Nonlinear Embeddings

-Statistical Machine Learning-

Lecturer: Darren Homrighausen, PhD

Lower dimesional (metric) embeddings

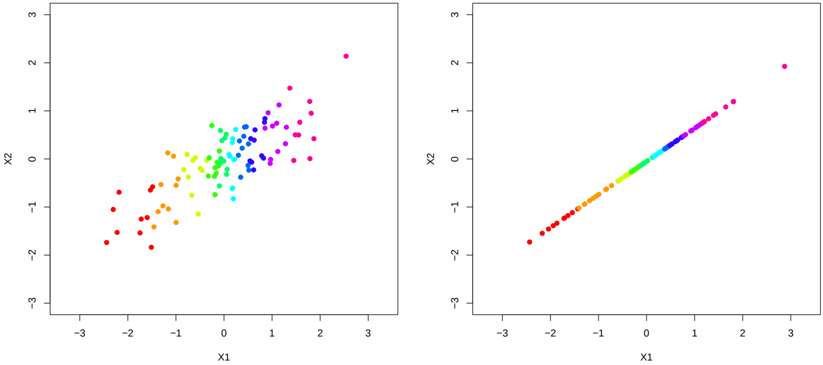
Spectral connectivity analysis (SCA) is a general process for finding lower dimensional structure in the data

It can be...

- Linear or nonlinear
- Used for dimension reduction or feature creation
- PCA, PLS, Fisher discriminant analysis, Locally linear embeddings, Hessian eigenmaps, Laplacian eigenmaps
- Useful as an input to classification, clustering, and regression approaches

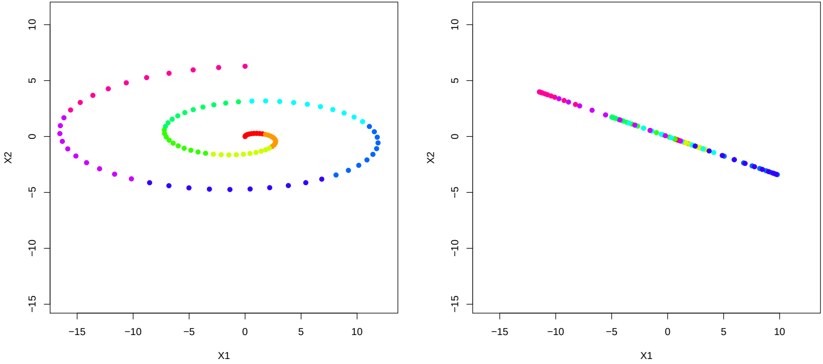
When PCA Works Well

PCA can do effective dimension reduction (that is, explain most of the data with $m < p$ components) as long as the data can be efficiently represented as 'lines' (or planes, or hyperplanes). So, in two dimensions:



When PCA doesn't work well

What about other data structures? Again in two dimensions



Here, we have failed miserably.

Explanation

- PCA wants to minimize distances (equivalently maximize variance). This means it slices through the data at the meatiest point, and then the next one, and so on. If the data are 'curved' this is going to induce artifacts.
- PCA also looks at things as being close if they are near each other in a Euclidean sense

  [this is essentially all correlation is].
- On the spiral, our intuition says that things are 'close' only if the distance is constrained to go along the curve. In other words, purple and blue are close, blue and red are not.

PCA and covariance

PCA: Find the directions of greatest variance. This doesn't on its face seem like it maintains correlations, but observe:

$$var(aX_1 + bX_2) = a^2 Var(X_1) + b^2 Var(X_2) + 2abCov(X_1, X_2)$$

If we standardize the matrix, then this reduces to

$$var(aX_1 + bX_2) = a^2 + b^2 + 2abCov(X_1, X_2)$$

This gets maximized over $a^2 + b^2 = 1$.

- If $Cov(X_1, X_2) \approx 0$, then this gets maximized by any $a^2 + b^2 = 1$ (it doesn't matter)
- If $Cov(X_1, X_2) \approx 1$, then this gets maximized by setting $a = b = 1/\sqrt{2}$

So, in either case, we are really maintaining correlations

Graphical example of the phenomenon

```
library(mvtnorm)
sigma   = matrix(c(1,sig,sig,1),nrow=2)
nsweep  = 1000
outcome = matrix(0,nrow=nsweep,ncol=2)
for(sweep in 1:nsweep){
x               = rmvnorm(200,c(0,0),sigma)
out.pca         = prcomp(x,center=T,scale=F)
outcome[sweep,] = out.pca$rotation[,1]
}
plot(outcome,xlab='PC1',ylab='PC2')
```

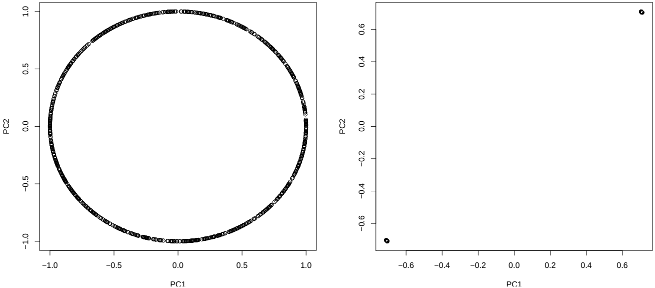


Figure: Left: sig = 0. Right: sig = .999

Nonlinear embeddings

Kernel PCA

Classical PCA comes from $\tilde{\mathbb{X}} = \mathbb{X} - M\mathbb{X} = UDV^\top$, where $M = 11^\top/n$ and $1 = (1, 1, \ldots, 1)^\top$

However, we can just as easily get it from the outer product

$$\tilde{K} = \tilde{\mathbb{X}}\tilde{\mathbb{X}}^\top = (I - M)\mathbb{X}\mathbb{X}^\top(I - M) = UD^2U^\top$$

The intuition behind KPCA is that $\tilde{K}$ is a (trivial) expansion into a kernel space, where

$$\tilde{K}_{i,i'} = k(\tilde{X}_i, \tilde{X}_{i'}) = \langle \tilde{X}_i, \tilde{X}_{i'} \rangle$$

Remember: Anytime we see an inner product, we can kernelize it

Kernel PCA

Following this intuition, the approach is simple:

1. Specify a kernel k
   (e.g. $k(x, x') = \exp\{-\gamma^{-1} ||x - x'||_2^2\}$)
2. Form $K_{i,i'} = k(X_i, X_{i'})$
3. Standardize and get eigenvector decomposition

$$\tilde{K} = (I - M)K(I - M) = UD^2U^\top$$

This implicitly finds the inner product:

$$k(X_i, X_{i'}) = \langle \phi(X_i), \phi(X_{i'}) \rangle$$

However, we need only specify the kernel

Kernel PCA

To get the first 'PC', we are solving for the function $\hat{g}_1$:

$$\max_{g \in \mathcal{H}_k} \mathbb{V} g(X) \text{ subject to } ||g||_{\mathcal{H}_k} = 1$$

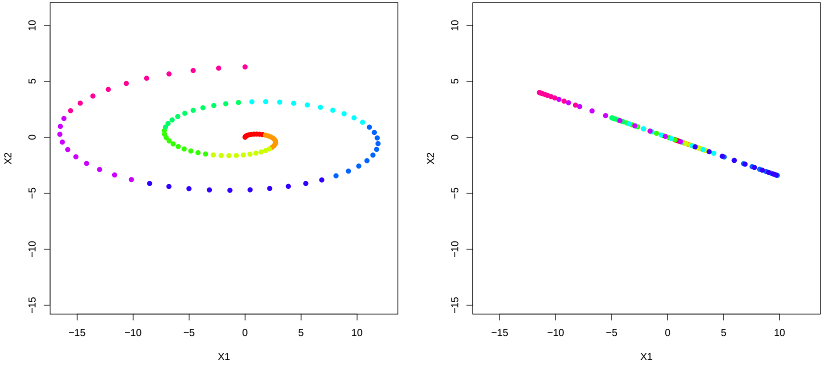
Due to the representer theorem, we know that the solution has the form

$$\hat{g}_1(X) = \sum_{i=1}^{n} c_i k(X, X_i)$$

Additional PCs can be found be enforcing an orthogonality condition

(w.r.t. the inner product on $\mathcal{H}_k$)

Recall



Laplacian Eigenmaps

In order to use the intuitive distance, we need to know the geometry of the data. This needs to be estimated.

We can get an estimate of the distance in the unknown geometry that the data come from (known as a manifold) by altering the usual Euclidean distance.

Some notes:

- The name Laplacian Eigenmaps comes from getting the eigenvector decomposition of the Laplacian restricted to the manifold (which is the second derivative version of the gradient).
- If the manifold is smooth, then local Euclidean distance is an approximation to the distance on the manifold.

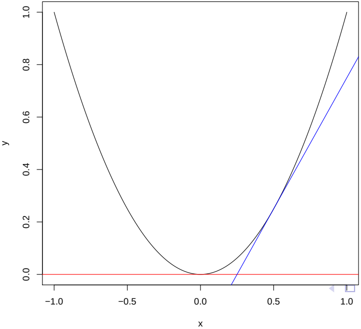
What is a manifold?

How good of an approximation is Euclidean distance?

This question is equivalent to how asking: how quickly does the tangent (space) change?

In 1-D, the tangent space is just the first derivative at that point:

$$f(x) = x^2 \Rightarrow f'(x) = 2x.$$



What is a Manifold?

Therefore, the quality of the (local) Euclidean distance, depends on the second derivative

(ie: how fast does the first derivative change?)

In higher dimensions, the second derivative is known as the Laplacian:

$$\sum_j \frac{\partial^2 f}{\partial x_j^2}$$

(Note: This is also known as the divergence of the gradient)

What are Laplacian Eigenmaps, then?

Imagine the operator $\mathbb{L}$ that performs this operation:

$$\mathbb{L}f = \sum_j \frac{\partial^2 f}{\partial x_j^2}$$

Then $\mathbb{L}$ is the Laplacian, mapping a function to the divergence of its gradient

Key Idea: We can get the eigenvectors/eigenvalues of $\mathbb{L}$. Analogously to PCA, we can now do inference with these eigenvectors.

Note: There is a substantial overlap with KPCA, the difference being the centering of K and the row sum versus column sum normalization

Laplacian Eigenmaps

Collect data: $X_1, \ldots, X_n$ where $X_i \in \mathbb{R}^p$.

1. Form the distance matrix $\Delta_{ij} = ||X_i - X_j||_2^2$.
2. Compute
$$\mathbb{K} = \exp\left(-\frac{\Delta}{\gamma}\right)$$
3. Form the Laplacian $\mathbb{L} = \mathbb{I} - \mathbb{M}^{-1}\mathbb{K}$,
$$\mathbb{M} = \mathtt{diag}(\mathtt{rowSums}(\mathbb{K}))$$
4. Compute the spectrum: $\mathbb{L} = U\Sigma U^\top$.
5. Return U_d, where U_d corresponds to the smallest d (nontrivial) eigenvalues of $\mathbb{L}$

   (Note that the eigenvectors of $\mathbb{L}$ and $\mathbb{M}^{-1}\mathbb{K}$ are the same but the order of the eigenvalues are reversed)

Deeper investigation

1. Form the distance matrix Δ.

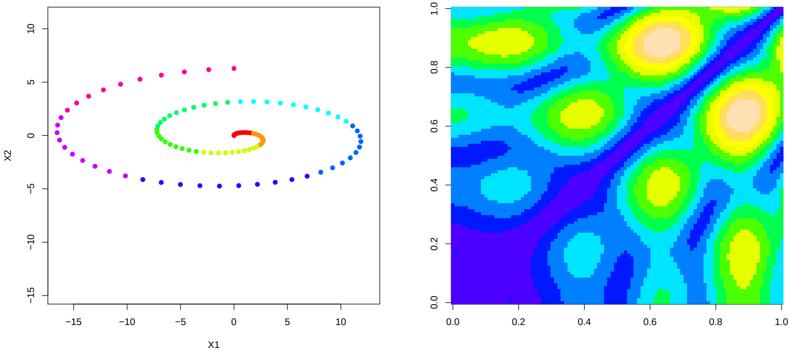


Figure: If we think about the center as 0 and the last blue circle as 1, then each entry the plot on the Right is the Euclidean distance between each data point on the plot on the Left (that is, Δ). The color on the Right plot goes from purple (small distance) to beige/pink (large distance).

Deeper investigation

```
Delta = as.matrix(dist(X,diag=TRUE,upper=TRUE))

image(Delta,col=topo.colors(10))
```

Deeper investigation

2. Exponentiate $-\Delta/\gamma$ to form $\mathbb{K}$ for some fixed γ.

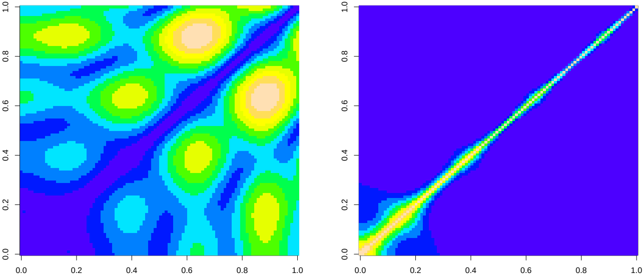
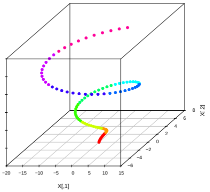


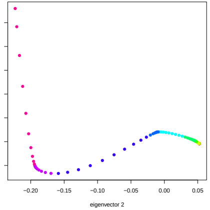
Figure: The Left plot is Δ and the Right plot is $\mathbb{K}$ for $\gamma = 0.95$.

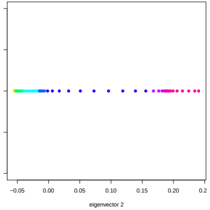
Deeper investigation

```
gamma = 0.95
Wgamma = exp(-Delta/gamma)
image(Wgamma,col=topo.colors(10))
```

Spiral in $\mathbb{R}^3$

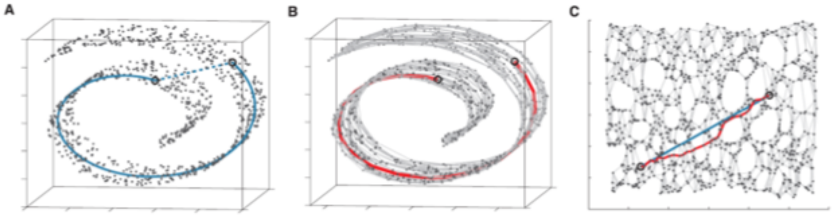






Original data $\quad$ 1^{st} & 2^{nd} nontrivial eigenvectors $\quad$ 1-dimensional

Local Euclidean distance approximates the geodesic



The red line is the local Euclidean path between the two points, while the blue line is the path along the manifold.