UNIVERSITY OF GEORGIA

CSCI 8945 | Fall 2024

Advanced Representation Learning

Jin Sun, PhD
School of Computing

Lec 4: Manifolds, subspaces, and their learning

Outline

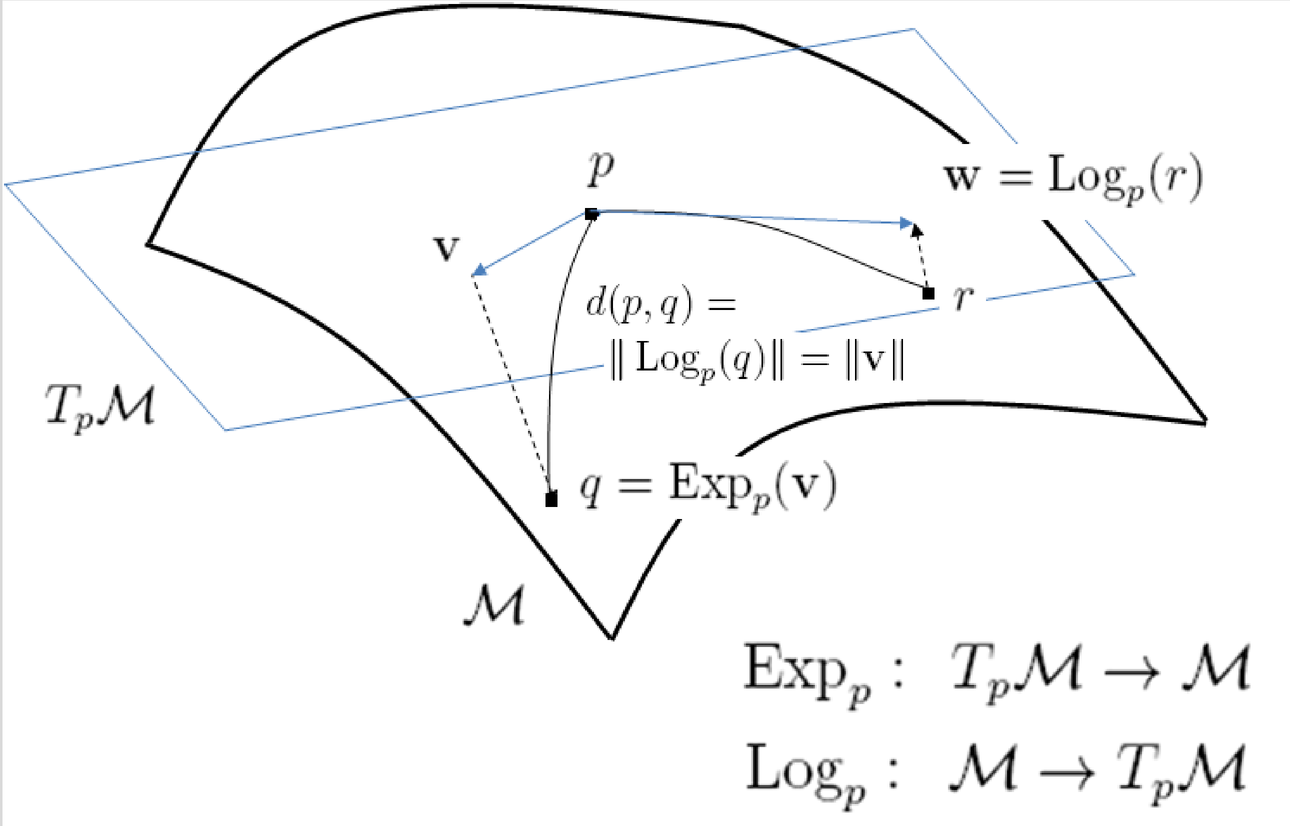
- Manifolds
  - Definition
  - Charts
  - Tangent space
  - Geodesics
- Manifolds learning
  - Isomap
  - Laplacian Eigenmap
  - t-SNE
- Example: image manifolds
- Sparse coding

Most contents will be on live demonstration.

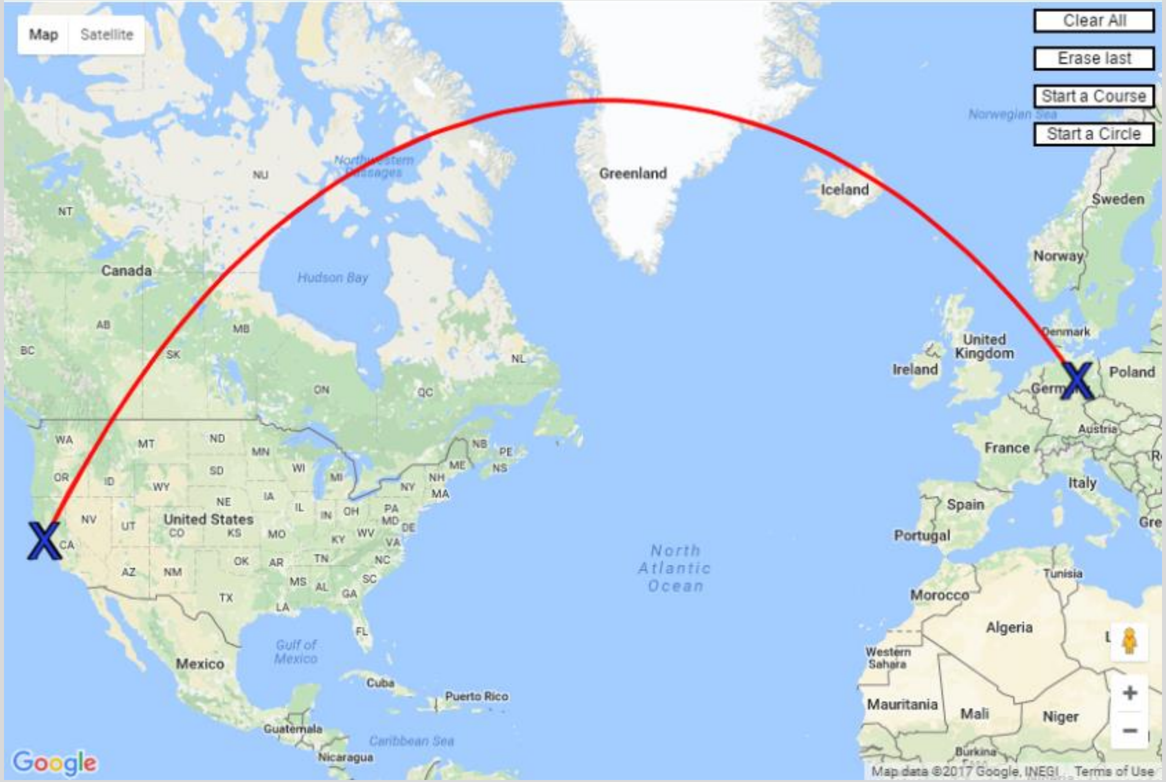
Manifolds - definition

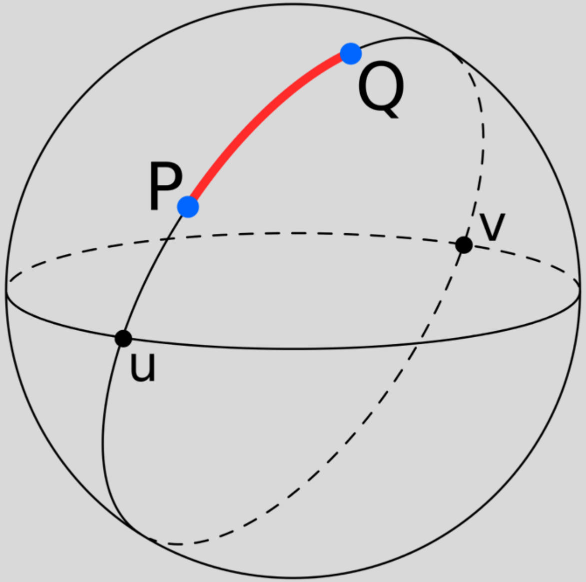
https://www.youtube.com/watch?v=0OauaSkYD44&list=PLD2r7XEOtm-AGjr3ynbljbx3oWHdus9Xb

Tangent space



Geodesic

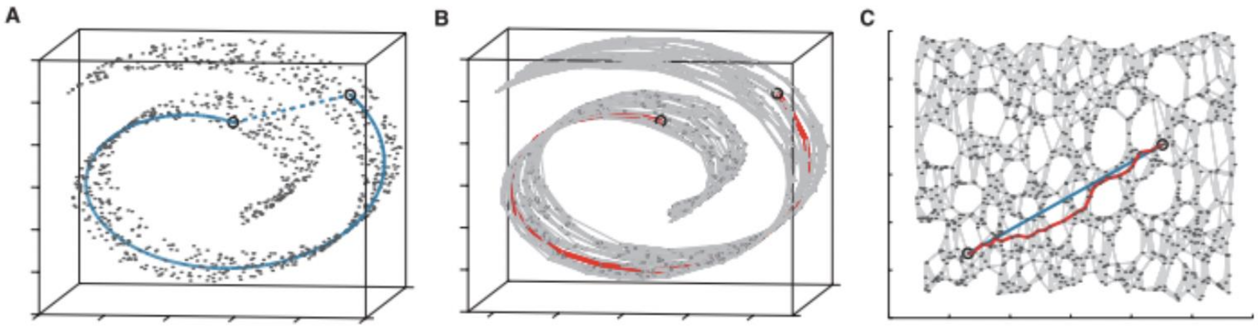

Map
Satellite
Clear All
Erase last
Start a Course
Start a Circle
Greenland
Iceland
Canada
Hudson Bay
United States
Mexico
North Atlantic Ocean
United Kingdom
Ireland
Norway
Sweden
Poland
France
Spain
Portugal
Italy
Morocco
Algeria
Mauritania
Mali
Niger
Cuba
Puerto Rico
Gulf of Mexico
Map data ©2017 Google, INEGI
Terms of Use


Q
P
v
u

ISOmap

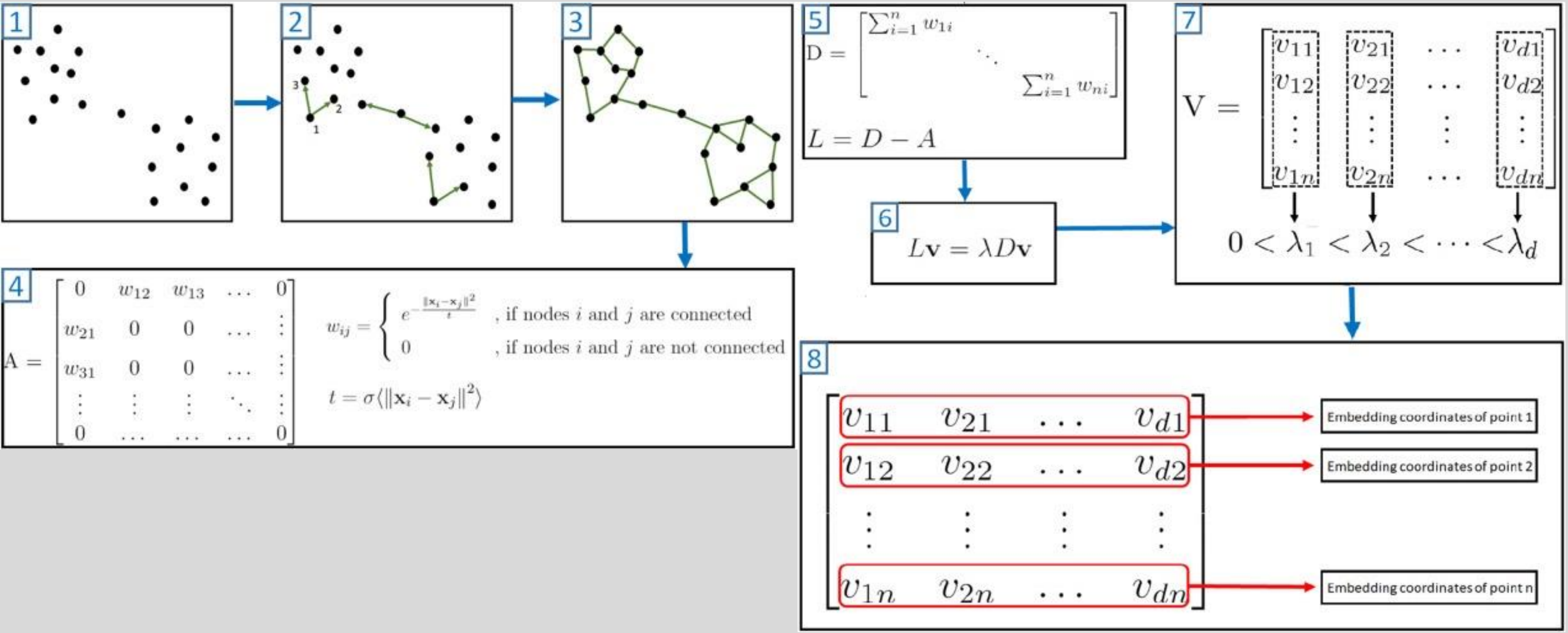
Strategy

In practical settings where we are only given a data set X sampled from an unknown manifold $\mathcal{M}$, we can approximate the true geodesic distances $d_{\mathcal{M}}(i,j)$ by the shortest-path distances $d_G(i,j)$ on a nearest-neighbor graph G built on the data set.



1. Build graph
2. Find shortest paths
3. Use MDS

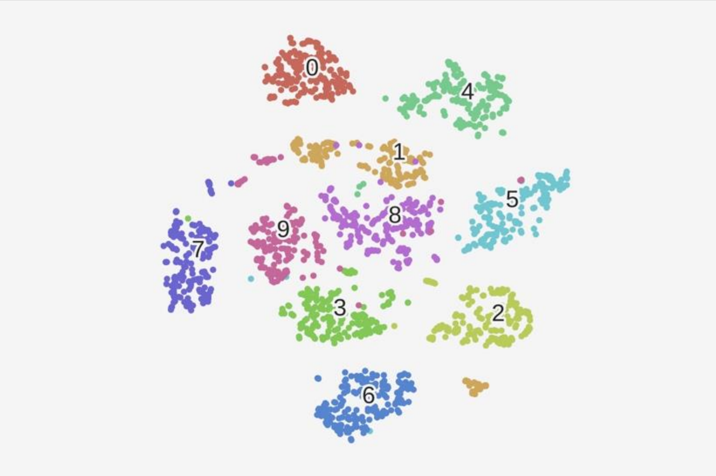
Laplacian Eigenmap



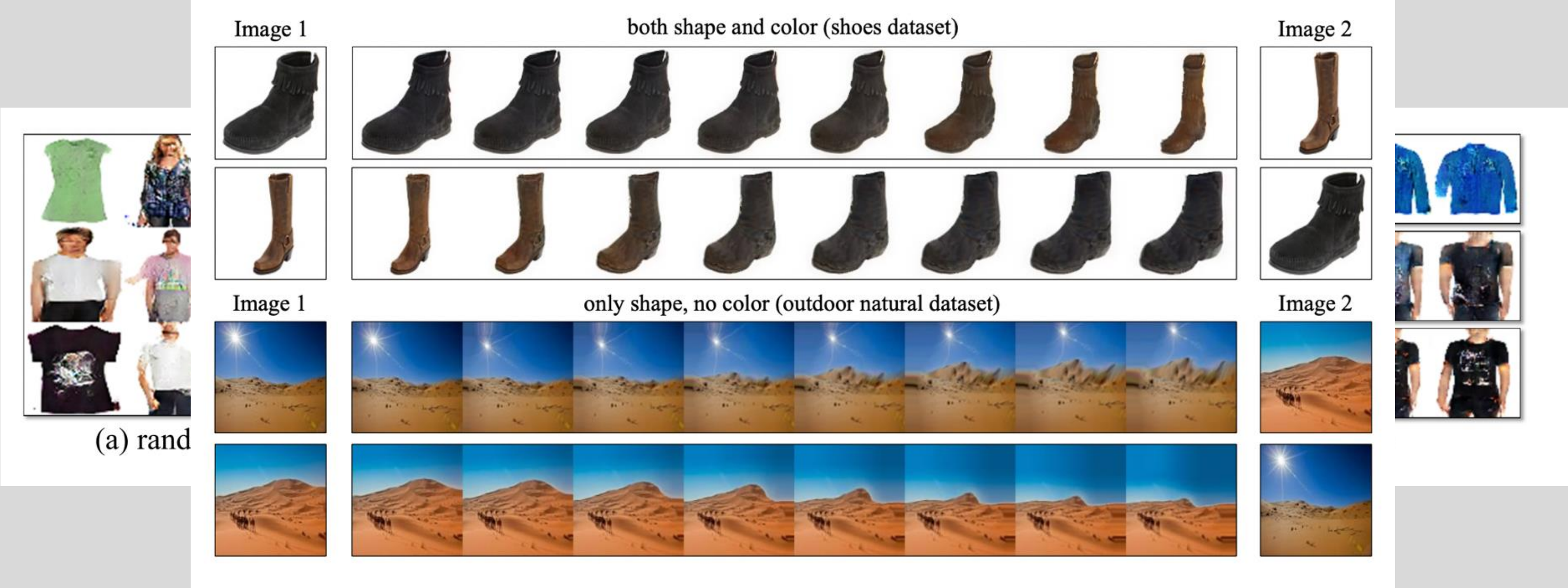
t-SNE

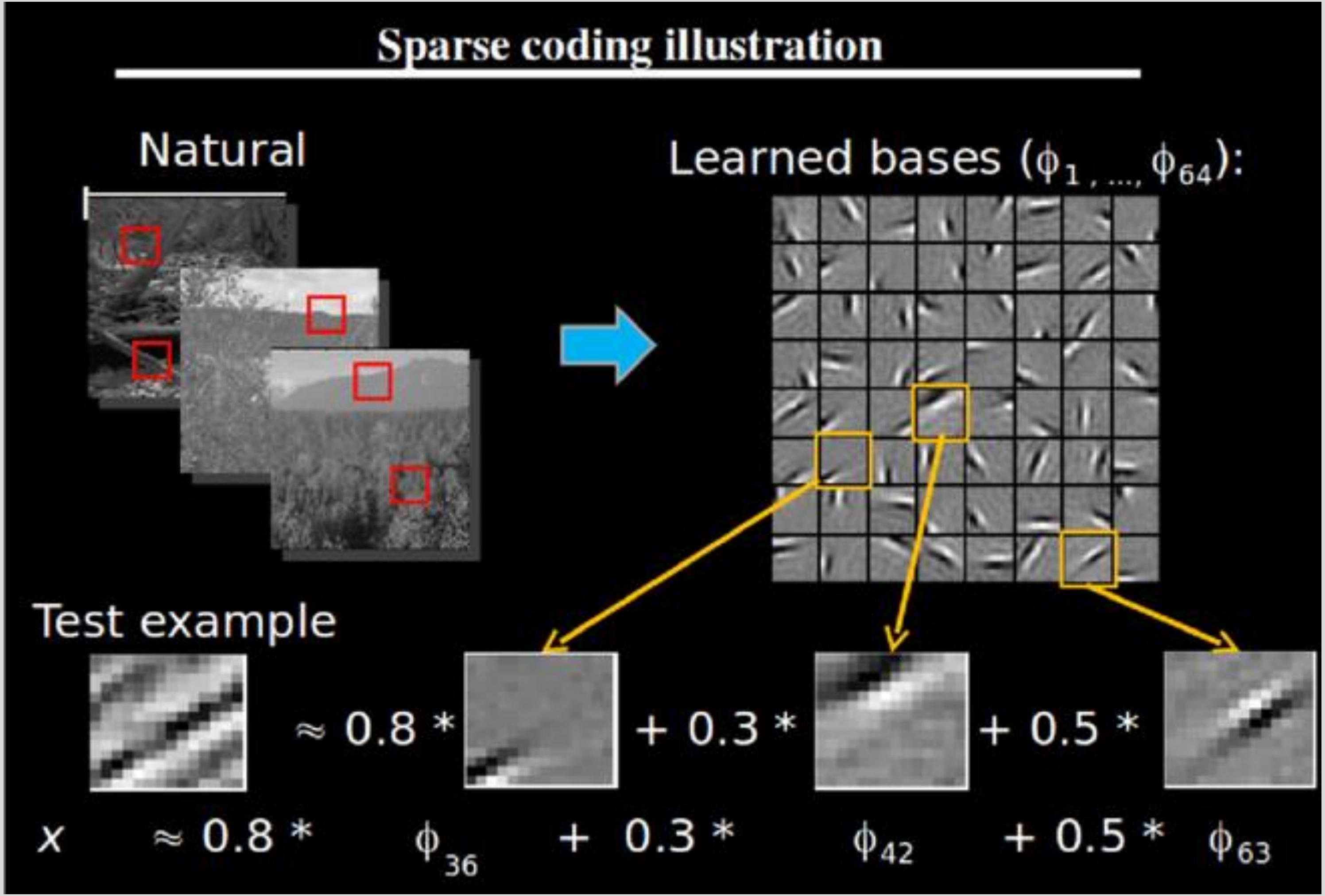
Nice introduction: https://www.oreilly.com/content/an-illustrated-introduction-to-the-t-sne-algorithm/

Good discussion:
https://distill.pub/2016/misread-tsne/



Generative Visual Manipulation on the Natural Image Manifold
Jun-Yan Zhu et al., ECCV 2016




Sparse coding illustration
Natural
Learned bases (ϕ1 , ..., ϕ64):
Test example
≈ 0.8 * + 0.3 * + 0.5 *
x ≈ 0.8 * ϕ36 + 0.3 * ϕ42 + 0.5 * ϕ63

Sparse Coding

Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently. The aim of sparse coding is to find a set of basis vectors ϕ_i such that we can represent an input vector $\mathbf{x}$ as a linear combination of these basis vectors:

$$\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$$

While techniques such as Principal Component Analysis (PCA) allow us to learn a complete set of basis vectors efficiently, we wish to learn an **over-complete** set of basis vectors to represent input vectors $\mathbf{x} \in \mathbb{R}^n$ (i.e. such that $k > n$). The advantage of having an over-complete basis is that our basis vectors are better able to capture structures and patterns inherent in the input data. However, with an over-complete basis, the coefficients a_i are no longer uniquely determined by the input vector $\mathbf{x}$. Therefore, in sparse coding, we introduce the additional criterion of **sparsity** to resolve the degeneracy introduced by over-completeness.

Here, we define sparsity as having few non-zero components or having few components not close to zero. The requirement that our coefficients a_i be sparse means that given a input vector, we would like as few of our coefficients to be far from zero as possible. The choice of sparsity as a desired characteristic of our representation of the input data can be motivated by the observation that most sensory data such as natural images may be described as the superposition of a small number of atomic elements such as surfaces or edges. Other justifications such as comparisons to the properties of the primary visual cortex have also been advanced.

We define the sparse coding cost function on a set of m input vectors as

$$\text{minimize}_{a_i^{(j)}, \phi_i} \sum_{j=1}^{m} \left\| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)})$$

where $S(.)$ is a sparsity cost function which penalizes a_i for being far from zero. We can interpret the first term of the sparse coding objective as a reconstruction term which tries to force the algorithm to provide a good representation of $\mathbf{x}$ and the second term as a sparsity penalty which forces our representation of $\mathbf{x}$ to be sparse. The constant λ is a scaling constant to determine the relative importance of these two contributions.

Although the most direct measure of sparsity is the "L_0" norm ($S(a_i) = \mathbf{1}(|a_i| > 0)$), it is non-differentiable and difficult to optimize in general. In practice, common choices for the sparsity cost $S(.)$ are the L_1 penalty $S(a_i) = |a_i|_1$ and the log penalty $S(a_i) = \log(1 + a_i^2)$.

src