Spectral Dimensionality Reduction via Learning Eigenfunctions

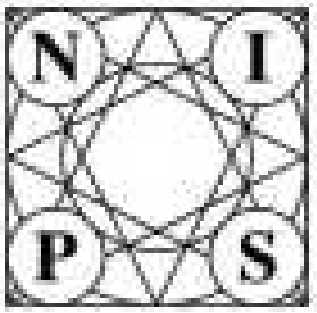
Yoshua Bengio

Laboratoire d'Informatique

des Systèmes Adaptatifs

http://www.iro.umontreal.ca/~lisa

Thanks to Pascal Vincent, Jean-François Paiement, Olivier Delalleau, Marie Ouimet, and Nicolas Le Roux.



Université de Montréal

Dimensionality Reduction

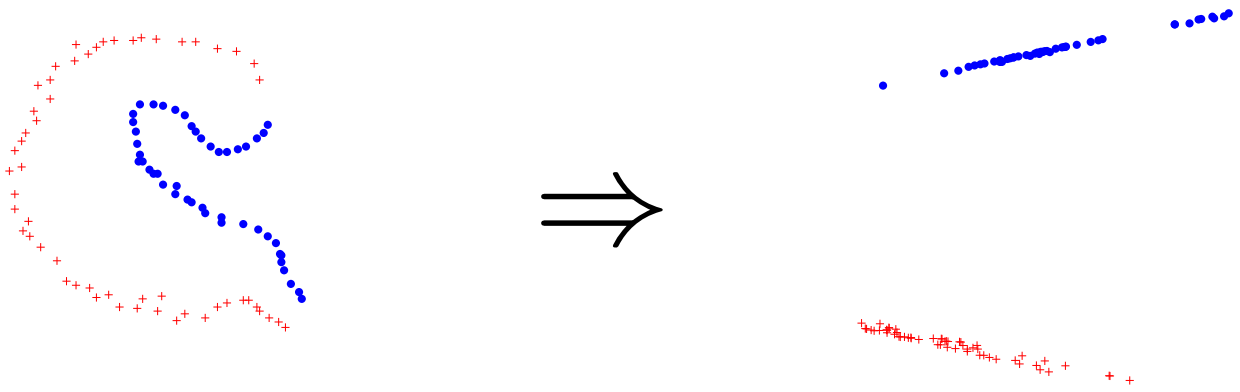
- For many distributions, it is plausible that most of the variations observed in the data can be explained by a small number of causal factors.
- If that is true there should exist a lower-dimensional coordinate system in which the data can be described with very little loss.
- Dimensionality reduction methods attempt to discover such representations.
- The reduced-dimension data can be fed in input for supervised learning.
- Unlabeled data can be used to discover the lower-dimensional representation.

Learning Modal Structures of the Distribution

Manifold learning and clustering

= learning where are the main high-density zones

Learning a transformation that reveals “clusters” and manifolds:



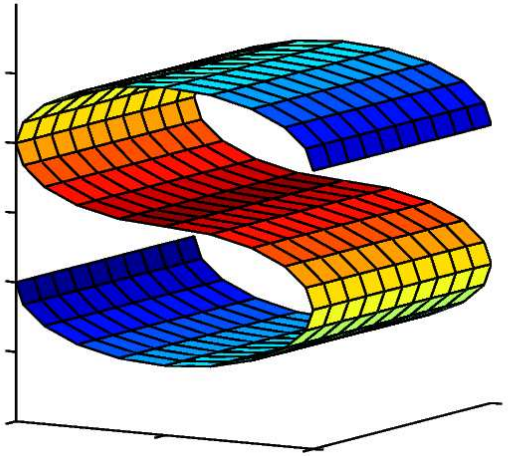
Cluster = zone of high density separated from other clusters by regions of low density

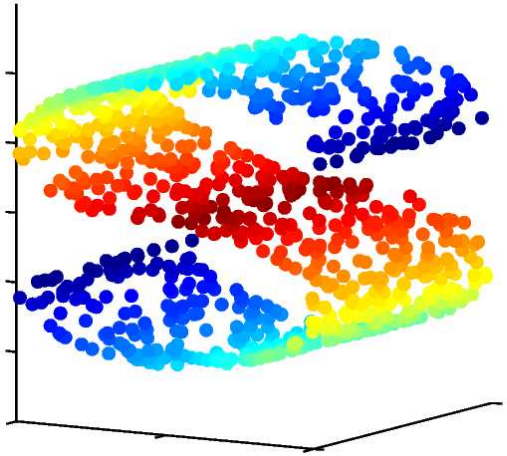
N.B. it is not always dimensionality reduction that we want, but rather “separating” the factors of variation. Here 2D → 2D.

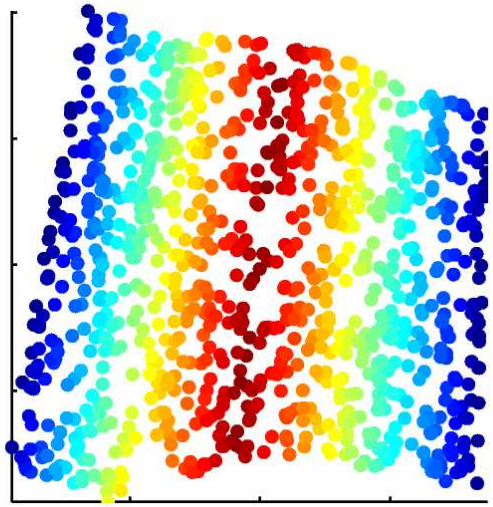
Local Linear Embedding (LLE)

Dimensionality reduction obtained with LLE:

(£g. *S. Roweis*)

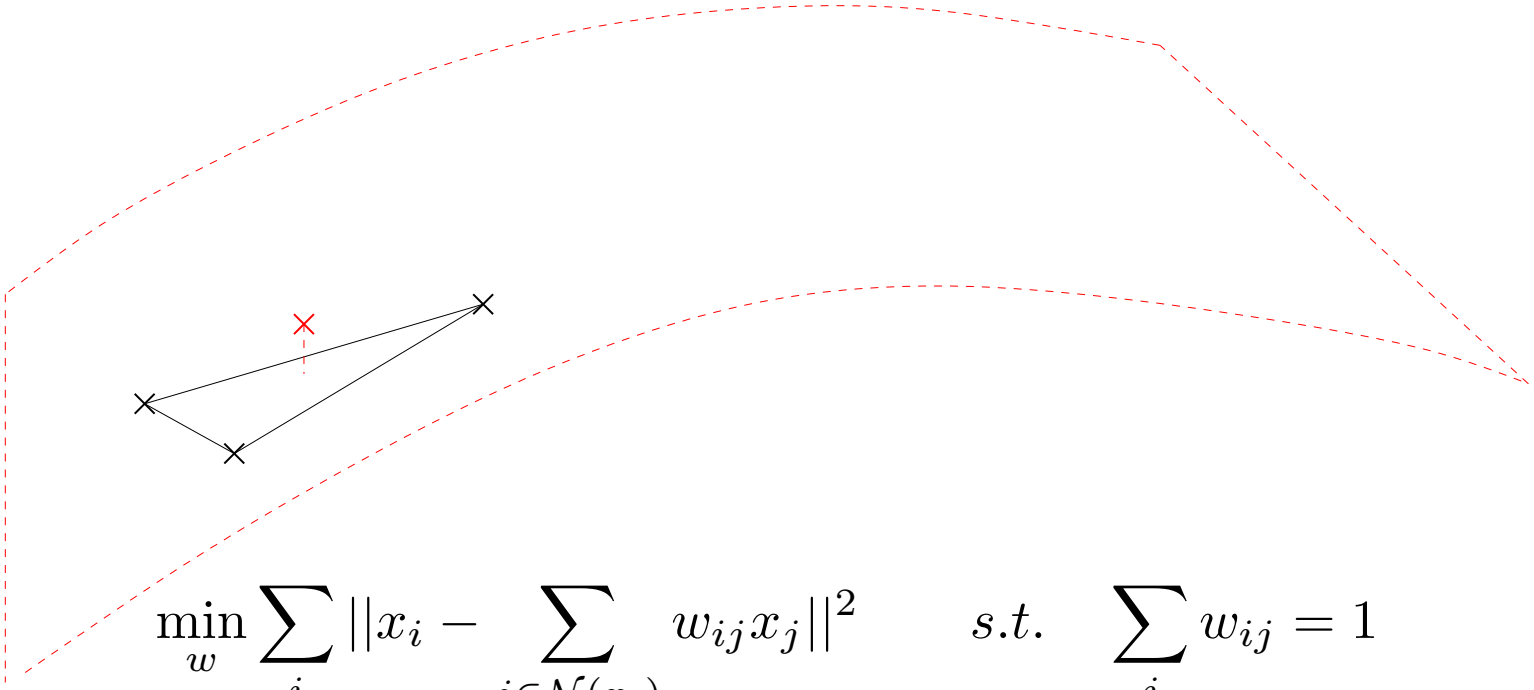






LLE: Local Af£ne Structure

The LLE algorithm estimates the local coordinates of each example in the basis of its nearest neighbors. Then looks for a low-dimensional coordinate system that has about the same expansion.



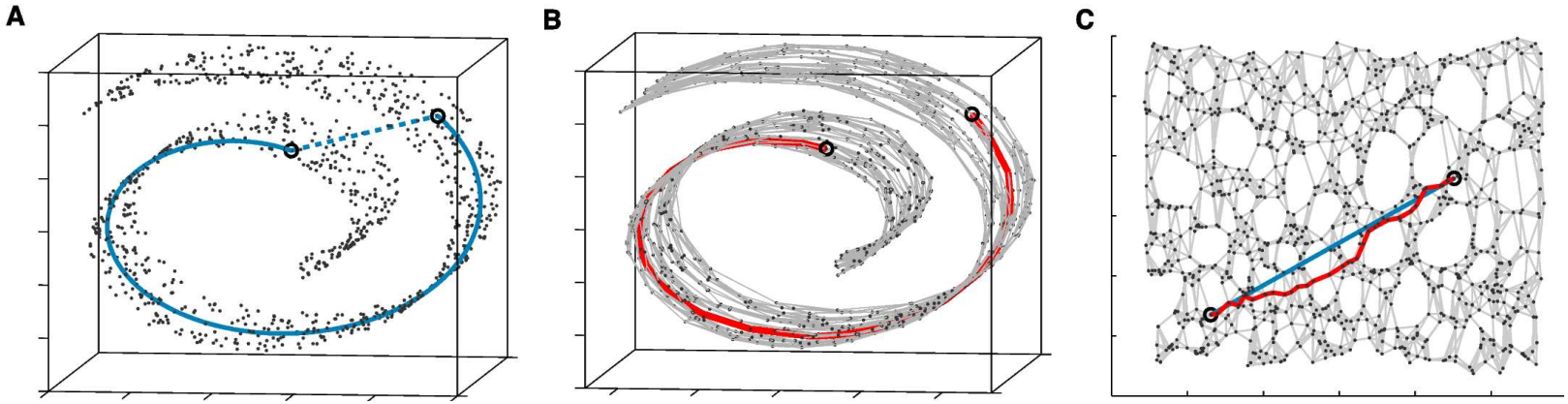
$$\min_w \sum_i ||x_i - \sum_{j \in \mathcal{N}(x_i)} w_{ij} x_j||^2 \qquad s.t. \quad \sum_j w_{ij} = 1$$

$$\min_y \sum_i ||y_i - \sum_{j \in \mathcal{N}(x_i)} w_{ij} y_j||^2 \qquad s.t. \quad y_{.k} \text{ orthonormal}$$

$\rightarrow$ solving an eigenproblem with sparse $n \times n$ matrix $(I - W)'(I - W)$

ISOMAP

Fig. from Tenenbaum et al 2000:



Isomap estimates the **geodesic distance** along the manifold using the shortest path in the nearest neighbors graph:

distance along path = sum of Euclidean distances between neighbors.

Then look for a low-dimensional representation that approximates those geodesic distances in the least square sense (MDS).

ISOMAP

Fig. from Tenenbaum et al 2000:

1. Build graph with 1 node/example, arcs for k-NN
2. for k-NN, weight(arc(x_i, x_j)) = $||x_i - x_j||^2$
3. new distance(x_i, x_j) = geodesic distance in graph [cost O(n^3)]
4. map distance matrix to dot product matrix: $-\frac{1}{2}(D_{ij} - \bar{D}_i - \bar{D}_j + \bar{\bar{D}})$
5. embedding y_i = i-th entry of principal eigenvectors

Spectral Clustering and Laplacian Eigenmaps

- Normalize kernel or Gram matrix divisively:

$$\tilde{K}(x,y) = \frac{K(x,y)}{\sqrt{E_x[K(x,y)]E_y[K(x,y)]}}$$

- map $x_i \rightarrow (\alpha_{1i}, \ldots, \alpha_{ki})$ where α_k is k-th e-vector of Gram matrix.

- principal e-vectors $\rightarrow$ reduced dim. data = Laplacian eigenmaps

(e.g. Belkin uses that for semi-supervised learning; see also justi£cation as a non-parametric regularizer)

- spectral clustering: perform clustering on the embedded points (e.g. after normalizing by dividing by their norm).

Fig. from *(Weiss, Ng, Jordan 2001)*

Spectral Embedding Algorithms

Many unsupervised learning algorithms, e.g.

Spectral clustering, LLE, Isomap, Multi-Dimensional Scaling, Laplacian eigenmaps

have this structure:

1. Start from n data points $D = \{x_1, \ldots, x_n\}$
2. Construct a $n \times n$ "neighborhood" matrix $\tilde{M}$
   (with corresponding [often **DATA-DEPENDENT**] kernel $\tilde{K}_D(x, y)$)
3. "Normalize" $\tilde{M}$, yielding M
   (**implicity** built with corresponding kernel $K_D(x, y)$)
4. Compute k largest (equivalently, smallest) e-values/e-vectors (ℓ_k, v_k)
5. Embedding of x_i = i-th elements of each e-vector v_k (possibly scaled by $\sqrt{\ell_k}$) NO EMBEDDING FOR TEST POINTS: **Generalization?**

Results: What they converge to

- What happens as the number of examples increases?

- **These algorithms converge towards learning eigen-functions of a linear operator K_p de£ned with a data-dependent kernel K and the true data density $p(x)$:** $(K_p g)(x) = \int K(x,y) g(y) p(y) dy$.

N.B. E-fns solve $K_p f_k = \lambda_k f_k$.

eigen-vectors $\longrightarrow$ eigen-functions

Empirical Linear Operator

We associate with each data-dependent K_n a linear operator G_n and with K_∞ a linear operator G, as follows:

$$G_n f = \frac{1}{n} \sum_{i=1}^{n} K_n(\cdot, x_i) f(x_i)$$

and

$$G_\infty f = \int K_\infty(\cdot, y) f(y) p(y) dy$$

so $G_n \to K_p$ (law large nb)

- Thm: the Nyström formula gives the eigenfunctions of G_n up to normalization.

- Normalization converges to 1 as $n \to \infty$, also by law of large nb.

- Thm: If K_n converges uniformly in prob. and if the e-fns $f_{k,n}$ of G_n converge uniformly in prob., then they converge to the corresponding e-fns of G_∞.

Results: What they minimize

- Problem with current algorithms: no notion of generalization error!

- New result: they min. training set avg of reconstruction loss

$$(K(x,y) - \sum_k \lambda_k f_k(x) f_k(y))^2,$$

i.e. £nd f_k e-fns of $K_{\hat{p}}$ ($\hat{p}$: empirical density).

$\Rightarrow$ **corresponding notion of generalization error: expected loss**.

$\Rightarrow$ SEMANTICS = **approximating / smoothing the notion of similarity given by the kernel**.

Generalizes the notion of learning a feature space, i.e. kernel PCA:

$$K(x,y) \approx g(x).g(y)$$

to the case of negative eigenvalues (which may occur!)

Results: Extension to new Examples

- Problem with current algorithms: only the low-dim coordinates of training examples can be computed!

- Nyström formula: Out-of-sample extensions can be de£ned:

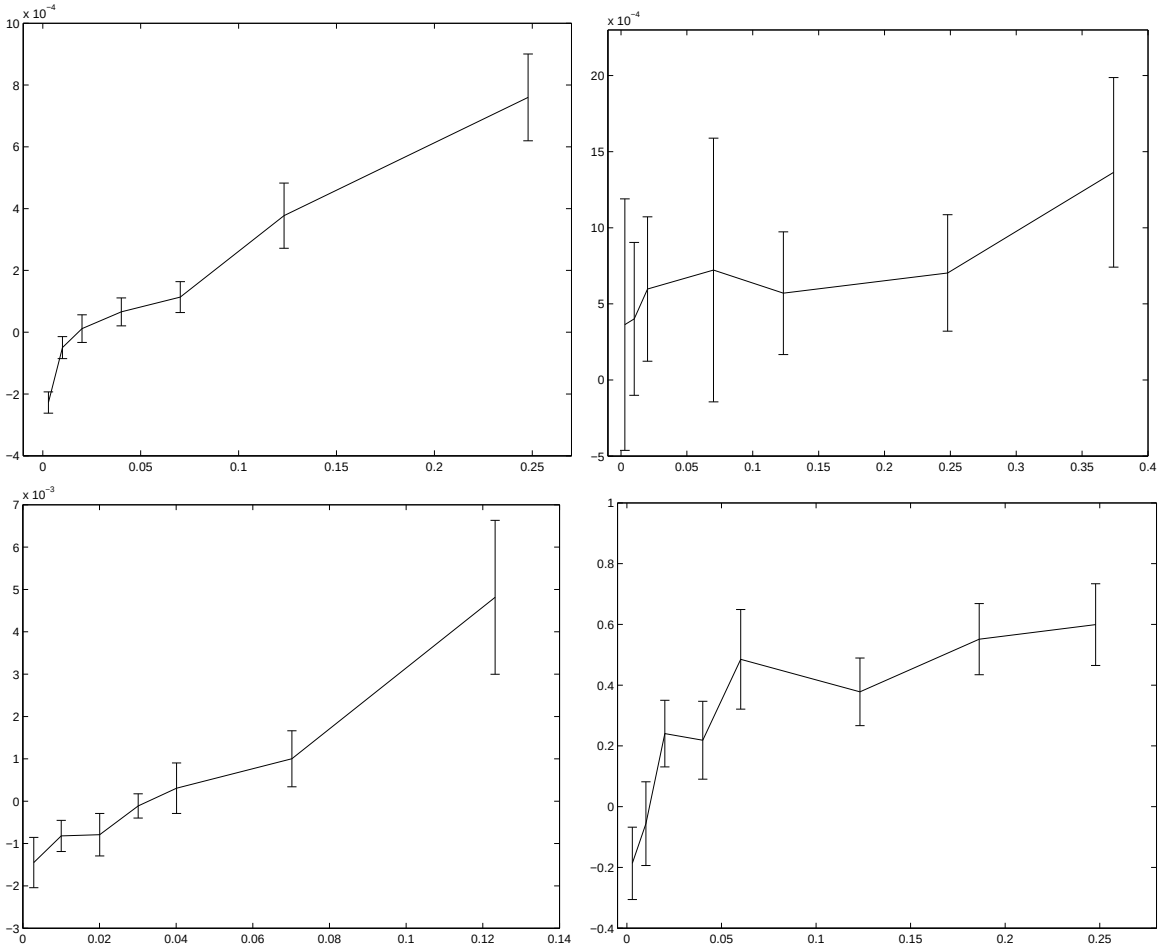
(which match the kernel PCA projection in pos. semi-de£nite case)

$$f_k(x) = \frac{1}{\lambda_k} \sum_{i=1}^{n} v_{ik} K(x, x_i)$$

to obtain embedding $f_k(x)$ or $\sqrt{\lambda_k} f_k(x)$ for new point x.

- New theoretical results *apply to kernels not necessarily positive semi-de£nite (e.g. Isomap)*, and give a simple justi£cation based on the law of large numbers.

Out-of-sample Error ≈ Training Set Sensitivity



Training set variability minus out-of-sample error, wrt fraction of training set substituted. Top left: MDS. Top right: spectral clustering or Laplacian eigenmaps. Bottom left: Isomap. Bottom right: LLE. Error bars are 95% con£dence intervals.

Equivalent Kernels for Generalizing the Gram Matrix

With $E[]$ averages over D (not including test point x):

- for spectral clustering and Laplacian eigenmaps:

$$K(a,b) = \frac{1}{n} \frac{\tilde{K}(a,b)}{\sqrt{E_y[\tilde{K}(a,y)] E_{y'}[\tilde{K}(b,y')]}}$$

- for MDS and Isomap:

$$K(a,b) = -\frac{1}{2}(d^2(a,b) - E_y[d^2(y,b)] - E_{y'}[d^2(a,y')] + E_{y,y'}[d^2(y,y')])$$

d is geodesic distance for Isomap: the test point x is not used to shorten the distance between training points.

Corollary

The out-of-sample formula for Isomap is equal to the **Landmark Isomap** *formula for the above equivalent kernel.*

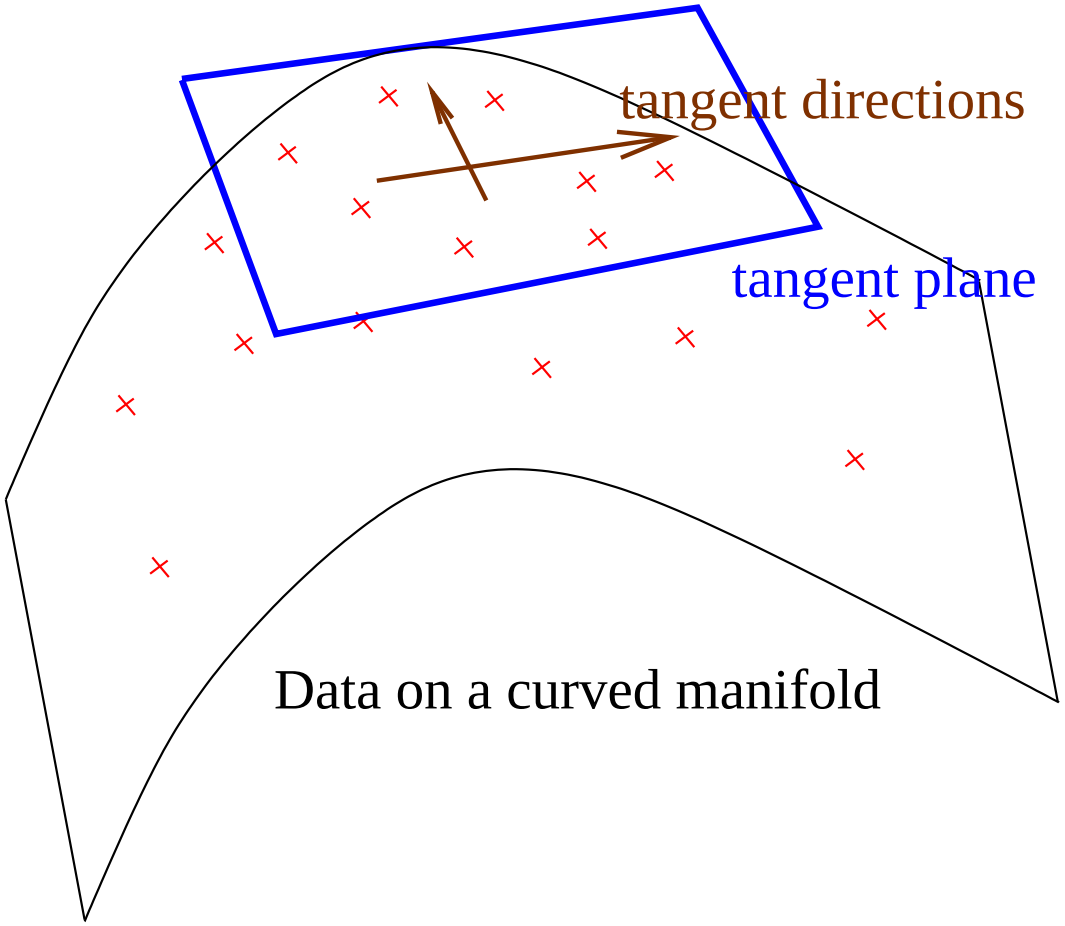
Algorithms with Better Generalization

- a kernel can be de£ned for LLE and Isomap. Experiments on **LLE, Isomap, spectral clustering, and Laplacian eigenmaps**, show that the resulting out-of-sample extensions work well: ***difference in embedding when test point is included or not in training set is comparable to the embedding pertubation from replacing a few examples from the training set.***

- **Generalization can be improved by replacing the empirical density $\hat{p}$ by a smoother one $\tilde{p}$** (a non-parametric density estimator). We used different sampling approaches and showed that statistically signi£cant improvements can be obtained on real data.

Challenge: Curved Manifolds

Current manifold learning algorithms cannot handle highly curved manifolds because they are based on locally linear approximations that require enough data locally to characterize the principal tangent directions of the manifold.



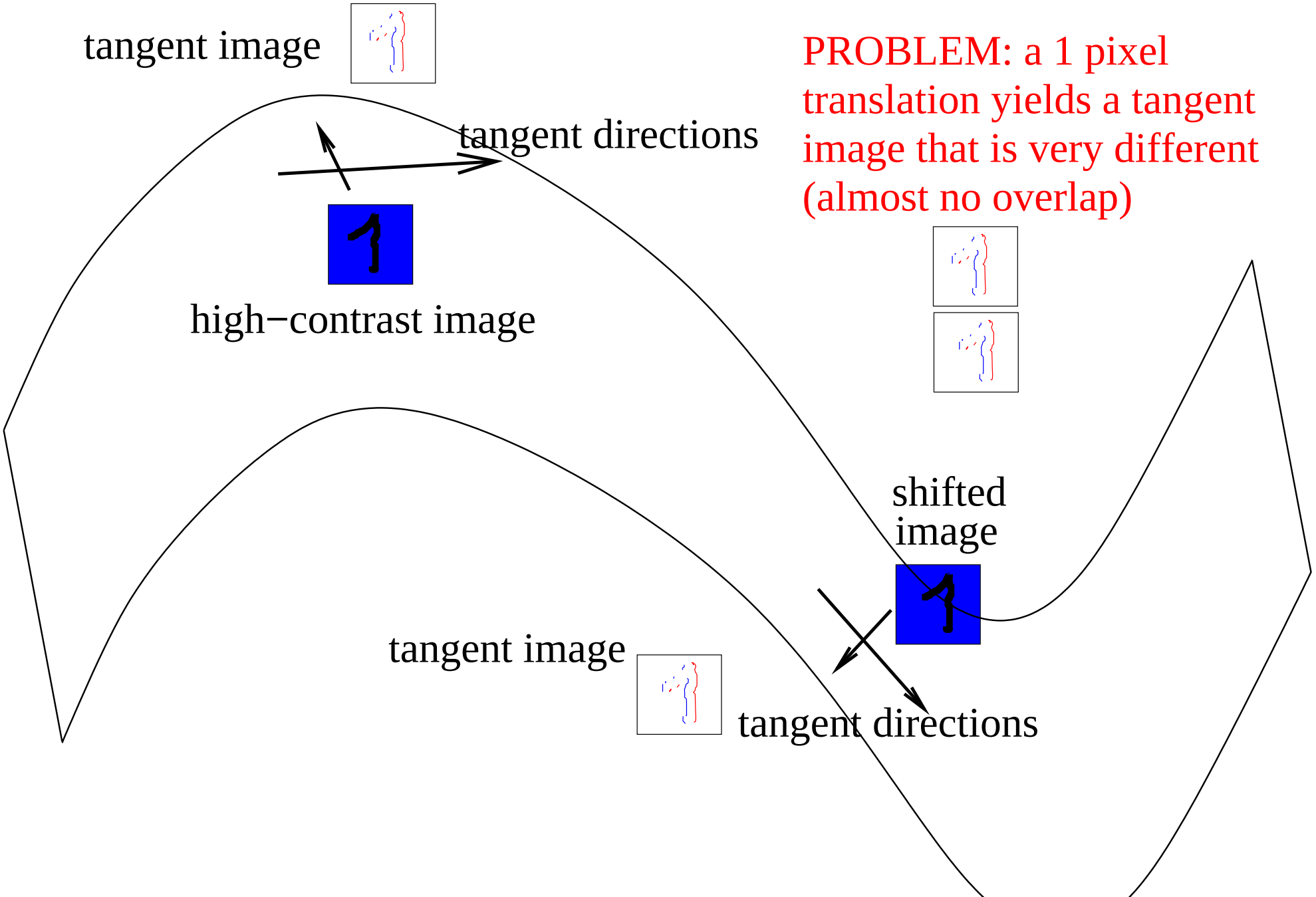
Other Local Manifold Learning Algorithms

Other examples of local manifold learning algorithms which would fail in the presence of highly curved manifolds:

- Mixture of factor analyzers
- Manifold Parzen windows (Vincent & Bengio 2002)

Approximate the density locally by a pancake, specifying only a few "interesting directions", but still locally linear, requires enough data locally to discover those directions and their relative variance.

Highly Curved Manifolds



Myopic vs Far-Reaching Learning Algorithms

- Most current algorithms are myopic because they must rely on highly local data to characterize the density.

- We should develop algorithms that allow one to generalize far from the training set, for example sharing information about global parameters that describe the structure of the manifold.

- In fact it is possible to parametrize the geometric operations on images as well as many other manifolds through **Lie Group** operations (e.g. a global single matrix characterizes horizontal translation).

Approximation of Lie Group Operators

• Consider images obtained by applying geometric operators to the view of an object (translations, rotations, scaling, etc...). Long-range deformation y of an original image x:

$$y = e^{\sum_i \alpha_i G_i} x$$

x, y are pixel vectors, the G_i are operator matrices (one per transformation), α_i = how much G_i to apply.

• (Rao & Ruderman 1999, "Learning Lie Groups for Invariant Visual Perception") show that one can learn G for translation of 1D images using pairs of images with a small translation and the approximation

$$e^{\alpha G} \approx (1 + \alpha G)$$

$\rightarrow$ quadratic optimization of α's (separately for each image pair) given G or G given the α's.

Conclusions

- High-dimensional data are both interesting and important to understand what it means to really generalize.
- Unsupervised learning: capturing the dependencies between a large number of variables, e.g. with manifold learning.
- Amazing progress in the last few years: non-linear manifolds can be learned, with easy to optimize convex criteria.
- We have uni£ed a large family of unsupervised learning algorithms and extended them to produce coordinates for new examples.
- They turn out to be methods to learn internal representations of the data that are faithful to a prior notion of similarity between objects.
- Great challenges ahead: how to deal with highly curved manifolds, i.e. how to generalize far from the training examples.
- Manifolds can be parametrized in richer ways, using parameters that have global extent, allowing to generalize to really fresh examples.