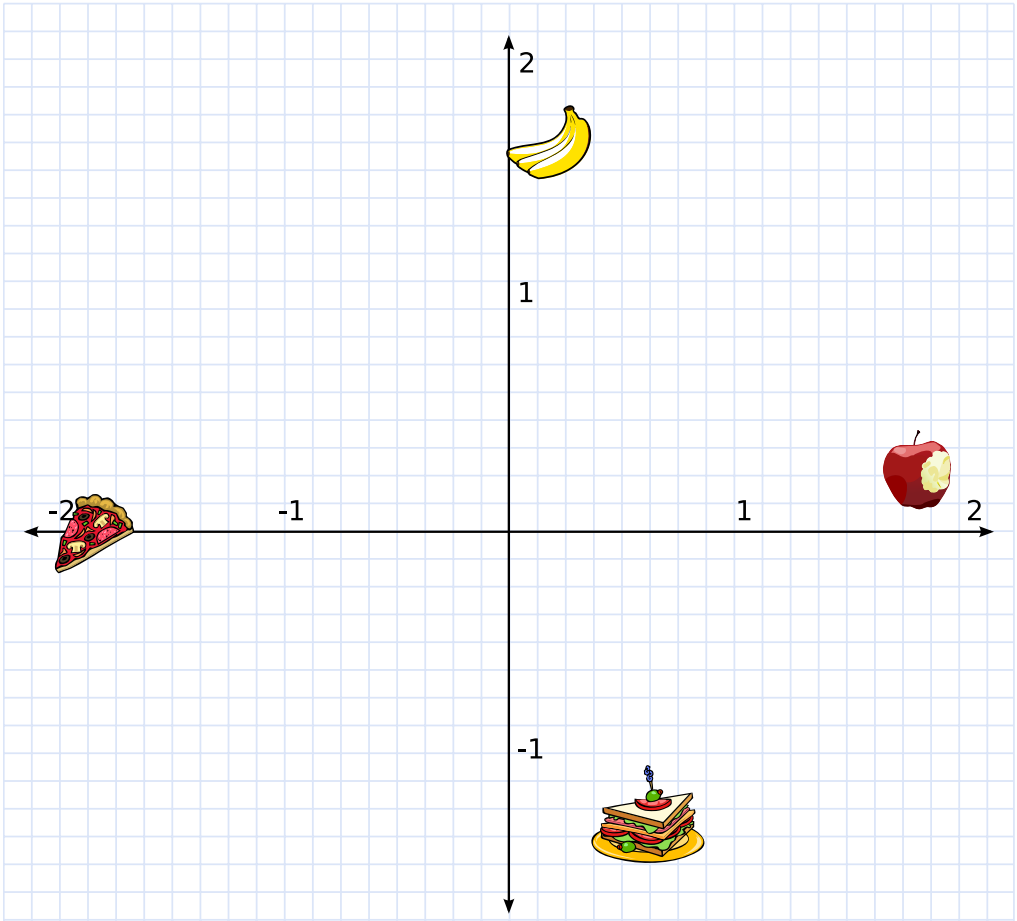
MDS example



	A	B	P	S
A	0	2	13	1
B	2	0	5	9
P	13	5	0	5
S	1	9	5	0

Δ

Output from Classical MDS ($d = 2$)



Limitations of MDS

- It's not clear why working with A is sensible.
  - Project Δ onto EDM and then run CMDS [GHHW90].
- Distances collapse when projecting X to $\mathbb{R}^d$.
  - Rank constraints are hard. We could fix d and settle for a locally optimal solution.
- MDS requires lots of quantitative data.
  - We can weight Δ and operate with incomplete constraints.
- What if quantitative measurements aren't reliable for the problem?
  - Non-metric MDS only preserves rank-order of distance.

Limitations of MDS

- It's not clear why working with A is sensible.
  - Project Δ onto EDM and then run CMDS [GHHW90].
- Distances collapse when projecting X to $\mathbb{R}^d$.
  - Rank constraints are hard. We could fix d and settle for a locally optimal solution.
- MDS requires lots of quantitative data.
  - We can weight Δ and operate with incomplete constraints.
- What if quantitative measurements aren't reliable for the problem?
  - Non-metric MDS only preserves rank-order of distance.

Limitations of MDS

- It's not clear why working with A is sensible.
  - Project Δ onto EDM and then run CMDS [GHHW90].
- Distances collapse when projecting X to $\mathbb{R}^d$.
  - Rank constraints are hard. We could fix d and settle for a locally optimal solution.
- MDS requires lots of quantitative data.
  - We can weight Δ and operate with incomplete constraints.
- What if quantitative measurements aren't reliable for the problem?
  - Non-metric MDS only preserves rank-order of distance.

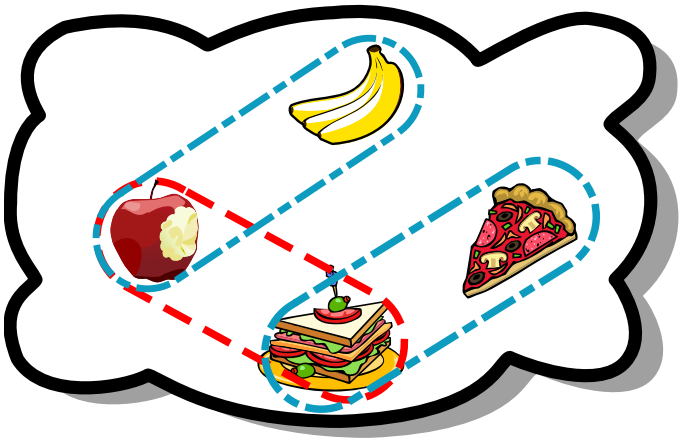
Limitations of MDS

- It's not clear why working with A is sensible.
  - Project Δ onto EDM and then run CMDS [GHHW90].
- Distances collapse when projecting X to $\mathbb{R}^d$.
  - Rank constraints are hard. We could fix d and settle for a locally optimal solution.
- MDS requires lots of quantitative data.
  - We can weight Δ and operate with incomplete constraints.
- What if quantitative measurements aren't reliable for the problem?
  - Non-metric MDS only preserves rank-order of distance.

Discrete proximity

- Exact numbers might be unreliable, but we can ask:

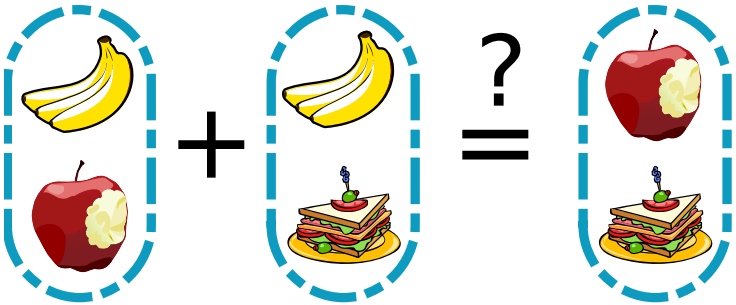
  "*Are* {**apple**, **banana**} similar *or* dissimilar*?*"

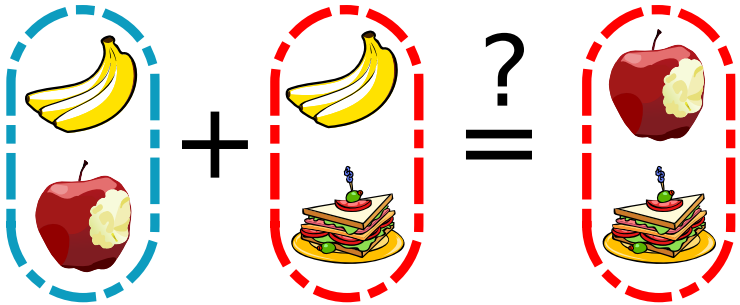


- Map *similar* pairs "close" and *dissimilar* pairs "far".
- "Closeness" must be defined...
  - ...minimal distance?
  - ...within a fixed radius?
  - ...k-nearest neighbors?

What is similarity?

- We have two competing notions:
  1. $\{x, y\}$ belong to the same *equivalence class*. *Similarity is transitive.*
  2. $\{x, y\}$ are neighbors. *Similarity may not be transitive.*
- We'll skip the equivalence class algorithms.
- We'll focus on more general *neighbor-preserving* algorithms.





Maximizing dissimilarity

- Xing et al.'s algorithm assumes the following [XNJR03]:
  - $\mathcal{X} \subset \mathbb{R}^D$.
  - The embedding is a linear transformation: $G \in \mathbb{R}^{D \times D}$.
- Idea:
  - *Bound the sum* of similar-pair distances.
  - *Maximize the sum* of dissimilar-pair distances.
  - Optimize $M = G^\mathsf{T} G \succeq 0$.

Notation: distance after transformation is

$$(Gx)^\mathsf{T}(Gx) = x^\mathsf{T} G^\mathsf{T} Gx = x^\mathsf{T} Mx = \|x\|_M^2.$$

Xing's metric learning algorithm

Input

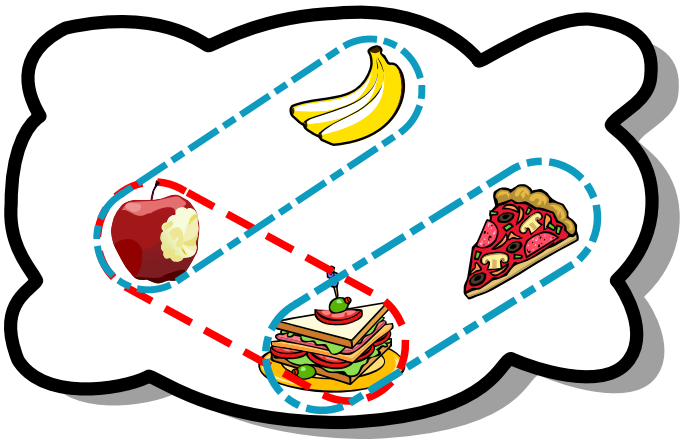
$\mathcal{X} \subset \mathbb{R}^D$

similarities $\mathcal{C}_+ \subseteq \mathcal{X}^2$

dissimilarities $\mathcal{C}_- \subseteq \mathcal{X}^2$

Output

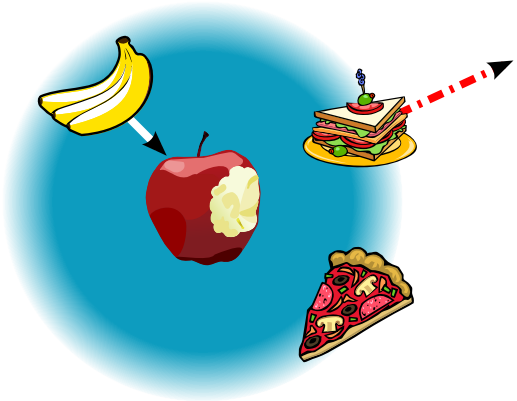
PSD matrix $M = G^\mathsf{T} G \in \mathbb{R}^{D \times D}$



Algorithm

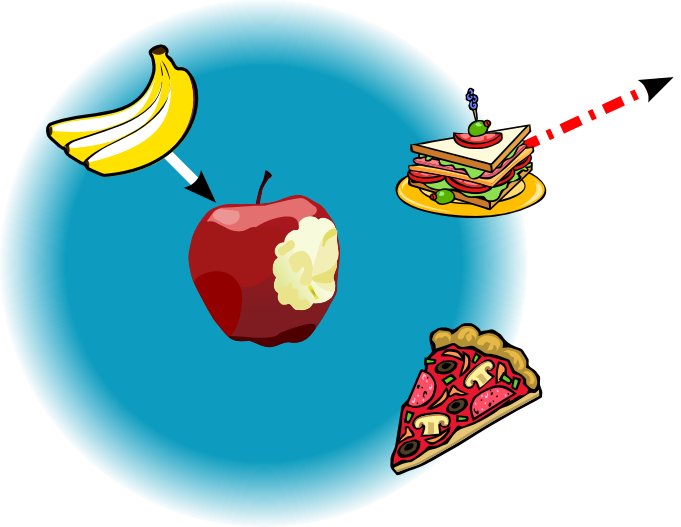
LearnMetric$(\mathcal{X}, \mathcal{C}_+, \mathcal{C}_-)$

$$\begin{aligned} \max_M \quad & \sum_{\mathcal{C}_-} \|x_i - x_j\|_M \\ \text{s.t.} \quad & \sum_{\mathcal{C}_+} \|x_i - x_j\|_M^2 \leq 1 \\ & M \succeq 0 \end{aligned}$$



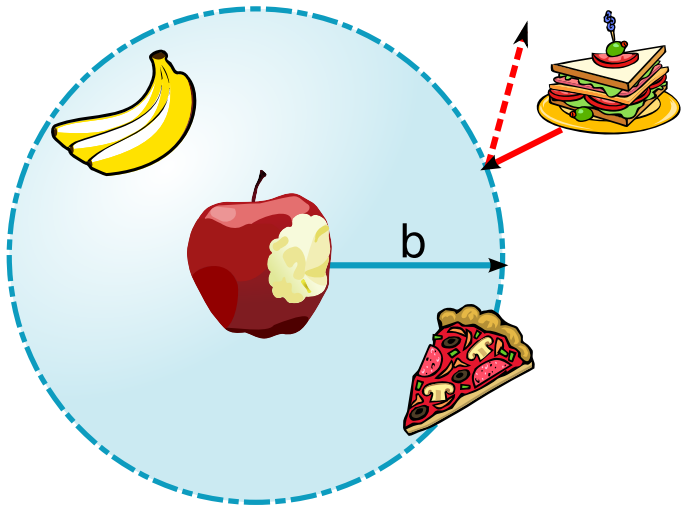
Limitations

- Requires feature descriptions: $\mathcal{X} \subset \mathbb{R}^D$.
- *Similar*-pair distances are only small on average.
- *We can't distinguish* similar *from* dissimilar!
- The similarity constraint is too rigid.

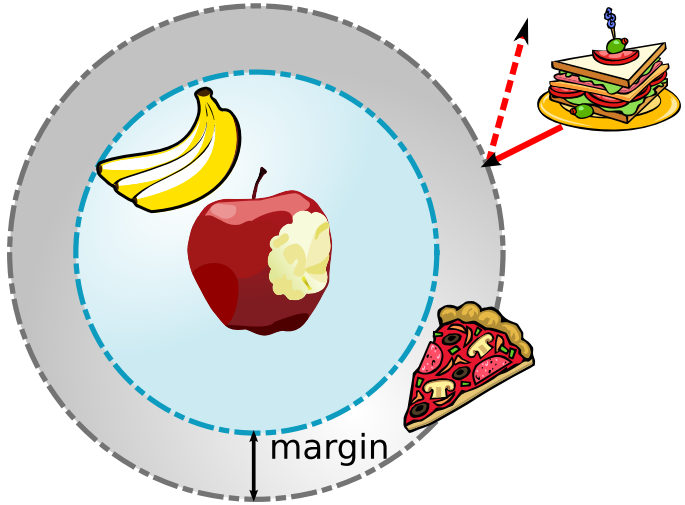


Margin methods

- Constrain each *similar* pair distance within a fixed radius.
- *Dissimilar* points cannot invade the neighborhood.
- This could take a few different forms. . .



Shalev-Shwartz et al. [SSSN04], Globerson and Roweis [GR07].



Weinberger et al. [WBS06]

A ball-constraint algorithm

- Globerson and Roweis's algorithm fits a ball around each x_i.
- All points *similar* to x_i have distance *at most* b_i.
- *Dissimilar* points have distance *at least* b_i.
- If $A = XX^{\mathsf{T}}$, then $\|X_i - X_j\|^2 = A_{ii} + A_{jj} - 2A_{ij}$.

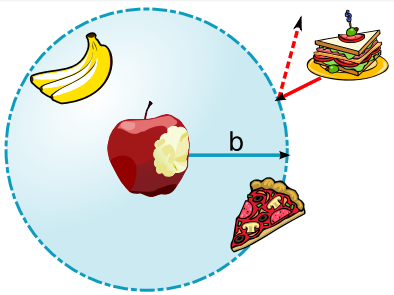
Pairwise semi-definite embedding

Input

A set $\mathcal{X}$
$\mathcal{C}_+ \subseteq \mathcal{X}^2$ ($s_{ij} = +1$)
$\mathcal{C}_- \subseteq \mathcal{X}^2$ ($s_{ij} = -1$)
target dimension d
slack tradeoff $\gamma > 0$

Output

$X \in \mathbb{R}^{n \times d}$



Algorithm

$PSDE(\mathcal{X}, \mathcal{C}_+, \mathcal{C}_-, d, \gamma)$

$$\min_{A,b,\xi} \sum_{\mathcal{C}_+,\mathcal{C}_-} s_{ij}\,(A_{ii} + A_{jj} - 2A_{ij}) + \gamma \sum \xi_{ij}$$

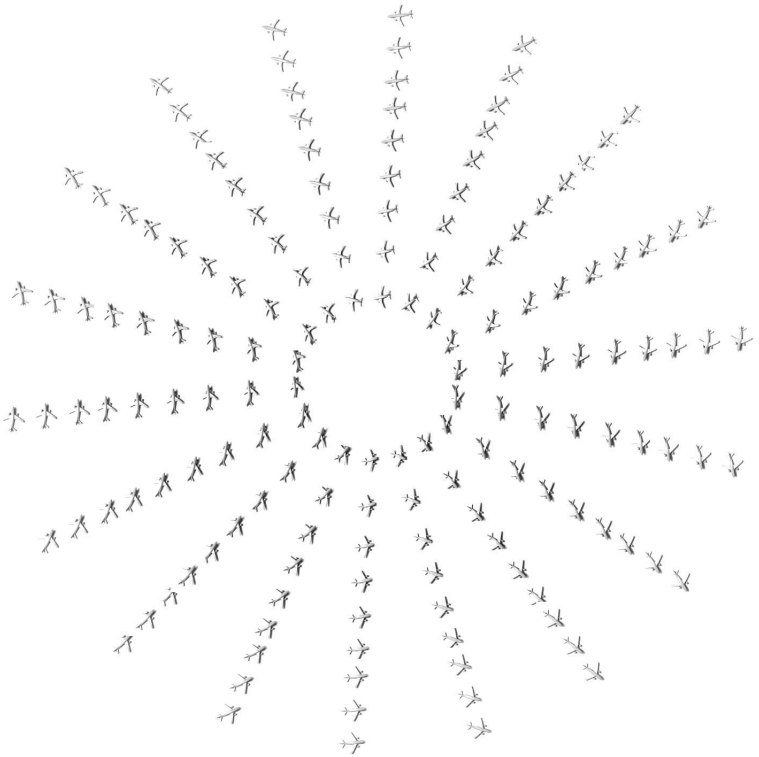
s. t.

$$\forall_{i,j} \qquad s_{ij}\,(A_{ii} + A_{jj} - 2A_{ij}) \leq s_{ij}\,b_i + \xi_{ij}$$

and feasibility constraints. . .

PSDE Results

NORB image set. Grid neighbors are similar, all other pairs are dissimilar.



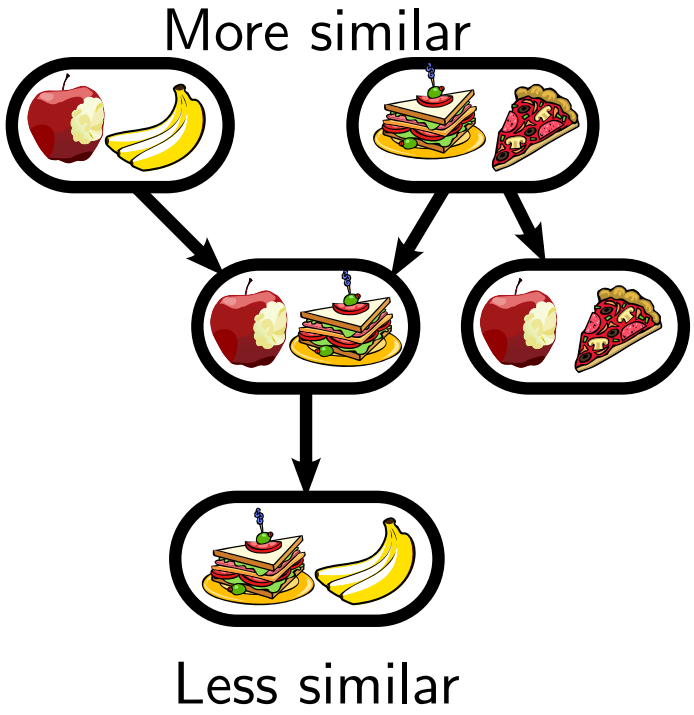
Embedding generated by PSDE: $(x\sqrt{z}, y\sqrt{z})$.

Partial Order

- Similarity might be too ambiguous or complex to define.
- Are apples and bananas similar?
  - In a set of food? Probably.
  - In a set of fruit? Maybe not.
- Ask an easier question:

  "Is **apple** *more like* **banana** *or* **sandwich***?"*

- We get a *partial order* over *pairs* from $\mathcal{X}$.



Embedding from partial order

- SVM-based embedding (Schultz and Joachims [SJ04]), Generalized Non-metric MDS (Agarwal et al. [AWC+07]).
- Given constraints

$$\mathcal{C} = \{(i, j, k, \ell) : \{x_i, x_j\} \text{ are } \textit{more similar} \text{ than } \{x_k, x_\ell\}\},$$

  embed $\mathcal{X}$ such that

$$\|x_i - x_j\| < \|x_k - x_\ell\|.$$

- Not enough structure to minimize/maximize distances...
  - force unit margin between distances,
  - regularize by sum of norms: $\min \mathrm{Tr}(A)$.

Generalized non-metric MDS

Input

$\mathcal{X}$

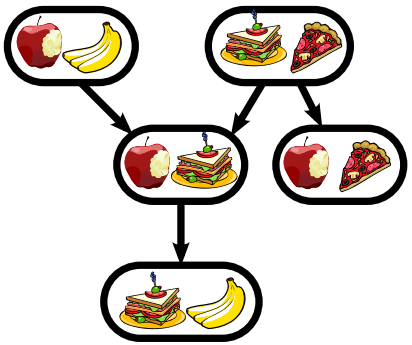
constraints $\mathcal{C}$

dimensionality d

slack tradeoff γ

Output

$X \in \mathbb{R}^{n \times d}$



Algorithm

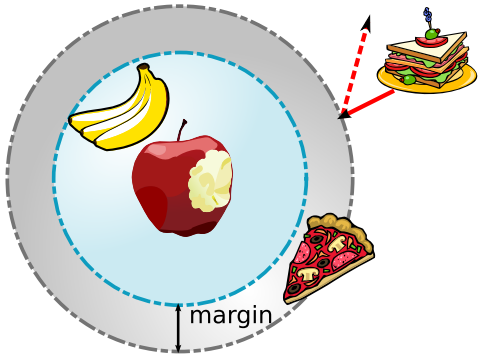
$GNMDS(\mathcal{X}, \mathcal{C}, d, \gamma)$

$$\min_{A,\xi} \quad \mathrm{Tr}(A) + \gamma \sum \xi_{ijk\ell}$$

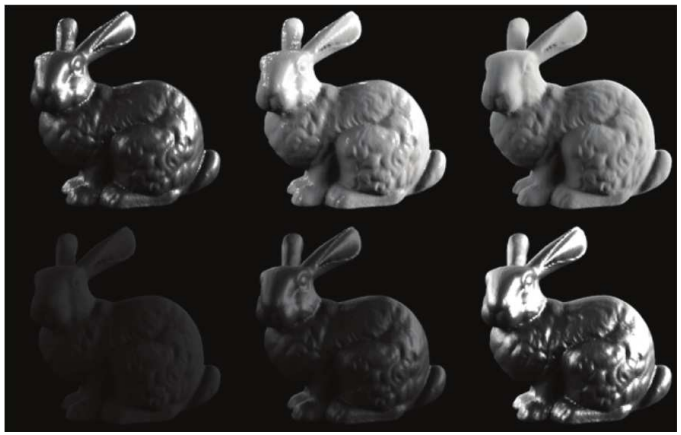
s. t.

$$\forall_{\{i,j,k,\ell\} \in \mathcal{C}} \quad A_{kk} + A_{\ell\ell} - 2A_{k\ell} \geq (A_{ii} + A_{jj} - 2A_{ij}) + (1 - \xi_{ijk\ell})$$

and feasibility constraints. . .



GNMDS Results



Varying textures on the Stanford Bunny.

The embedding obtained by GNMDS.

Generality

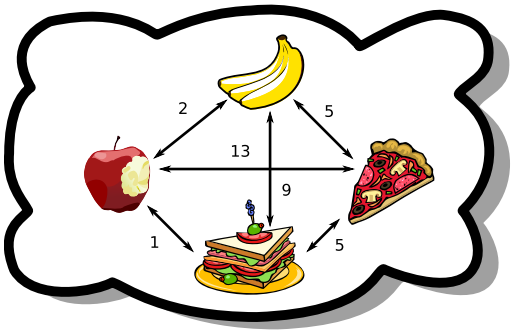
- Maybe more natural than discrete proximity for neighbor retrieval.
- For neighbor-search applications, partial order generalizes MDS and discrete proximity.
- Partial order constraints can always be satisfied in $\mathbb{R}^n$.

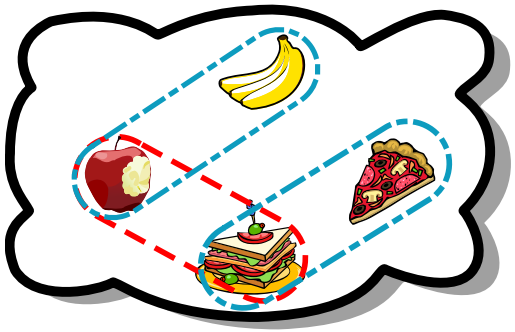
The bad news...

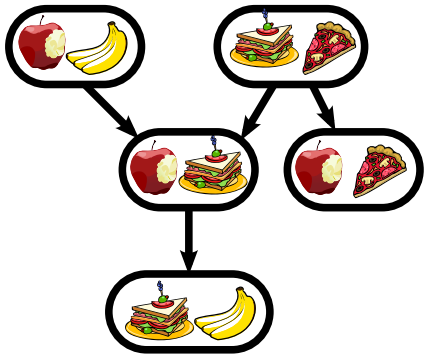
- Constraints are semantically weak, so we might need a lot of them.
- There are lots of constraints to be had: $\mathcal{O}(n^4)$.
- Compare to $\mathcal{O}(n^2)$ for MDS and discrete proximity.
- Good news: constraints tend toward DAG structure, but this hasn't been exploited yet.

Summary

- We've seen three frameworks for proximity-based embedding.
- Each framework has benefits and limitations.







Future work

- Embedding from multiple kernels
- Efficient constraint acquisition for partial order
- Asymmetric proximity
- Applications, applications, applications!

Thanks!

Questions?

Sameer Agarwal, Joshua Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie.
Generalized non-metric multi-dimensional scaling.
In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2007.

W. Glunt, T.L. Hayden, S. Hong, and J. Wells.
An alternating projection algorithm for computing the nearest euclidean distance matrix.
SIAM Journal on Matrix Analysis and Applications, 11(4):589–600, October 1990.

J.C. Gower.
Some distance properties of latent root and vector methods in multivariate analysis.
Biometrika, 53:325–338, 1966.

Amir Globerson and Sam Roweis.
Visualizing pairwise similarity via semidefinite embedding.

In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2007.

Matthew Schultz and Thorsten Joachims.
Learning a distance metric from relative comparisons.
In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

Shai Shalev-Shwartz, Yoram Singer, and Andrew Y. Ng.
Online and batch learning of pseudo-metrics.
In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

W.S. Torgerson.
Multidimensional scaling: 1. theory and method.
Psychometrika, 17:401–419, 1952.

Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul.

Distance metric learning for large margin nearest neighbor classification.
In Yair Weiss, Bernhard Schölkopf, and John Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 451–458, Cambridge, MA, 2006. MIT Press.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell.
Distance metric learning, with application to clustering with side-information.
In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2003. MIT Press.