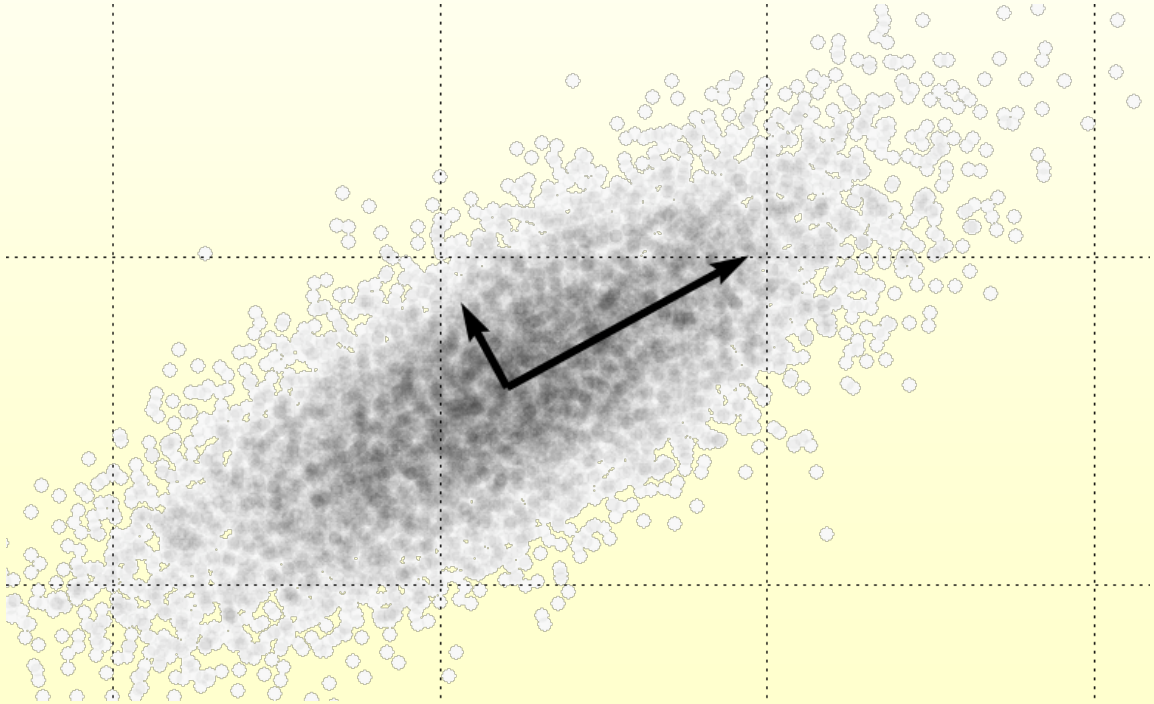
CS233: Geometric and Topological Data Analysis

Principal Components Analysis (PCA)

9 April 2018



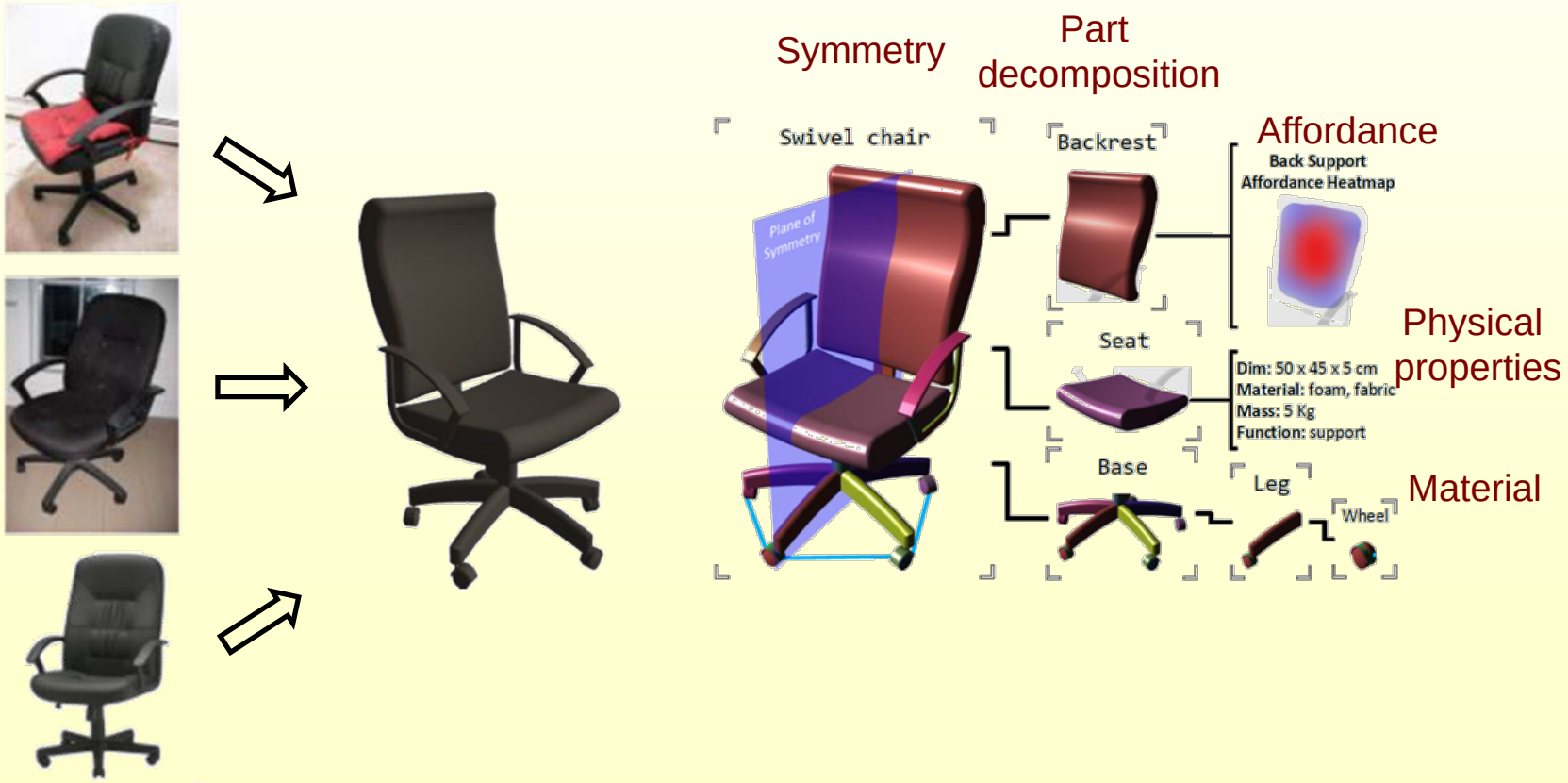
Data Sets and Data Annotation

Data Sets for Supervision

- Explain how big visual datasets including ImageNet and ShapeNet are organized

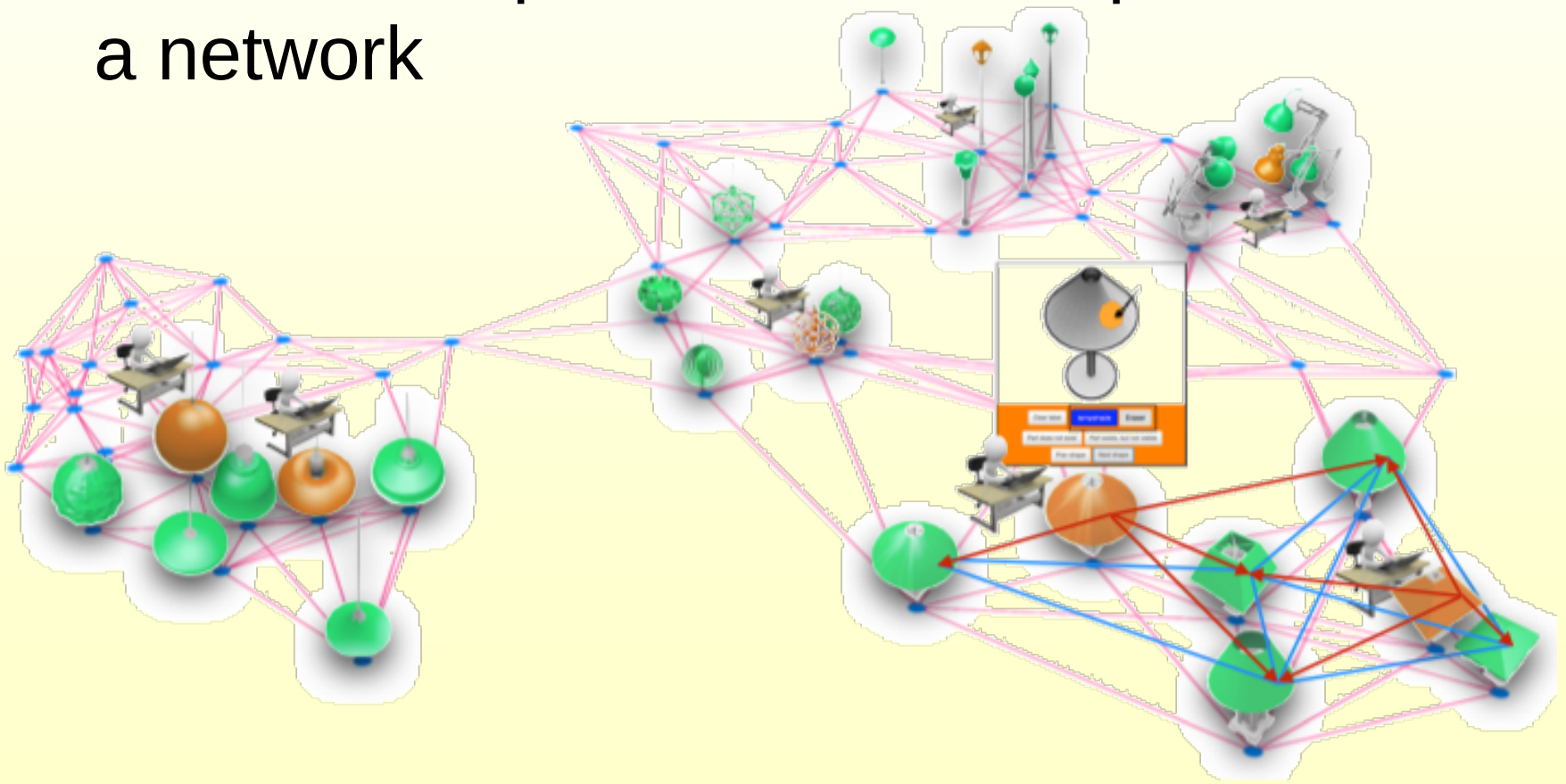
Data Set Annotation

- Explain how ShapeNet is annotated



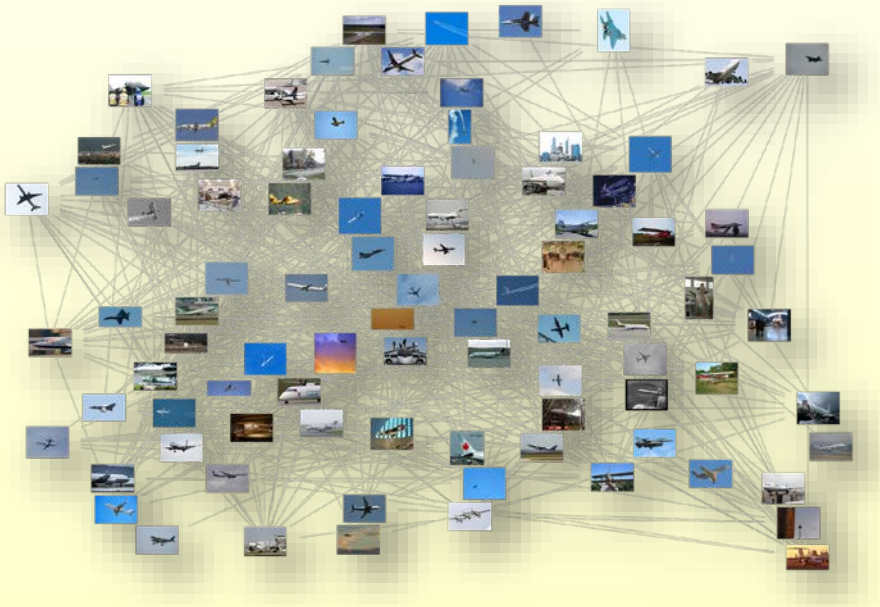
Annotation Transport

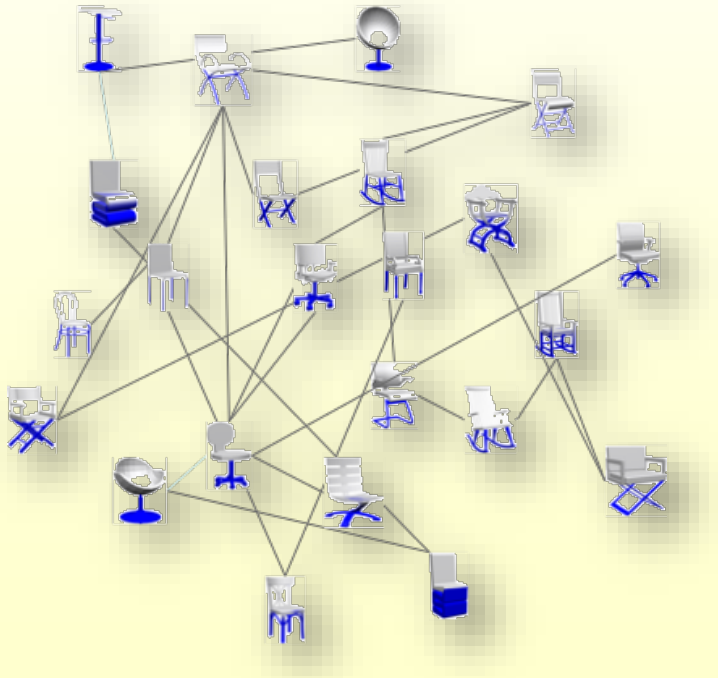
- Show examples of label transportation in a network



Use of Horizontal Data Networks

- Demo that networks can be an effective tool for organizing and annotating big data

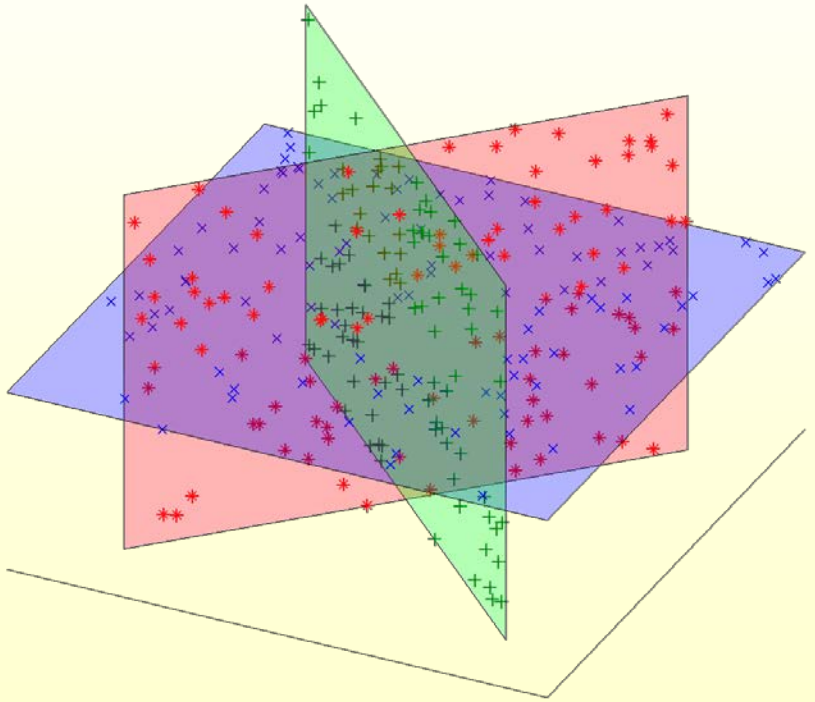




Data as Points in a Euclidean Space

Data as Points in a Euclidean Space

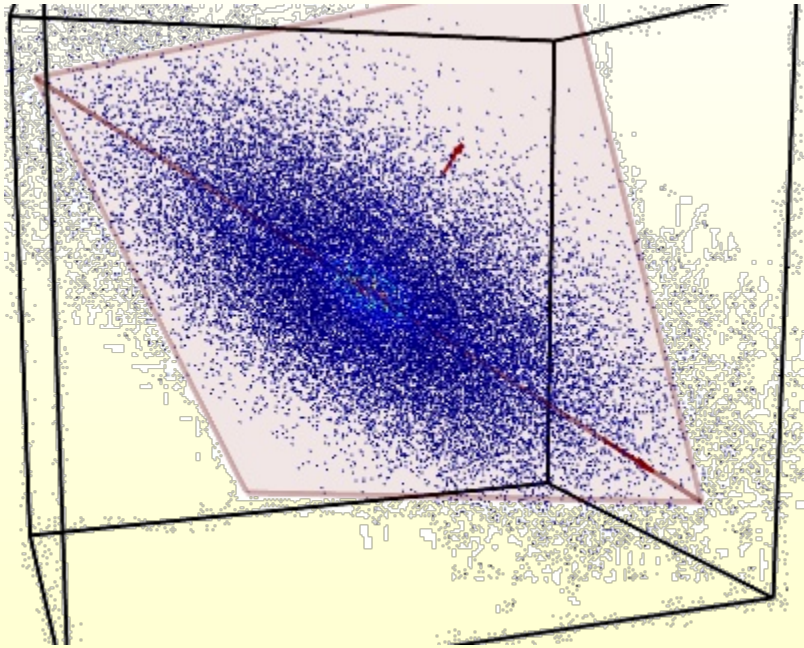
- Given a collection of objects w. many scalar “attributes” – we consider the attributes as dimensions, and therefore of the objects as points in a high dimensional Euclidean space.
- However, lower-dimensional structure is often present in the data.
- Sometimes this lower-d structure corresponds to linear subspaces within the original space.



PCA Intro

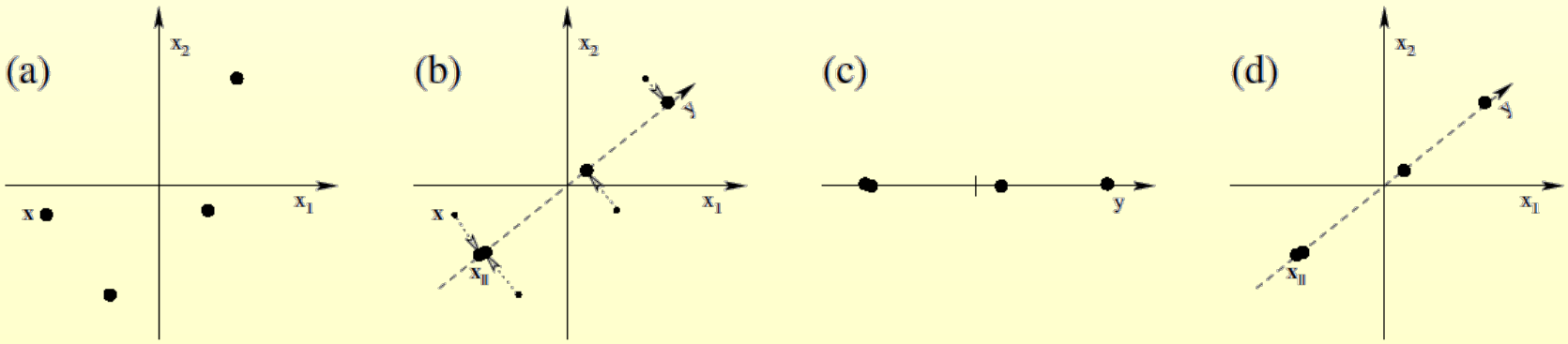
Principal Components Analysis (PCA)

- Introduced by Pearson (1901) and Hotelling (1933) to describe the variation in a set of multivariate data in terms of a set of uncorrelated variables.
- PCA looks for a single lower dimensional subspace that captures most of the variation in the data.
- Specifically, it aims to minimize the error introduced by projecting the data into this linear subspace.



Projection and Reconstruction Error

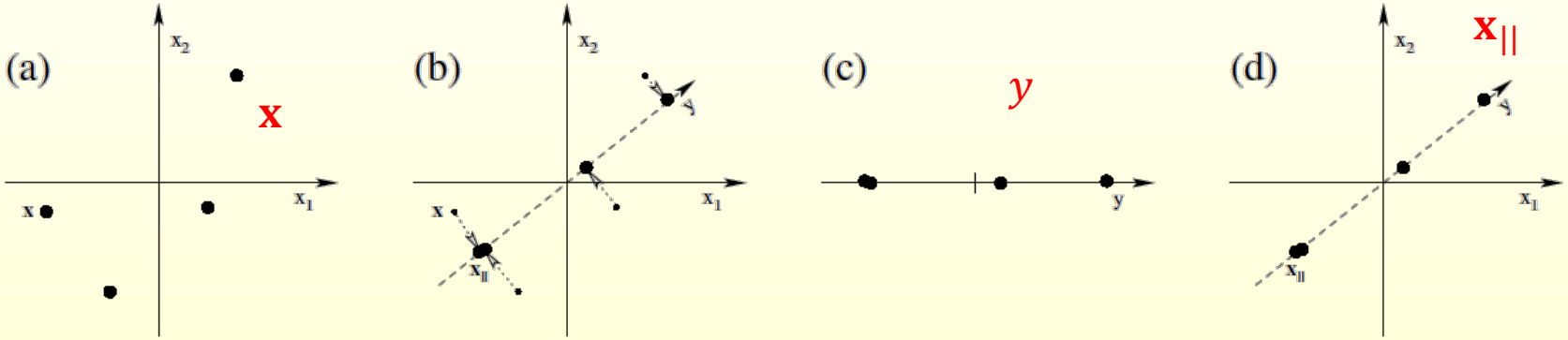
- In practice we often assume zero mean for the data (mean is at the origin, ``centered data'').
- a consequence $\rightarrow$ 2nd moment = variance
- Simple 2D to 1D example



Projection and Reconstruction Error

$\mathbf{x} = (x_1, x_2)$ a point; $\mathbf{v}$ a unit vector normal to projection space

$\mathbf{x}_{||} \coloneqq \mathbf{v}\mathbf{v}^T\,\mathbf{x}$; $y \coloneqq \mathbf{v}^T\,\mathbf{x}$; therefore $\mathbf{x}_{||} \coloneqq \mathbf{v}\,y$



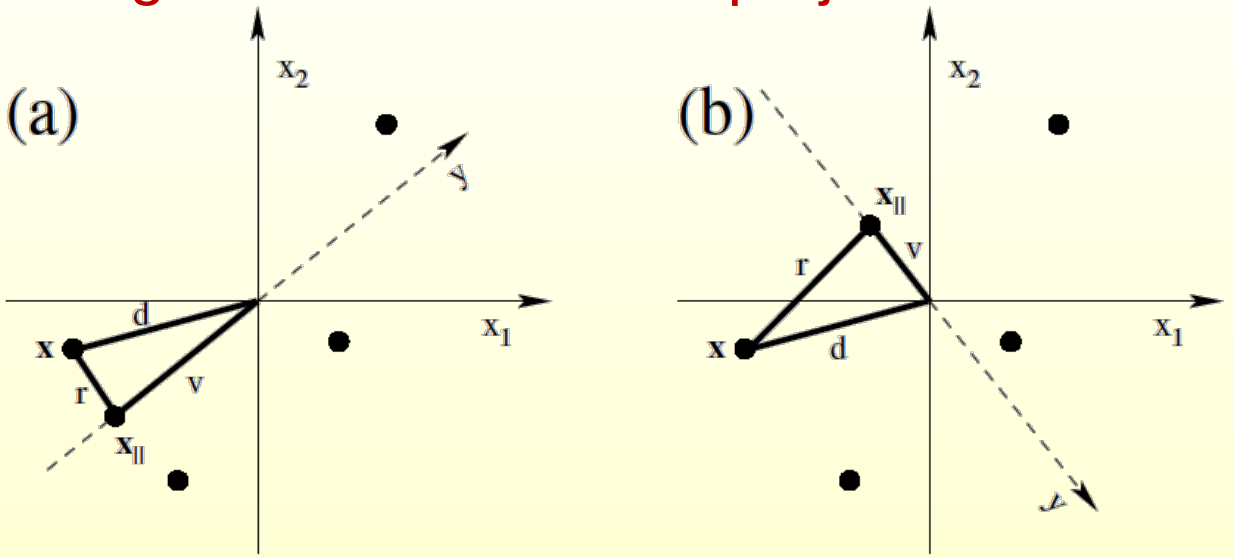
data points in 2D | projection onto 1D | data points in 1D | reconstruction in 2D

Error $E \coloneqq \left\langle \left\| x^\mu - x_{||}^\mu \right\|^2 \right\rangle_\mu = \frac{1}{M} \sum_{\mu=1}^{M} \sum_{i=1}^{I} (x_i^\mu - x_{||i}^\mu)^2$

M points, *I* dimensions

Reconstruction Error and Variance

- Minimizing the reconstruction error is equivalent to maximizing the variance of the projected data.



- Remember the mean is 0

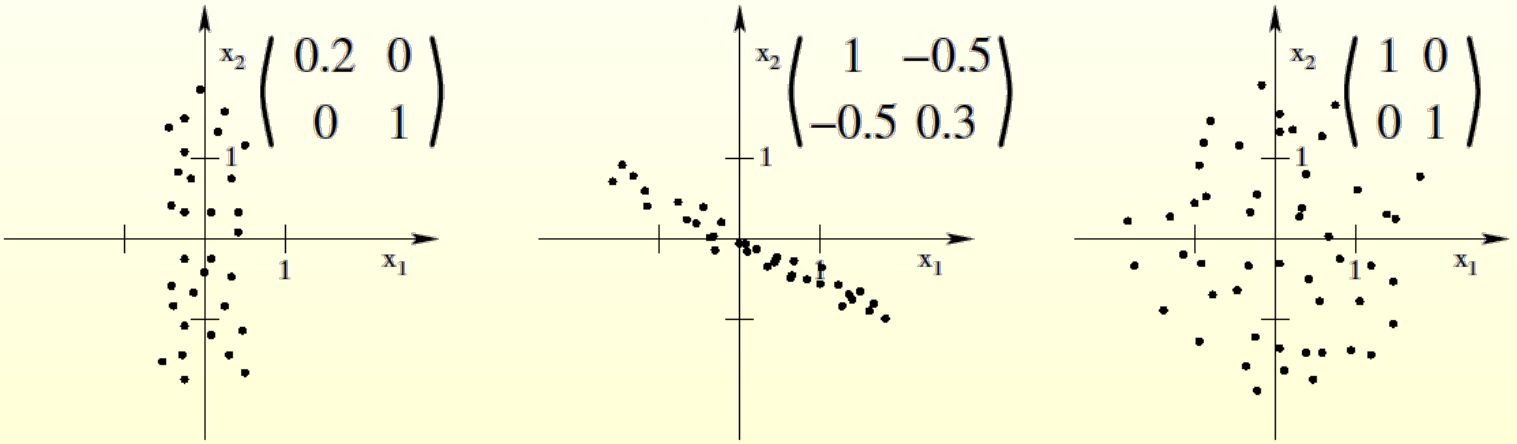
$$r^2 + v^2 = d^2$$

NB, projections of centered data are centered

Covariance Matrix

<> indicate inner products

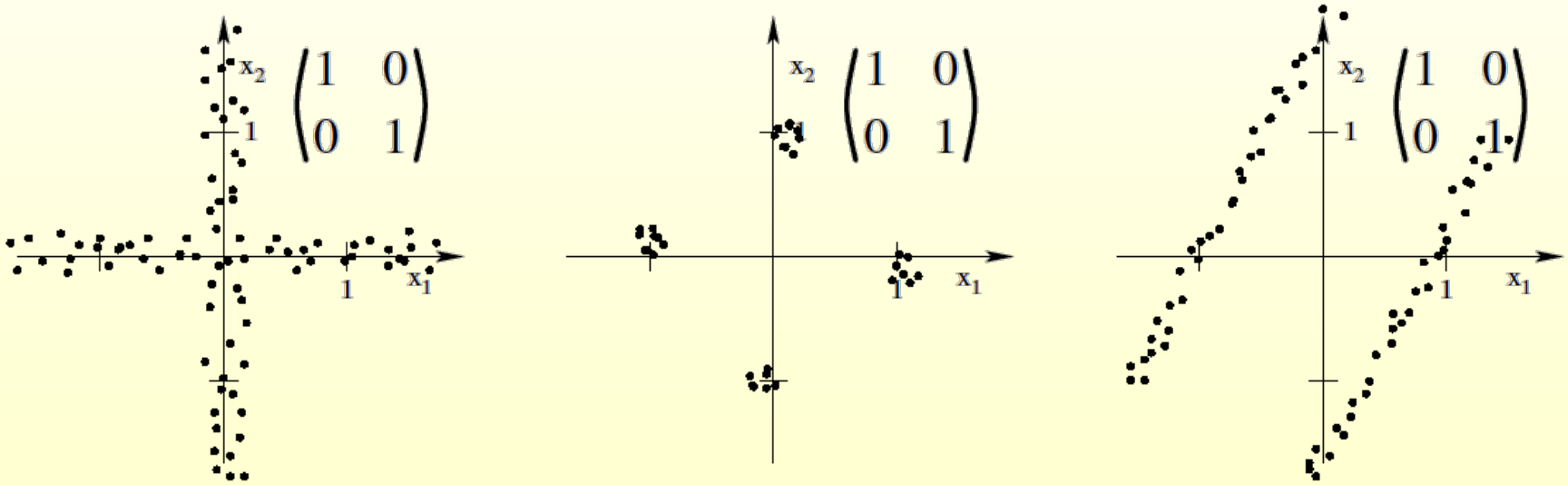
- $\mathbf{x} = (x_1, x_2)^T$, covariance $C_{12} = \langle x_1 x_2 \rangle$ [Remember: centered data]
- (Co-)variances of the 1st and 2nd components: $C_{11} = \langle x_1 x_1 \rangle$, $C_{22} = \langle x_2 x_2 \rangle$



- 2x2 covariance matrix with entries $C_{ij} = C_{ji} := \langle x_i x_j \rangle$
- $\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$

Covariance is Not the Full Story

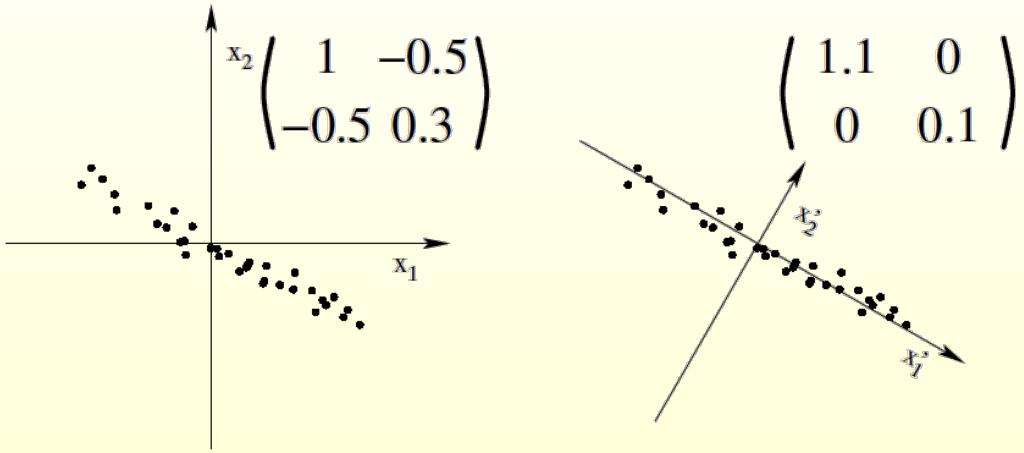
- The covariance matrix tells us about the 2nd order moments, but not about the higher-order structure of the data.



- When it comes to 1st and 2nd order moments, might as well assume a normal distribution, if all we know is the covariance matrix.

PCA by Diagonalizing the Covariance Matrix

- Finding the direction of maximum variance is easy if the covariance matrix is diagonal.



- So, in general, we want to rotate the coordinate system so as to make the covariance matrix diagonal.
- This is an eigenvalue problem; the eigenvectors of the covariance matrix point in the directions of maximal and minimal variance.

The General Case

The PCA Problem

Principal Component Analysis (PCA): Given a set $\{\mathbf{x}^\mu : \mu = 1, ..., M\}$ of I-dimensional data points $\mathbf{x}^\mu = (x_1^\mu, x_2^\mu, ..., x_I^\mu)^T$ with zero mean, $\langle \mathbf{x}^\mu \rangle_\mu = \mathbf{0}_I$, find an orthogonal matrix $\mathbf{U}$ with determinant $|\mathbf{U}| = +1$ generating the transformed data points $\mathbf{x}'^\mu := \mathbf{U}^T \mathbf{x}^\mu$ such that for any given dimensionality P the data projected onto the first P axes, $\mathbf{x}'^\mu_{||} := (x'^\mu_1, x'^\mu_2, ..., x'^\mu_P, 0, ..., 0)^T$, have the smallest

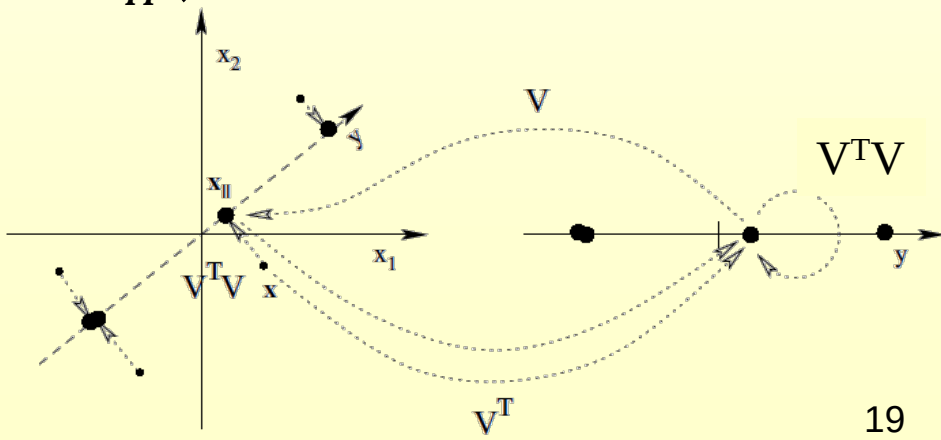
$$\text{reconstruction error} \quad E := \langle \|\mathbf{x}'^\mu - \mathbf{x}'^\mu_{||}\|^2 \rangle_\mu \tag{8}$$

among all possible projections onto a P-dimensional subspace. The row vectors of matrix $\mathbf{U}$ define the new axes and are called the *principal components*.

High-D to Low-D Projection: The Projection Matrix $\mathbf{V}^T$

- Within the large *I*-dimensional space lies a *P*-dimensional subspace spanned by *P* orthonormal vectors $\mathbf{v}_p$ ($P < I$).
- $\mathbf{v}_p^T \mathbf{v}_q = \delta_{pq} = \{1 \text{ if } p = q,\ 0 \text{ otherwise}$
- $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_P) = \begin{pmatrix} v_{11} & \cdots & v_{1P} \\ \vdots & \ddots & \vdots \\ v_{I1} & \cdots & v_{IP} \end{pmatrix}$
- $\mathbf{y} = \mathbf{V}^T \mathbf{x}$

Projection, and a new coordinate system



Low-D Inside High-D: $\mathbf{V}$

- What does $\mathbf{V}^T\mathbf{V}$ do?
- Recall $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_P) = \begin{pmatrix} v_{11} & \cdots & v_{1P} \\ \vdots & \ddots & \vdots \\ v_{I1} & \cdots & v_{IP} \end{pmatrix}$
  - $\mathbf{v}_p^T\mathbf{v}_q = \delta_{pq}$
- $\boldsymbol{V^TV = I}$, simply the *p*-dimensional identity matrix

---

- Projection $\mathbf{x}_{||} = \mathbf{V}\mathbf{y} = \mathbf{V}\mathbf{V}^T\mathbf{x}$
- These are the original points projected in the low dimensional space (now sitting inside the high dimensional space).

The Projection Operator $\mathbf{V}\mathbf{V}^T$

- This simply projects the points of the big *I*-dimensional space onto the small *P*-dimensional space, living inside the big *I*-dimensional space

- $\mathbf{P} = \mathbf{V}\mathbf{V}^T$; $\mathbf{P}\mathbf{P} = (\mathbf{V}\mathbf{V}^T)(\mathbf{V}\mathbf{V}^T) = \mathbf{V}(\mathbf{V}^T\mathbf{V})\mathbf{V}^T = \mathbf{V}\mathbf{V}^T$.
- $\mathbf{P}$ is an *I*x*I* matrix
- If $P = I$, then in fact $\mathbf{P}$ is the identity matrix
- But in general $P = \sum_{p=1}^{P} \mathbf{v}_p \mathbf{v}_p{}^T$ loses information, since $P < I$

Variance and Reconstruction Error

- For a zero mean vector variance is the 2nd moment
  - $\mathrm{Var}(x) \coloneqq \sum_{i=1}^{I} \langle x_i^2 \rangle = \langle \sum_{i=1}^{I} x_i^2 \rangle = \langle \mathbf{x}^T \mathbf{x} \rangle$.

- Reconstruction error
  - Look at $\mathbf{x} - \mathbf{x}_{||}$
  - $E = \left\langle \left(\mathbf{x} - \mathbf{x}_{||}\right)^T \left(\mathbf{x} - \mathbf{x}_{||}\right) \right\rangle = \langle (\mathbf{x} - \mathbf{V}\mathbf{V}^T\mathbf{x})^T (\mathbf{x} - \mathbf{V}\mathbf{V}^T\mathbf{x}) \rangle$
  - $= \langle \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} + \mathbf{x}^T\mathbf{V}(\mathbf{V}^T\mathbf{V})\mathbf{V}^T\mathbf{x} \rangle$
  - $= \langle \mathbf{x}^T\mathbf{x} \rangle - \langle \mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} \rangle = \langle \mathbf{x}^T\mathbf{x} \rangle - \langle (\mathbf{V}^T\mathbf{x})^T\mathbf{V}^T\mathbf{x} \rangle$
  - $= \langle \mathbf{x}^T\mathbf{x} \rangle - \langle \mathbf{y}_{||}^T\mathbf{y}_{||} \rangle$
  - Therefore maximizing the variance of the projected data minimizes the reconstruction error.

Eigenvalues of the Covariance Matrix

- The covariance matrix $\mathbf{C}_x$
  - $\mathbf{C}_x \coloneqq \langle \mathbf{x}^T \mathbf{x} \rangle = \frac{1}{M} \sum_\mu \mathbf{x}^\mu \mathbf{x}^{\mu T}$
- Since the covariance matrix is symmetric, it always has real eigenvalues and orthogonal eigenvectors
  - $\mathbf{C}_x \mathbf{u}_i = \mathbf{u}_i \lambda_i$
  - $\lambda_i \geq \lambda_{i+1}$
  - $\mathbf{u}_i \mathbf{u}_j = \delta_{ij}$
- Now form
  - $\mathbf{U} \coloneqq (\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_I)$ – the eigenvectors of the covariance matrix
  - $\mathbf{\Lambda} \coloneqq diag(\lambda_1, \lambda_2, \ldots \lambda_I)$ – the eigenvalues of the covariance matrix

Some Useful Facts

- $\mathbf{U}^T\mathbf{U} = 1_I$
- $\mathbf{U}\mathbf{U}^T = 1_I$
- $\mathbf{C}_x\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda}$
- $\mathbf{U}^T\mathbf{C}_x\mathbf{U} = \boldsymbol{\Lambda}$
- $\mathbf{C}_x = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$

- The total variance of $\mathbf{x}$
  - $\langle \mathbf{x}^T\mathbf{x} \rangle = \langle \mathrm{tr}(\mathbf{x}^T\mathbf{x}) \rangle = \mathrm{tr}(\langle \mathbf{x}\mathbf{x}^T \rangle) = \mathrm{tr}(\mathbf{C}_x) = \mathrm{tr}(\mathbf{U}\mathbf{U}^T\mathbf{C}_x) = \mathrm{tr}(\mathbf{U}^T\mathbf{C}_x\mathbf{U}) = \mathrm{tr}(\boldsymbol{\Lambda}) = \sum_i \lambda_i$
  - Thus the total variance of the data is just the sum of the eigenvalues of the covariance matrix

Diagonalizing the Covariance Matrix

- Now we use orthogonal matrix **U** to transform the data so that the covariance matrix becomes diagonal

$$
\begin{aligned}
\bullet \quad \mathbf{x}' &:= \mathbf{U}^T\mathbf{x} \\
\bullet|\circ \quad \mathbf{C}'_x &:= \left\langle \mathbf{x}'\mathbf{x}'^T \right\rangle \\
\circ &\overset{(58)}{=} \left\langle (\mathbf{U}^T\mathbf{x})(\mathbf{U}^T\mathbf{x})^T \right\rangle \\
\circ &= \mathbf{U}^T \left\langle \mathbf{x}\mathbf{x}^T \right\rangle \mathbf{U} \\
\circ &\overset{(39)}{=} \mathbf{U}^T\mathbf{C}_x\mathbf{U}\,, \\
\bullet &\overset{(48)}{=} \mathbf{\Lambda}
\end{aligned}
$$

Variance for a Diagonalized Covariance Matrix

- Is this new coordinate system, which *P*-dimensional subspace minimizes the reconstruction error?
- For any $V' = (v'_1, v'_2, \ldots v'_P)$ formed from *P* orthonormal vectors
  - $\mathbf{y} := \mathbf{V}'^T \mathbf{x}'$
  - $\implies \langle \mathbf{y}^T \mathbf{y} \rangle \overset{(64)}{=} \langle \mathbf{x}'^T \mathbf{V}' \mathbf{V}'^T \mathbf{x}' \rangle$
    - $= \langle \mathrm{tr}(\mathbf{x}'^T \mathbf{V}' \mathbf{V}'^T \mathbf{x}') \rangle$ (since $s = \mathrm{tr}(s)$ for any scalar s)
    - $= \langle \mathrm{tr}(\mathbf{V}'^T \mathbf{x}' \mathbf{x}'^T \mathbf{V}') \rangle$ (since $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{BCA})$ if defined)
      - $\overset{(59)}{=} \mathrm{tr}(\mathbf{V}'^T \mathbf{C}'_x \mathbf{V}')$ (since $\mathrm{tr}(\cdot)$ and $\langle \cdot \rangle$ commute)
      - $\overset{(63)}{=} \mathrm{tr}(\mathbf{V}'^T \mathbf{\Lambda} \mathbf{V}')$
      - $= \sum_i \lambda_i \sum_p (v'_{ip})^2 .$ (as one can work out on a sheet of paper)

Variance Maximization

- To maximize the variance $< \boldsymbol{y}^T \boldsymbol{y} >$ we clearly need to put as much weight as possible on the large eigenvalues
- So the variance is maximized by using V' to project $\boldsymbol{x}'$ onto the first P eigenvectors of the covariance matrix C_x
- It is easy to check that optimal variance is $\sum_{i=1}^{P} \lambda_i$
- and that the reconstruction error is $\sum_{i=P+1}^{I} \lambda_i$

Detail

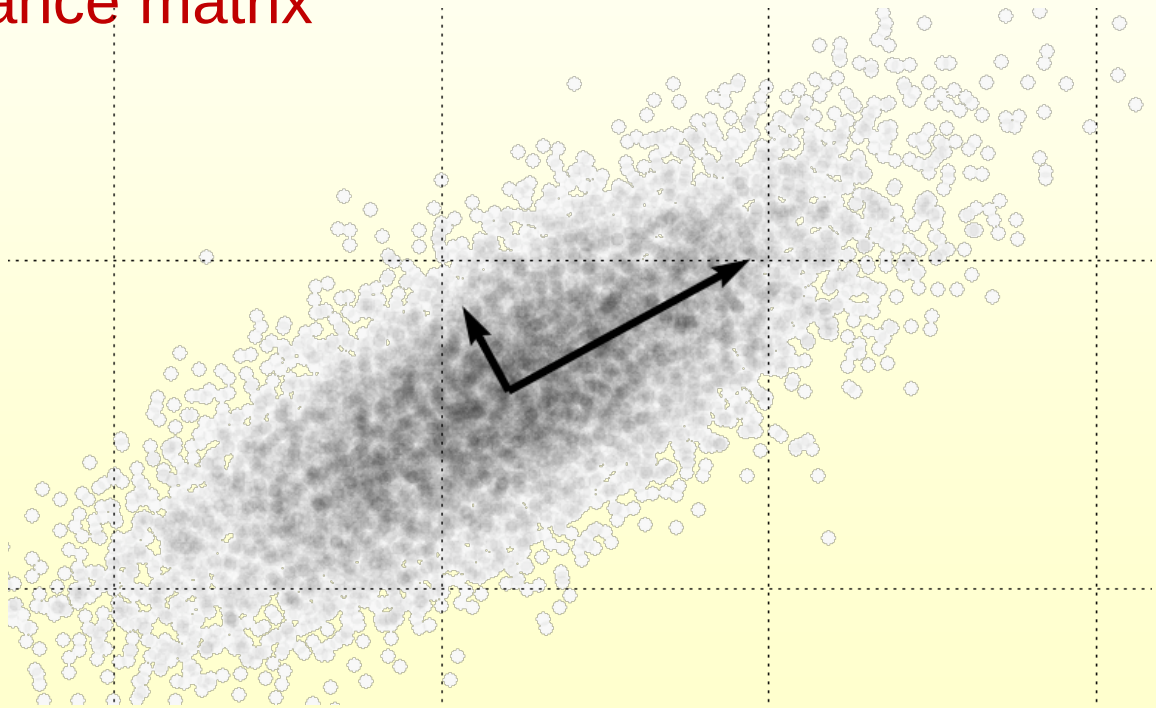
- Constraints on V'
  - $$\sum_i (v'_{ip})^2 = 1 \quad \text{(column vectors of } \mathbf{V}' \text{ have norm one)},$$
  - $$\Longrightarrow \quad \sum_{ip} (v'_{ip})^2 = P \quad \text{(square sum over all matrix elements equals } P),$$
  - $$\sum_p (v'_{ip})^2 \leq 1 \quad \text{(row vectors of } \mathbf{V}' \text{ have norm less or equal one)}.$$
- Best choice
  - $$v'_{ip} := \delta_{ip} := \begin{cases} 1 & \text{if } i = p \\ 0 & \text{otherwise} \end{cases}$$
- Since $P \leq I$

$$\sum_p (v'_{ip})^2 \overset{(74)}{=} \sum_p \delta_{ip}^2 = \left\{ \begin{array}{ll} 1 & \text{if } i \leq P \\ 0 & \text{otherwise} \end{array} \right\} \leq 1,$$

$$\sum_i (v'_{ip})^2 \overset{(74)}{=} \sum_i \delta_{ip}^2 = \delta_{pp}^2 = 1,$$

PCA Final

- For any *P*, the optimal *P*-dimensional subspace is the one spanned by the first *P* eigenvectors of the covariance matrix



PCA Examples

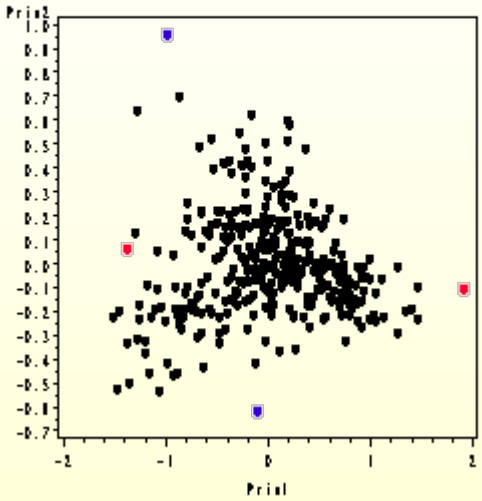
PCA Example 1: Place Ratings

- The “Places Rated Almanac” rates 329 communities according to the following nine criteria:
  - Climate and Terrain
  - Housing
  - Health Care & the Environment
  - Crime
  - Transportation
  - Education
  - The Arts
  - Recreation
  - Economics

Place Rated

	Principal Component		
Variable	**1**	**2**	**3**
Climate	0.190	0.017	0.207
Housing	**0.544**	0.020	0.204
Health	**0.782**	**-0.605**	0.144
Crime	0.365	0.294	**0.585**
Transportation	**0.585**	0.085	0.234
Education	0.394	-0.273	0.027
Arts	**0.985**	0.126	-0.111
Recreation	**0.520**	0.402	**0.519**
Economy	0.142	0.150	0.239

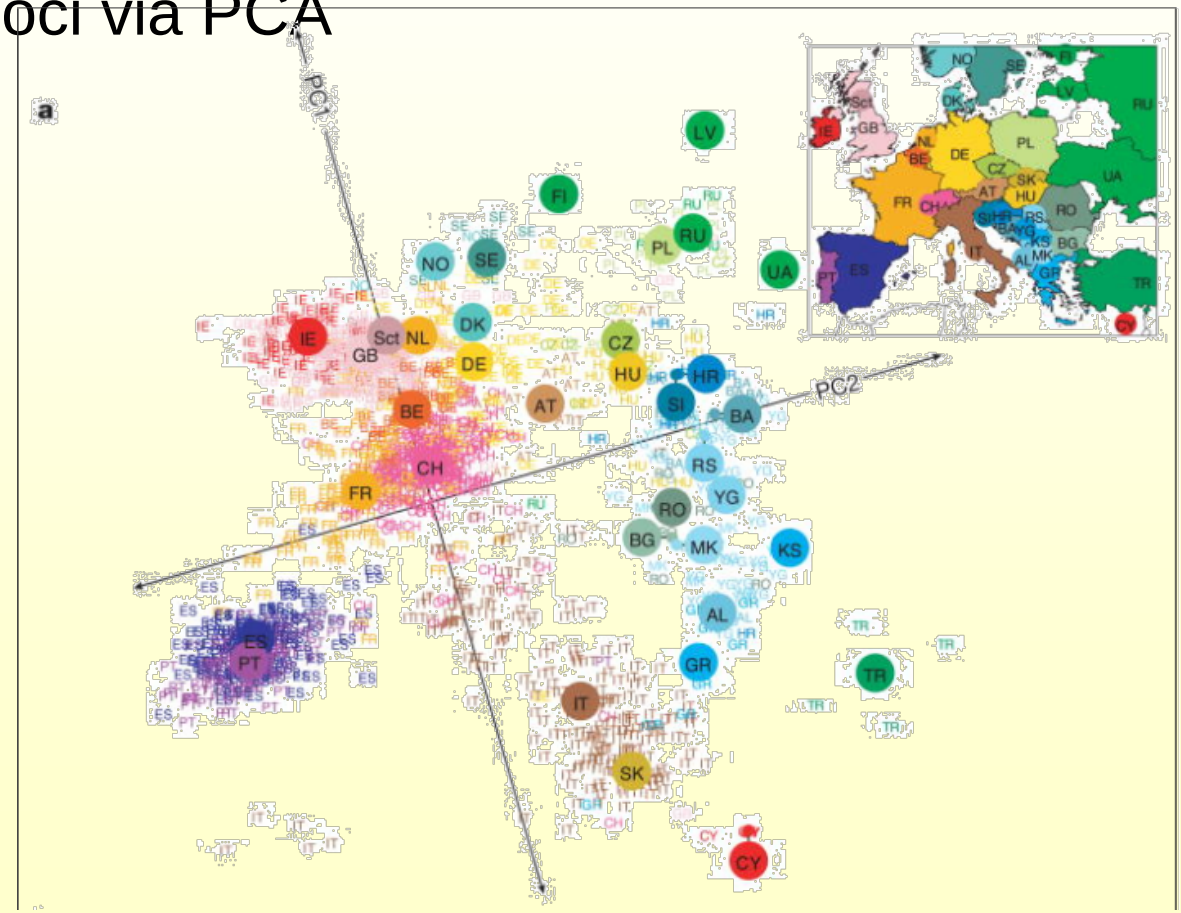
PCA — Covariance Matrix — Places Rated



- The first principal component is strongly positively correlated with five of the original variables: Arts, Health, Transportation, Housing and Recreation scores. This suggests that these five criteria vary together. If one increases, then the remaining also increase.
- Furthermore, we see that the first principal component correlates most strongly with the Arts

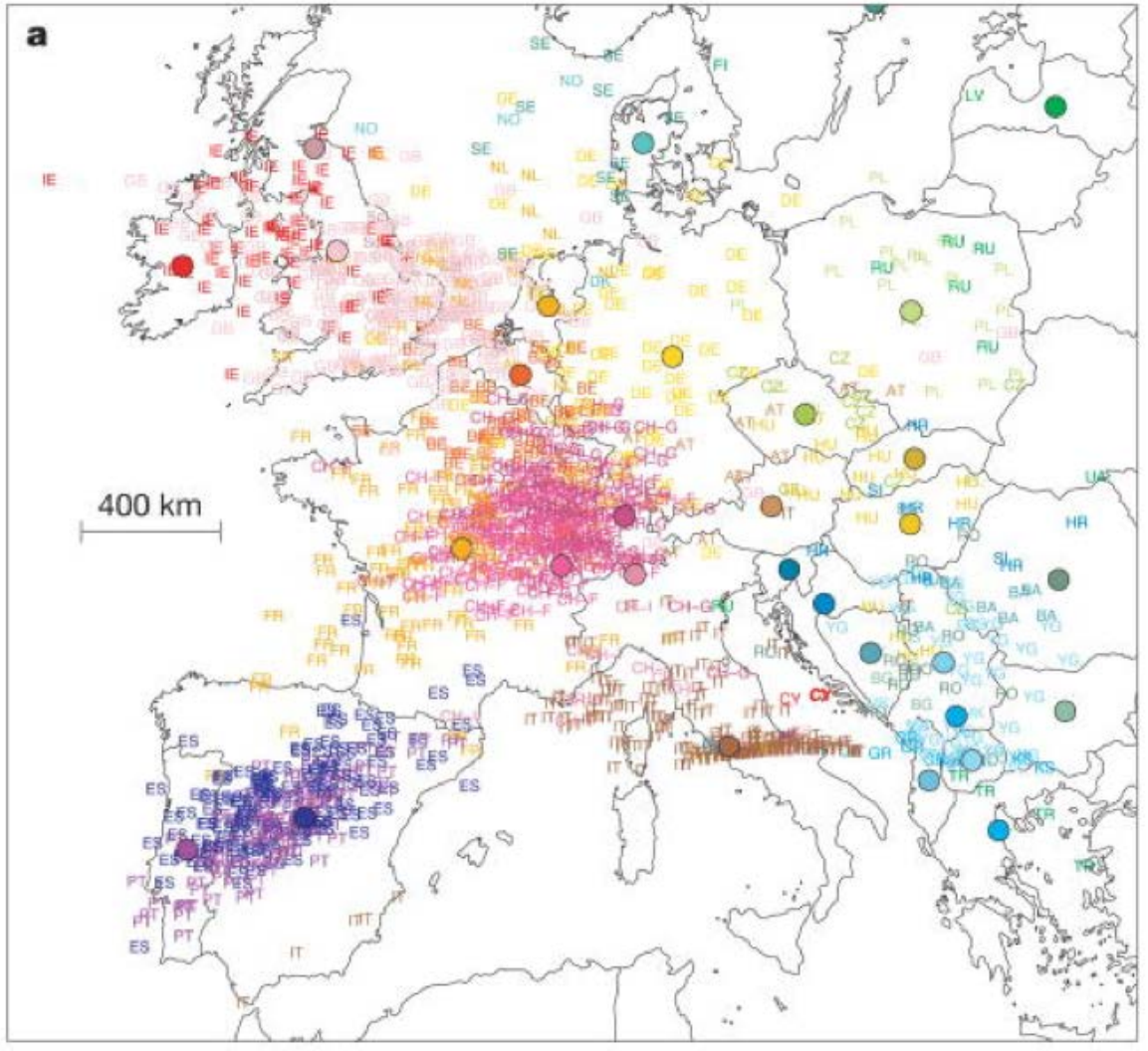
PCA Example 2: Genetic vs. Geographic Diversity in Europe

- Geonome of 3,192 individuals studied based on 197,146 SNIP loci via PCA

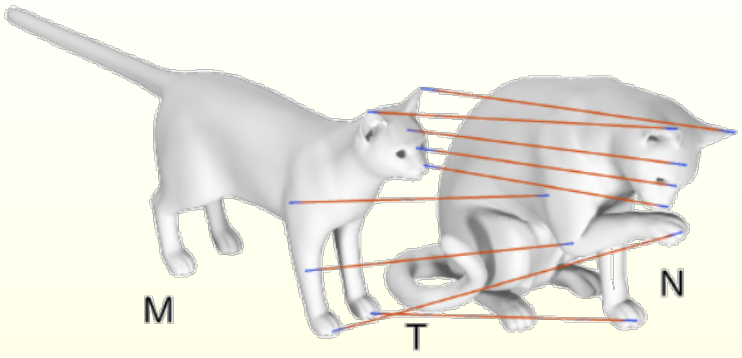


J Novembre *et al. Nature* **456**, 1-4 (2008) doi:10.1038/nature07331

Close Up



Functional Maps

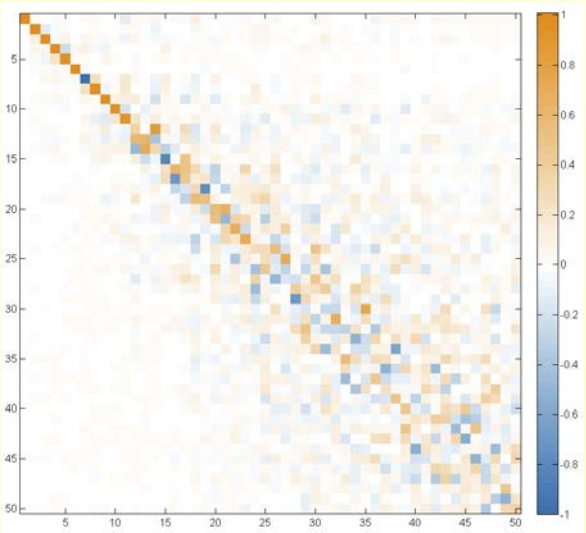


f

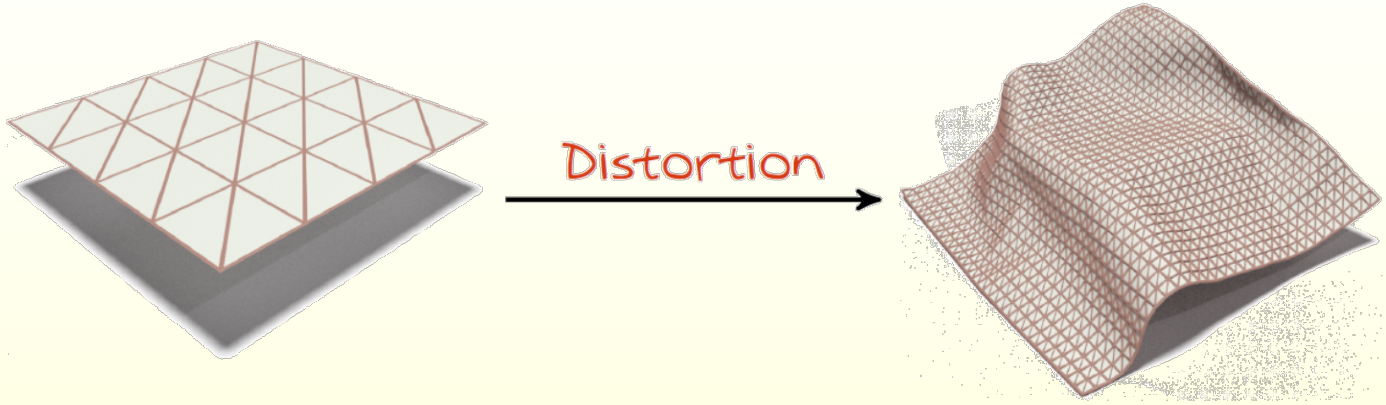
f ∘ T

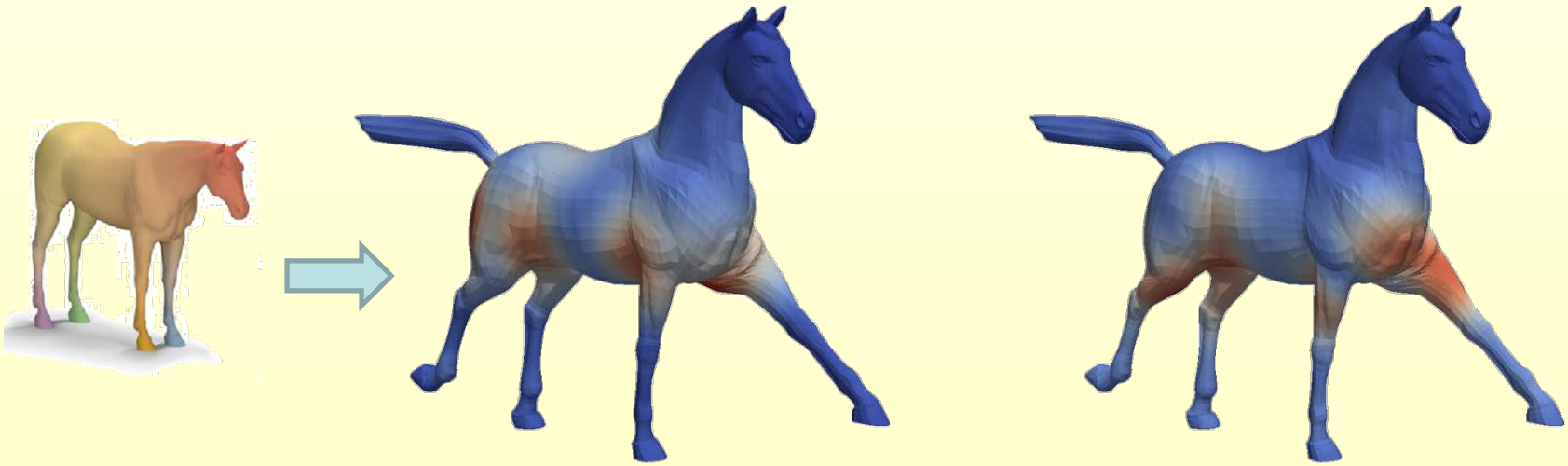
$$T : N \to M$$

$$C_T : \begin{array}{ccc} L^2(M) & \to & L^2(N) \\ f & \mapsto & f \circ T \end{array}$$



Distortions Caused by a Map

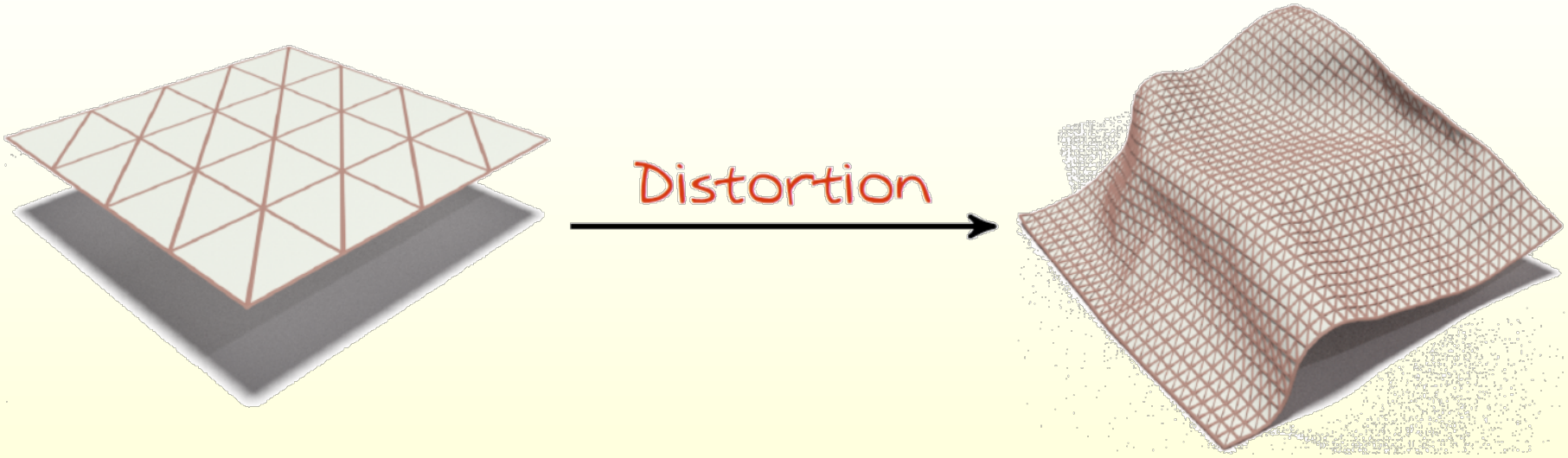




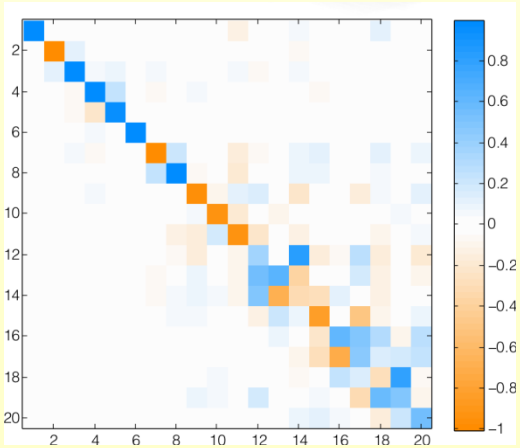
Area distortion

Conformal distortion

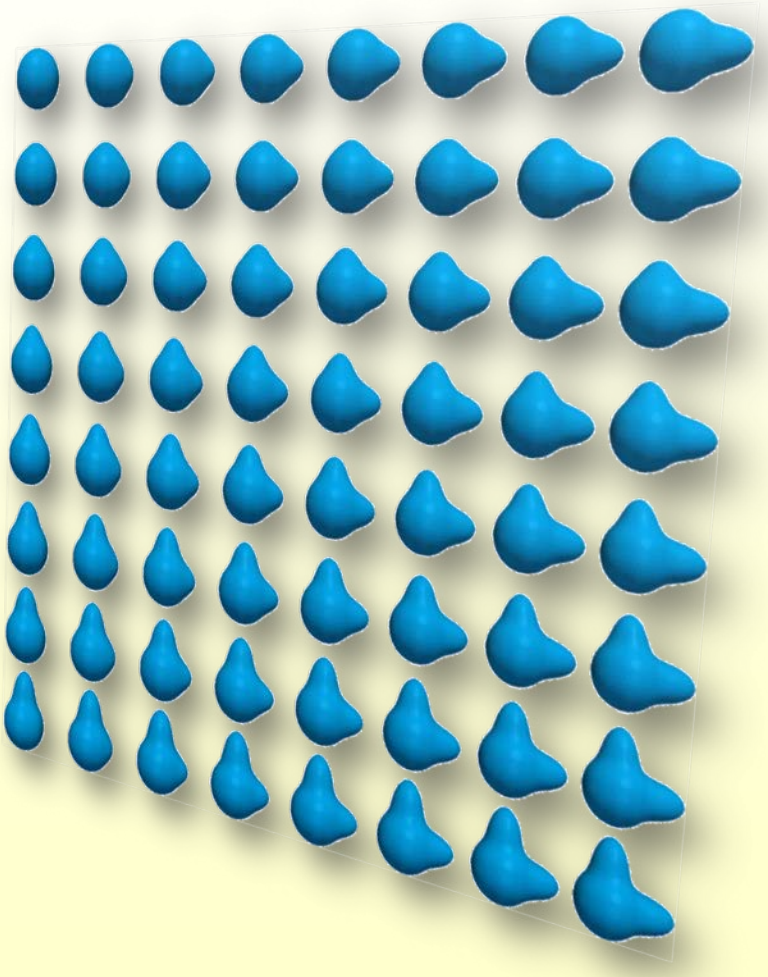
Shape Differences are a Change Recipe

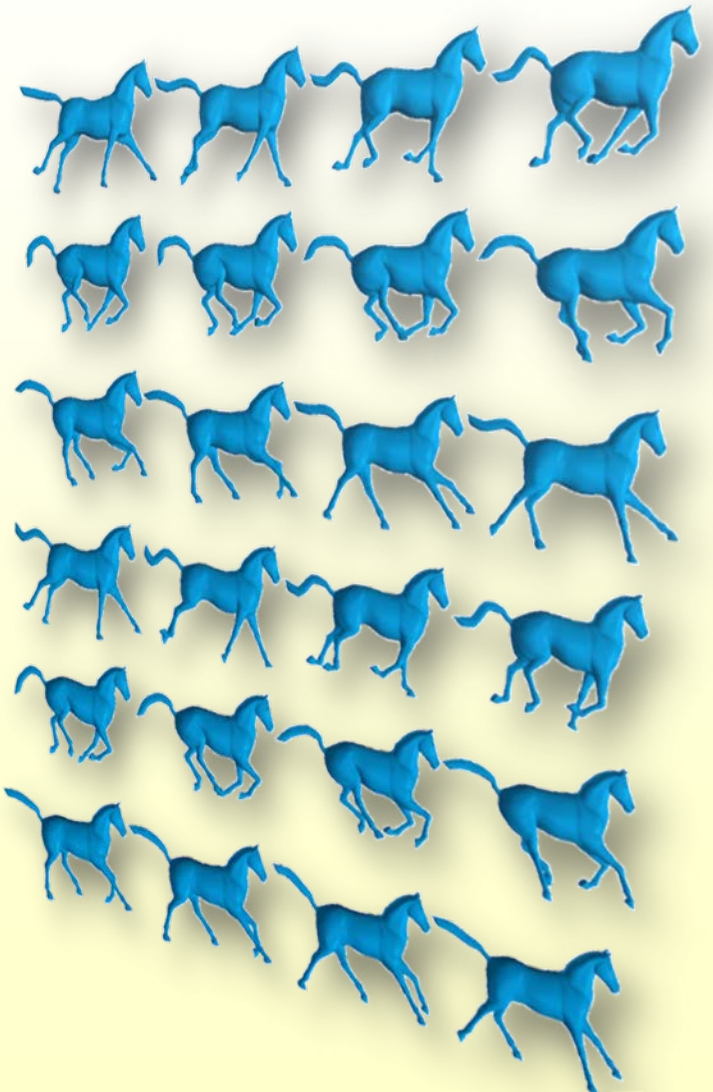


A recipe encoded as a matrix:

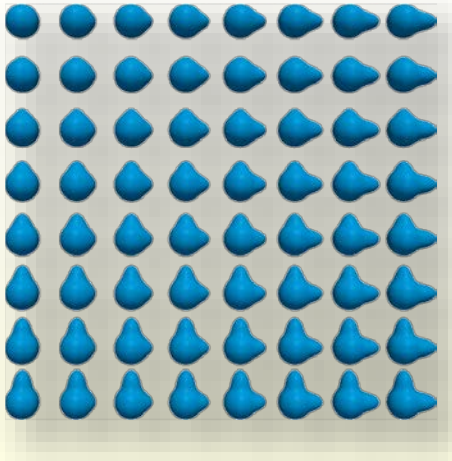


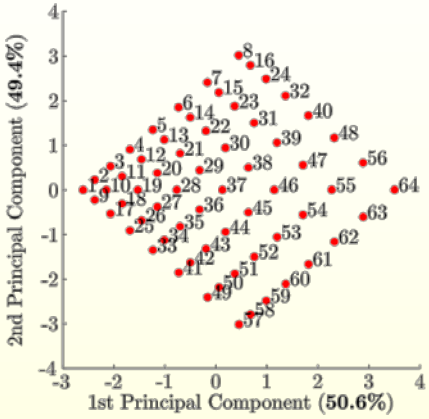
Shape Differences in Collections

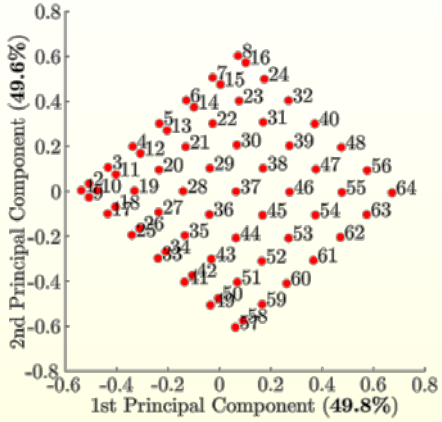


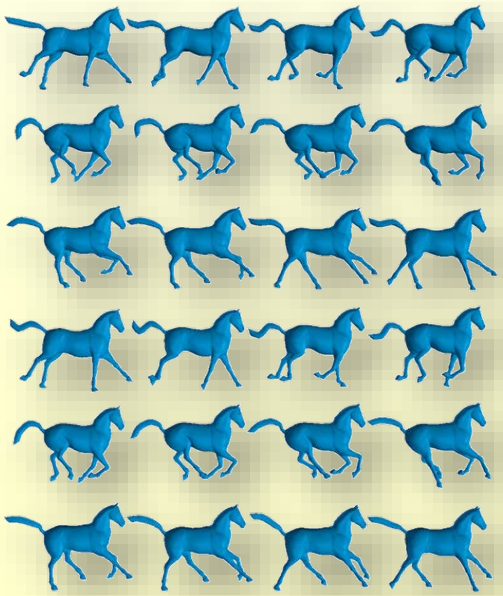


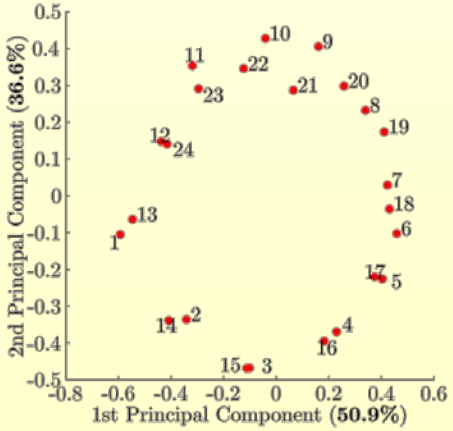
Can Be Used to Analyze Shape Collections

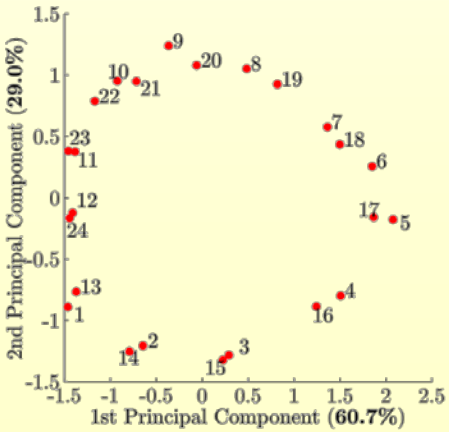












A Tidbit

Sometimes SVD to the Rescue

- What to do when we have fewer data points than dimensions ($M < I$)?

- We have, $\mathbf{X} := (\mathbf{x}^1, ..., \mathbf{x}^M)$, $\mathbf{C}_1 := \mathbf{X}\mathbf{X}^T/M$
- So

$$\mathbf{C}_1\mathbf{U}_1 = \mathbf{U}_1\mathbf{\Lambda}_1$$
$$\Longleftrightarrow \quad \mathbf{C}_1 = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T$$

- But $\mathbf{Y}_1 := \mathbf{U}_1^T\mathbf{X}$ is still high dimensional, because the eigenvectors are of dim I

Data SVD, II

- So let's transpose $\boldsymbol{X} \rightarrow \boldsymbol{X}^T$
- We get,

$$\begin{aligned} \mathbf{C}_2 &:= \mathbf{X}^T\mathbf{X}/I\,, \\ \mathbf{C}_2\mathbf{U}_2 &= \mathbf{U}_2\mathbf{\Lambda}_2 \\ \Longleftrightarrow \quad \mathbf{C}_2 &= \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T\,, \\ \mathbf{Y}_2 &:= \mathbf{U}_2^T\mathbf{X}^T\,. \end{aligned}$$

- Moreover, the rows of $\mathbf{Y}_2$ are eigenvectors of C_1

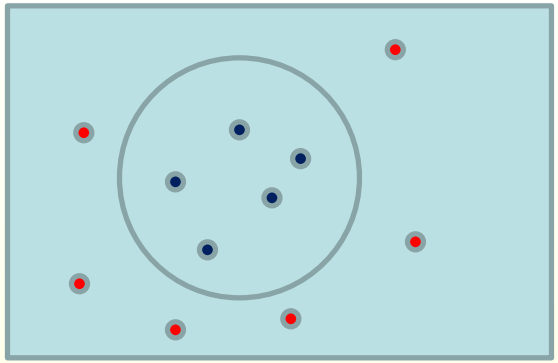
$$\begin{aligned} \mathbf{C}_1\mathbf{Y}_2^T &\overset{(100,107)}{=} (\mathbf{X}\mathbf{X}^T/M)(\mathbf{X}\mathbf{U}_2) \\ &= \mathbf{X}(\mathbf{X}^T\mathbf{X}/I)\mathbf{U}_2 I/M \\ &\overset{(104)}{=} \mathbf{X}\mathbf{C}_2\mathbf{U}_2 I/M \\ &\overset{(105)}{=} \mathbf{X}\mathbf{U}_2\mathbf{\Lambda}_2 I/M \\ &\overset{(107)}{=} \mathbf{Y}_2^T\mathbf{\Lambda}_2 I/M\,. \end{aligned}$$

- The corresponding eigenvalues are just those of C_2 scaled by *I*/*M*.

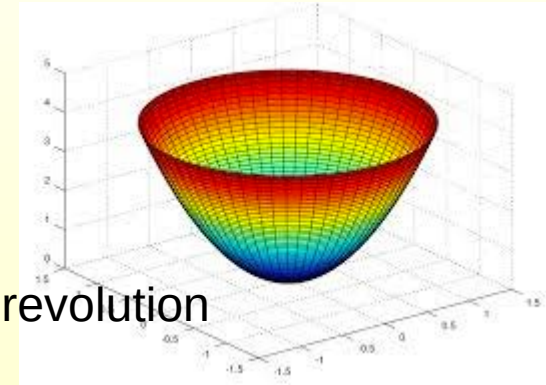
Kernels

Kernel Methods – Non-Linear Relationships

- "Lifting" maps can linearize non-linear structure



- Example: circular separability in the 2D plane lifts to planar separability in 3D space, via



paraboloid of revolution

$$\Phi(x, y) = (x, y, x^2 + y^2)$$

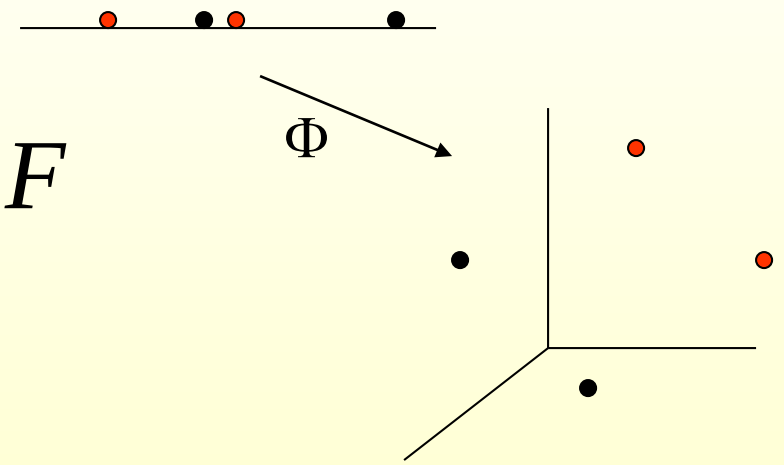
$$(x - a)^2 + (y - b)^2 \leq r^2$$

$$-2ax - 2by + x^2 + y^2 + a^2 + b^2 - r^2 \leq 0$$

Feature Spaces

- Map data to a high-dimensional feature space *F* via a non-linear mapping

$$\Phi : x \rightarrow \Phi(x), \quad R^d \rightarrow F$$



- Ex. $\Phi(x, y) = (x, y, x^2 + y^2)$

The Kernel Trick

- In many applications, we only need inner products (the Gram matrix) between mapped vectors for the solution
- We do not need to compute $\Phi(x)$ directly!

kernel

$$x \to \Phi(x),$$

$$G_{ij} =< x_i, x_j > \to G_{ij}^{\Phi} =< \Phi(x_i), \Phi(x_j) >= K(x_i, x_j)$$

- If we use algorithms that only depend on the Gram-matrix *G*, then we never have to know (compute) the actual features Φ

Modularity

Kernel methods consist of two modules:

1) The choice of kernel (this is non-trivial)
2) The algorithm which takes kernels as input

Modularity: Any kernel can be used with any kernel-algorithm.

some kernels:

$$k(x,y) = e^{(-\|x-y\|^2/c)}$$ RBF

$$k(x,y) = (< x,y > +\theta)^d$$

$$k(x,y) = \tanh(\alpha < x,y > +\theta)$$

$$k(x,y) = \frac{1}{\sqrt{\| x-y \|^2 + c^2}}$$

some kernel algorithms:

- support vector machines
- Fisher discriminant analysis
- kernel regression
- kernel PCA
- kernel CCA

Kernel PCA

Kernel PCA (KPCA)

- Assumption behind PCA is that the data points **x** are well represented by a small-dimensional Euclidean subspace
- Often this assumption does not hold …
- However, it may still be possible that a non-linear transformation $\phi(\mathbf{x})$ "linearizes" the data -- then we can perform PCA in the space of $\phi(\mathbf{x})$
- Kernel PCA performs this "lifted" PCA; however, because of "kernel trick," it never computes the mapping $\phi(\mathbf{x})$ explicitly.

KPCA Basic Idea

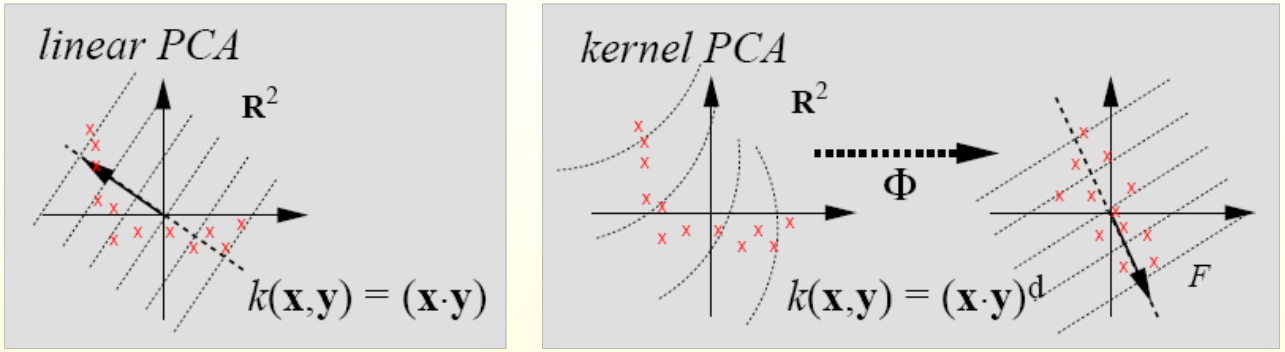


Fig. 1. Basic idea of kernel PCA: by using a nonlinear kernel function k instead of the standard dot product, we implicitly perform PCA in a possibly high–dimensional space F which is nonlinearly related to input space. The dotted lines are contour lines of constant feature value.

Kernel PCA Formulation

- We need the following fact: $C = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$
- Let **v** be a eigenvector of the covariance matrix: $C = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$

  scatter matrix
- Then **v** belongs to the linear space spanned by the data points $\mathbf{x}_i$ $I = 1, 2, \ldots N$.
- Proof:

$$C\mathbf{v} = \lambda\mathbf{v} \Rightarrow \mathbf{v} = \frac{1}{\lambda}\sum_{i=1}^{N}(\mathbf{x}_i \mathbf{x}_i^T)\mathbf{v} = \frac{1}{\lambda}\sum_{i=1}^{N}\mathbf{x}_i(\mathbf{x}_i^T \mathbf{v}) = \sum_{i=1}^{N}\alpha_i \mathbf{x}_i$$

Kernel PCA Formulation II

- Let C be the covariance matrix of the centered mapping $\phi(\mathbf{x})$:

$$C = \sum_{i=1}^{N} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T$$

- Let $\mathbf{w}$ be an eigenvector of C, then $\mathbf{w}$ can be written as a linear combination: $\mathbf{w} = \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)$

$$C\mathbf{w} = \lambda\mathbf{w}$$

- Also, we have:

$$\left(\sum_{i=1}^{N} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\right)\left(\sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)\right) = \lambda \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)$$

- Combining, we get:

Kernel PCA Formulation III

$$(\sum_{i=1}^{N} \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T)(\sum_{k=1}^{N} \alpha_k \varphi(\mathbf{x}_k)) = \lambda \sum_{k=1}^{N} \alpha_k \varphi(\mathbf{x}_k) \Rightarrow$$

$$\sum_{i=1}^{N}\sum_{k=1}^{N} \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_k)\alpha_k = \lambda \sum_{k=1}^{N} \alpha_k \varphi(\mathbf{x}_k) \Rightarrow$$

$$\sum_{i=1}^{N}\sum_{k=1}^{N} \varphi(\mathbf{x}_l)^T \varphi(\mathbf{x}_i)\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_k)\alpha_k = \lambda \sum_{k=1}^{N} \alpha_k \varphi(\mathbf{x}_l)^T \varphi(\mathbf{x}_k),\ l = 1, 2, \ldots, N \Rightarrow$$

$$K^2\boldsymbol{\alpha} = \lambda K\boldsymbol{\alpha} \Rightarrow$$

$$K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha},\ \text{where}\ K_{ij} = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j).$$

Kernel or Gram matrix

Kernel PCA Formulation IV

From the eigen equation $K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$

And the fact that the eigenvector **w** is normalized to 1, we obtain:

$$\| \mathbf{w} \|^2 = (\sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i))^T (\sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)) = \boldsymbol{\alpha}^T K \boldsymbol{\alpha} = 1 \Rightarrow$$

$$\boldsymbol{\alpha}^T \boldsymbol{\alpha} = \frac{1}{\lambda}$$

The KPCA Algorithm

Step 1: Compute the Gram matrix: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j),\ i, j = 1, \ldots, N$

Step 2: Compute (eigenvalue, eigenvector) pairs of K : $\mathbf{w}^l, \lambda_l$

Step 3: Normalize the eigenvectors: $\tilde{\boldsymbol{\alpha}}^l \leftarrow \frac{\boldsymbol{\alpha}^l}{\lambda_l}$

Thus, an eigenvector $\mathbf{w}^l$ of C is now represented as: $\mathbf{w}^l = \sum_{k=1}^{N} \alpha_k^l \varphi(\mathbf{x}_k)$

To project a test feature $\phi(\mathbf{x})$ onto $\mathbf{w}^l$ we need to compute:

$$\phi(\mathbf{x})^T \mathbf{w}^l = \phi(\mathbf{x})^T \left(\sum_{k=1}^{N} \alpha_k^l \phi(\mathbf{x}_k)\right) = \sum_{k=1}^{N} \alpha_k^l k(\mathbf{x}_k, \mathbf{x})$$

So, we never need ϕ explicitly

Feature Map Centering

So far we assumed that the feature map $\phi(\mathbf{x})$ is centered for the data points $\mathbf{x}_1, \ldots \mathbf{x}_N$

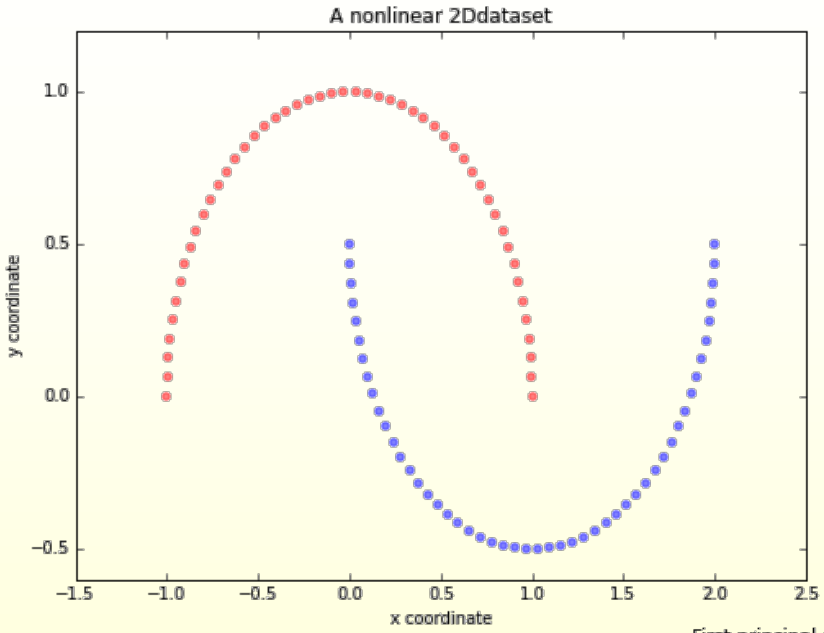
Actually, this centering can be done on the Gram matrix without ever explicitly computing the feature map $\phi(\mathbf{x})$.

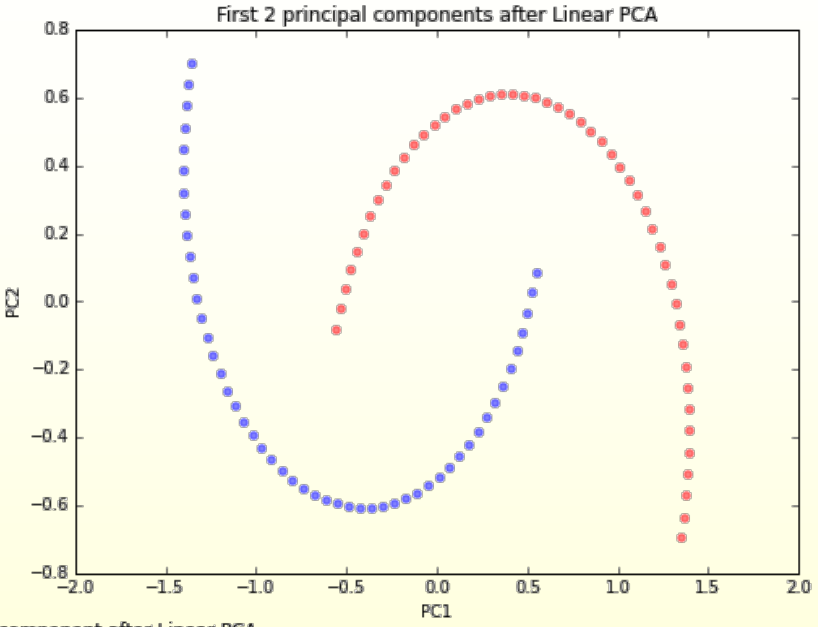
$$\tilde{K} = (I - 11^T / N) K (I - 11^T / N)$$

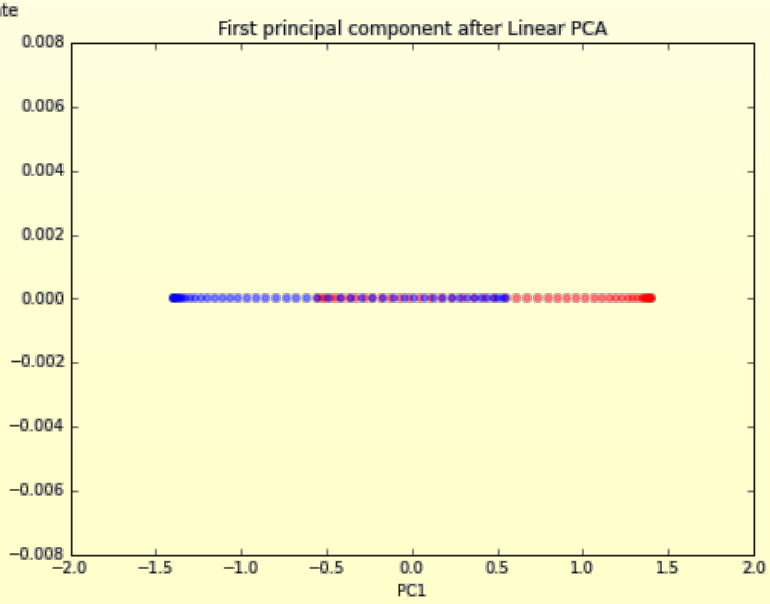
is the kernel matrix for centered features, *i.e.*, $\sum_{i=1}^{N} \phi(\mathbf{x}_i) = 0$

A similar expression exists for projecting test features on the feature eigenspace

KPCA Example

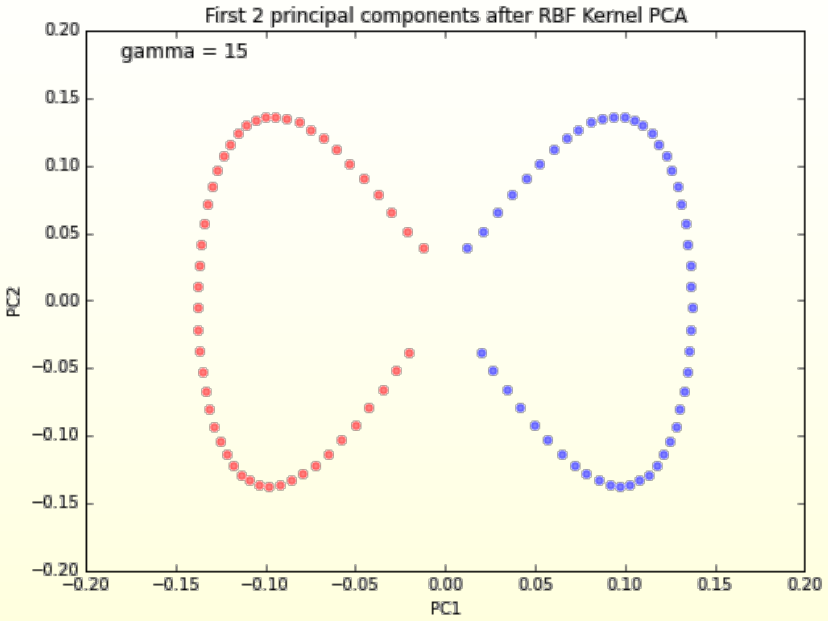


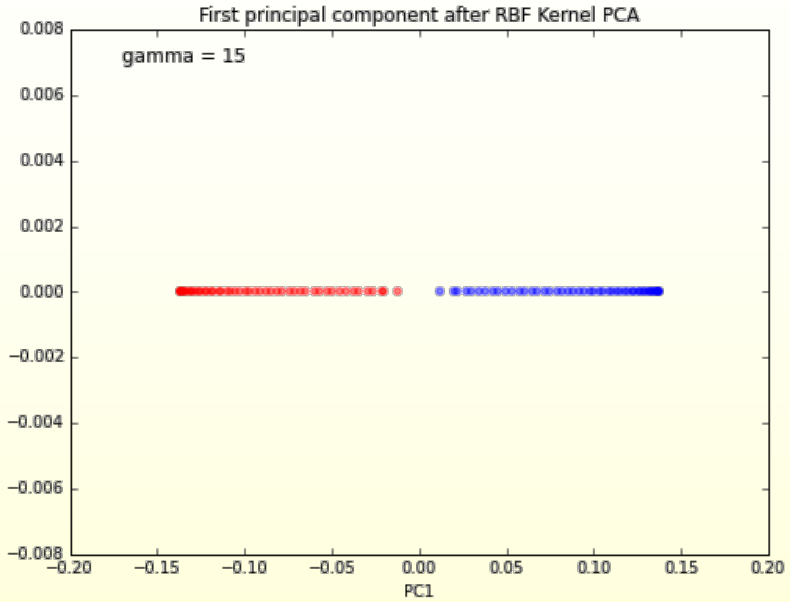




With linear PCA

KPCA Example





With kernel PCA,
using RBF kernel

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = exp(-\gamma \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2)$$

The End