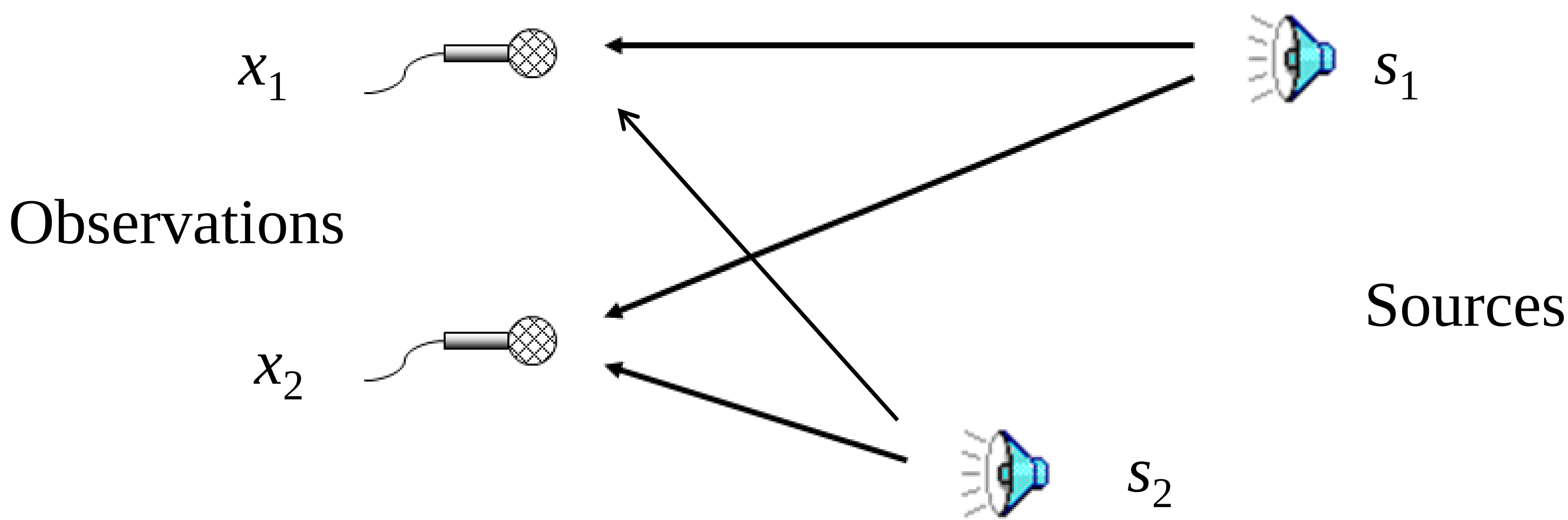
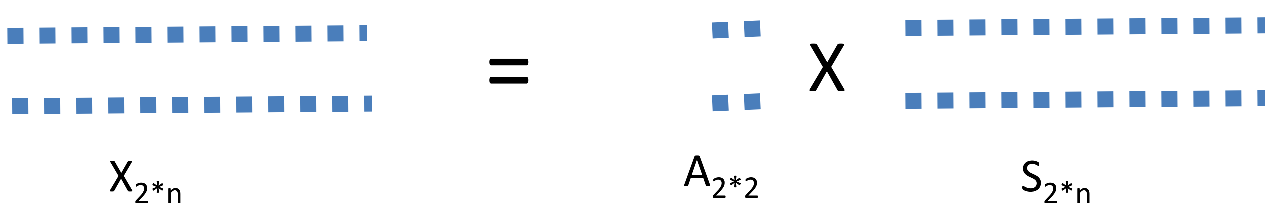
ICA

ICA: 2-D examples



$$\mathbf{x} = \mathbf{A}\mathbf{s}$$



Independent Components Analysis

$$X_1 = a_{11}S_1 + a_{12}S_2 + \ldots + a_{1p}S_p$$

$$X_2 = a_{21}S_1 + a_{22}S_2 + \ldots + a_{2p}S_p$$

$$\vdots$$

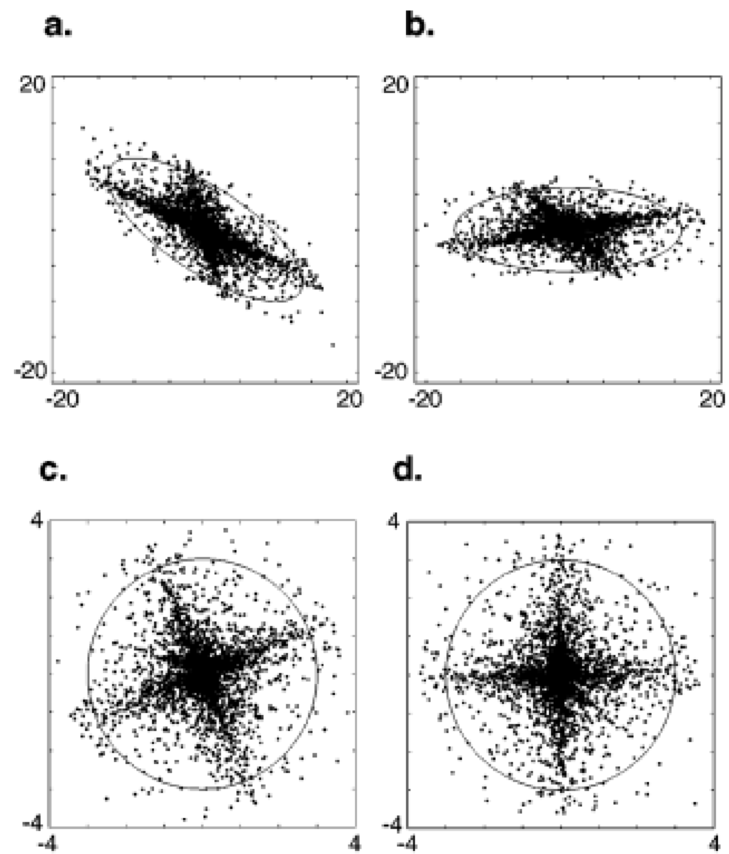
$$X_p = a_{p1}S_1 + a_{p2}S_2 + \ldots + a_{pp}S_p$$

$$X = AS$$

If we knew A we could solve for the sources S
But we have to solve for *both*

We will look for a solution that will make S *independent*

PCA and ICA



$X = AS$

- Getting a simpler form

- We can always express A by SVD as $U\Sigma V^T$
- U and V are orthonormal and Σ is diagonal
- (we don't know any of them)
- So now $X = U\Sigma V^T S$

- Taking the covariance matrix of the data:
- $XX^T = U\Sigma V^T S \; S^T V \Sigma U^T$
- We can assume that $SS^T = I$
- They are independent, therefore uncorrelated.
- We can assume all of length = 1
- This is just scaling; we can scale S and A

- $X = AS$
- $A = U\Sigma V^T$ (the SVD of A)
- $X = U\Sigma V^T S$

- $XX^T = U\Sigma V^T S \; S^T V \Sigma U^T$ with $SS^T = I$
- $XX^T = U\Sigma^2 U$

With the *same* U, Σ we used for A above

- XX^T is known, so we can find the U, Σ of A from the data
- (by diagonalizing $XX^T = U \Lambda U^T$)

ICA procedure

- Looking for X = AS with S independent

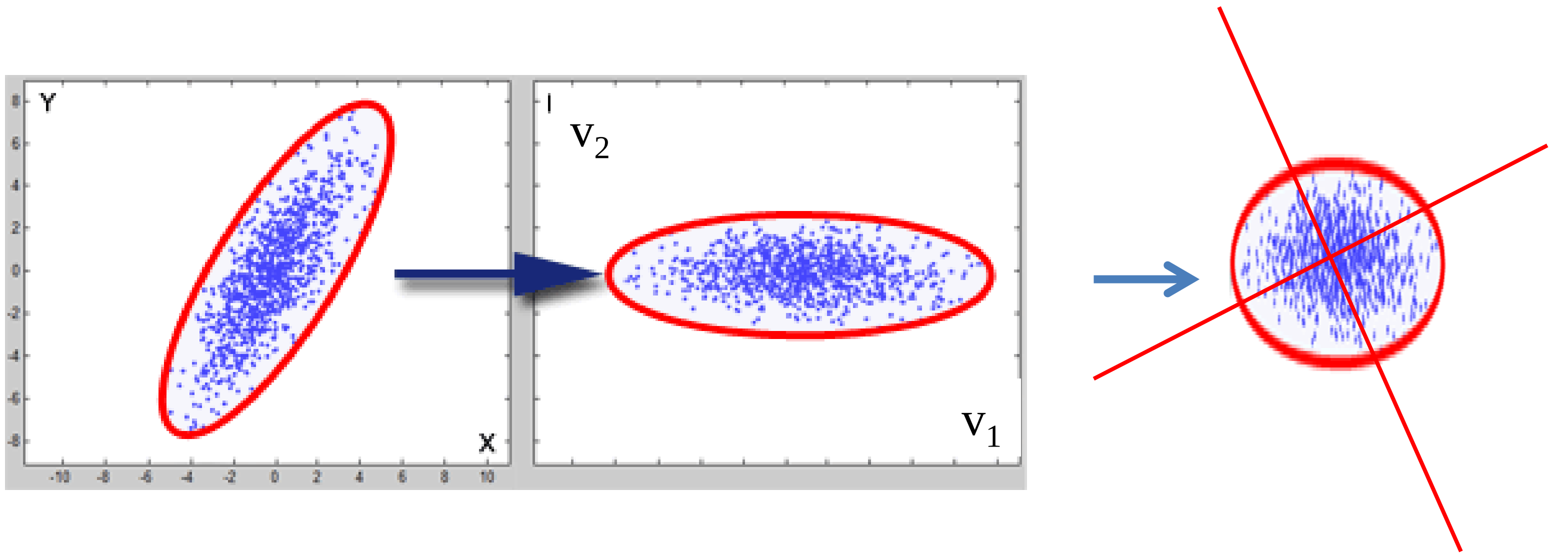
- Start by whitening X:
- Do PCA, then: $X' \leftarrow \Sigma^{-1} U^T X$

- In the new data solve for X' = VS
- Both V,S unknown, but V is rotation, and S are independent.

- Search over rotations and test for independence

- For a given V, S is easy to obtain, we need some measure of independence

Whitening the data



Perform PCA
Re-scale the coordinates by their variance

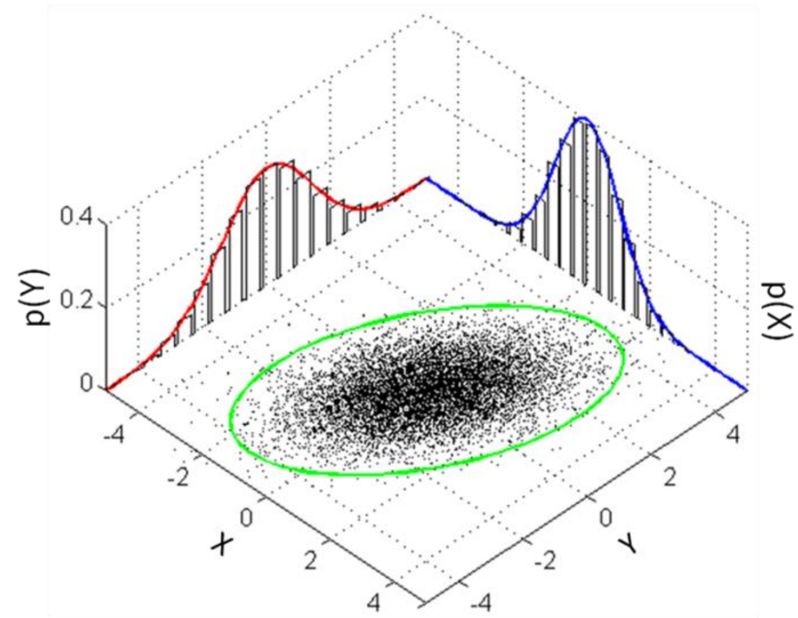
ICA: Final step – look for rotation that will make S as independent as possible

Testing for Independence

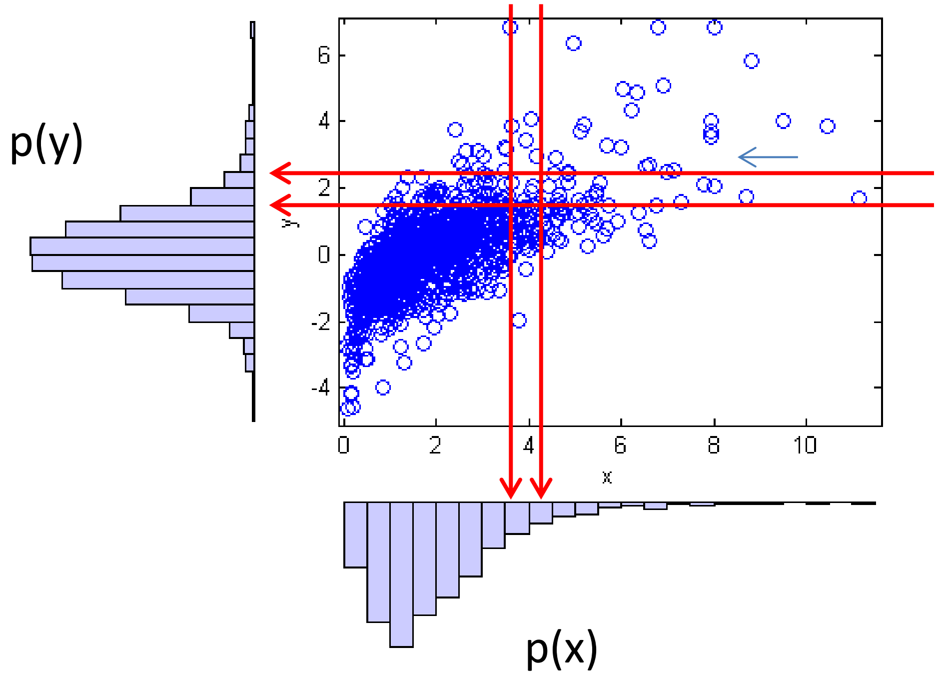
- Suppose that a source produces variables $(x_1\ y_1)$ $(x_2\ y_2)$..
- It is straightforward to test if they are correlated or not by $\Sigma x_i y_i = 0$
- In practice, $\Sigma x_i y_i > \varepsilon$
- How to test independence?

- Several methods, describe briefly one.

1-D projection



Testing independence

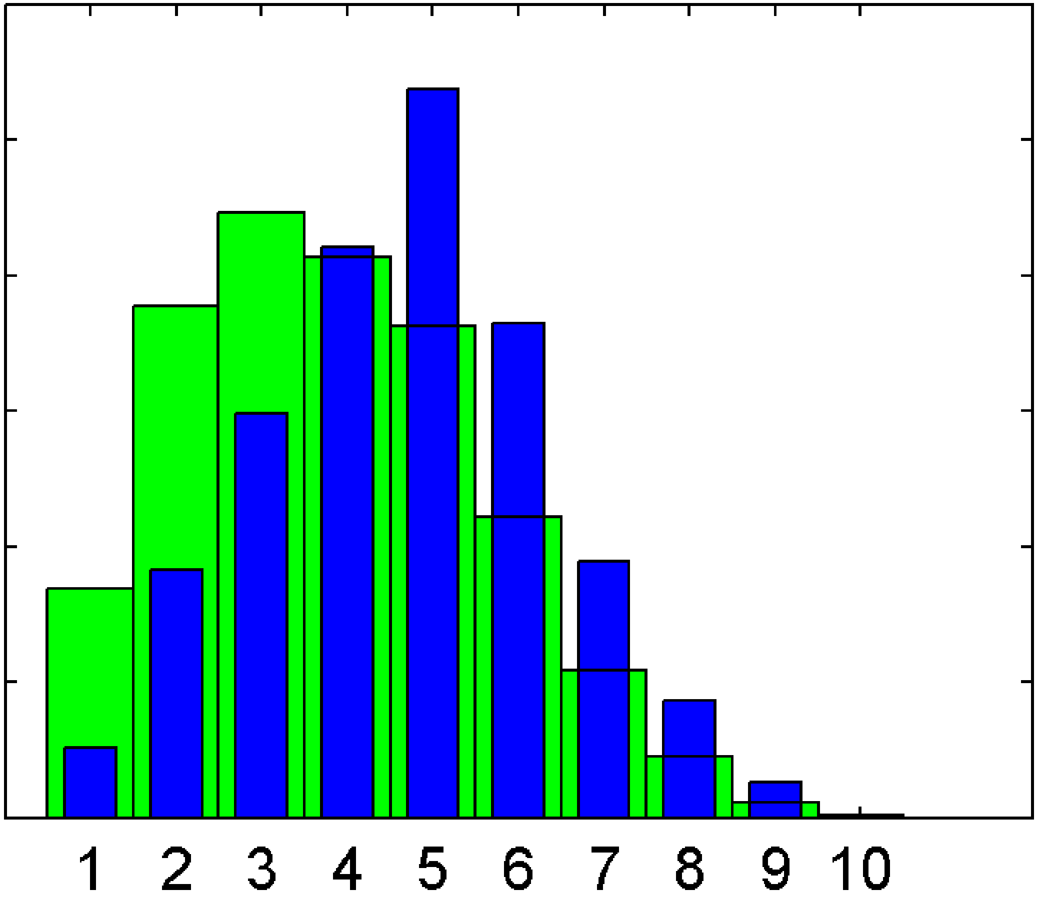


$$p(x,y) = p(x)\, p(y)$$

- In principle for each pair x_i y_j verify that $p(x_i\ y_j) = p(x_i)\ p(y_i)$
- We have many pairs, how to use them together in an efficient test
- We look at the two distributions $p(x,y)$ and $q(x,y) = p(x)p(y)$
- We want to test if they the same (or very close)

- How to compare two distributions?

Two distributions – how different are they?

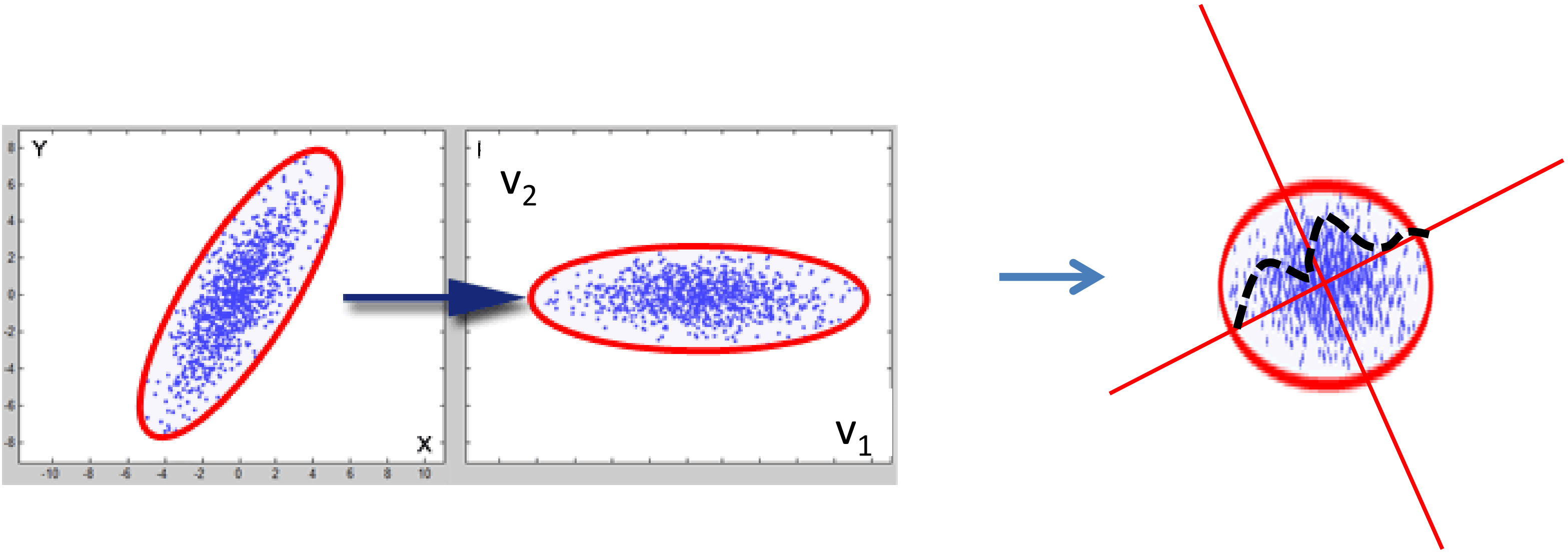


Testing for Independence

- Use the KL divergence: Kullback-Leibler
- KL(p||q) = Σ [p log (p/q)]
- Non-negative, it is 0 only iff they are the same.

- In our case
- KL [p(x y) || p(x) p(y)] = Σ [p(x y) log (p(x,y)/p(x) p(y))] =
- Σp(x,y) log p(x,y) - (Σp(x,y) log p(x) + Σp(x,y) log p(y))
- = -H(p(x,y)) +[H(p(x)) + H(p(y))]
-
- ΣH_i - H

- H is constant, minimize ΣH_i (marginal distribution after rotation)



Final step: optimize iteratively over rotation. For each rotation project the data on the axes and measure Hi of the projections.

Technical difficulties:

- Minimizing ΣH_i on all the axes
- Non-convex, complex, minimization

- Estimating entropy H, requires enough samples, sensitive to outliers

- Various algorithms to optimize the numeric process

- FastICA (Hyvärinen), Proceeds one component at a time, then combines them

Equivalent Criterion

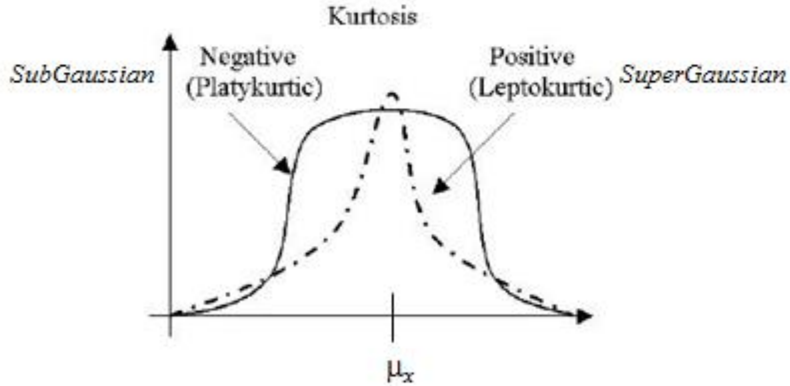
- Rotation that maximizes $H - \Sigma H_i$ also maximizes the “non-Gaussianity” of the transformed data.
-
- Non-Gaussianity (‘negentropy’): as the Kullback-Leibler divergence of a distribution from a Gaussian distribution with equal variance.
-
- Non-gaussianity is also measured by Kurtosis
-
- Family of algorithms that maximize Kurtosis rather than marginal entropies

Kurtosis

Higher order moments (4^{th}-*kurtosis*)

$$\hat{\kappa}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}\left[\frac{x_i - \hat{\mu}_x}{\hat{\sigma}}\right]^4$$

Gaussians are *mesokurtic* with $\kappa = 3$



Non-Gaussianity: Kurtois should be far from 3

A family of algorithms that use Kurtosis rather than marginal entropies

On Whitening the Data

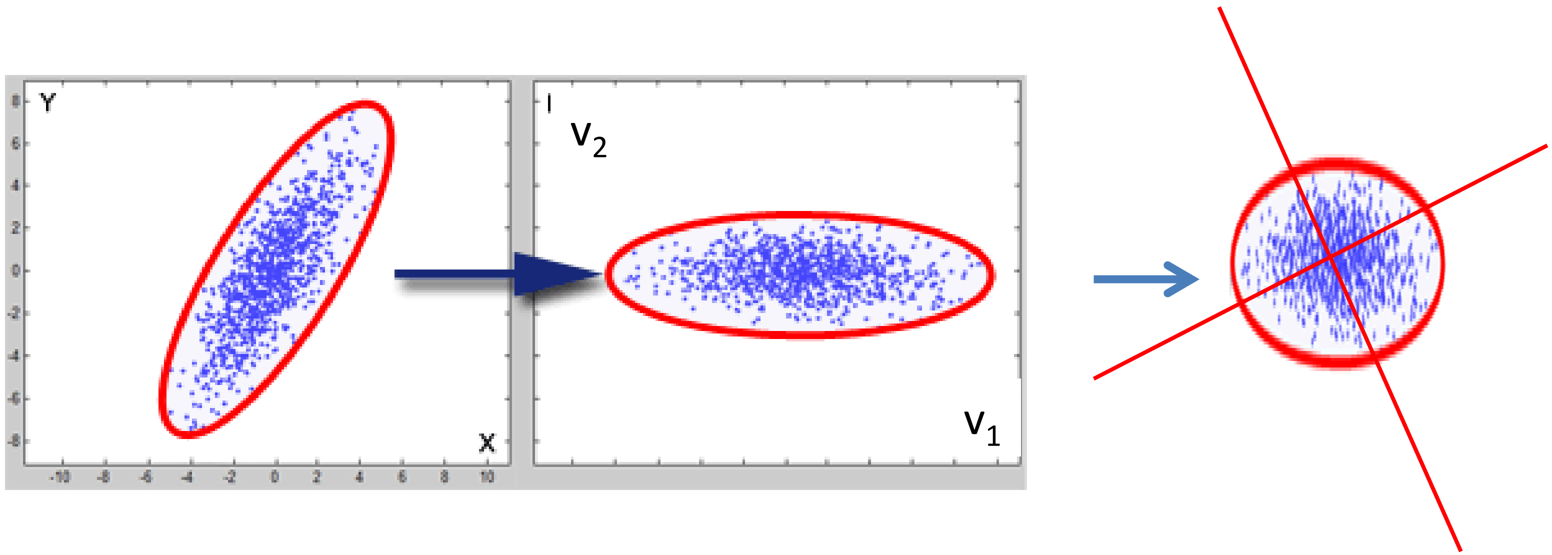
- An important step in general, additional comments:
- The data matrix XX^T can be expressed as: $U\Lambda U^T$
-
- Whitening X is:
- $X_W = \Lambda^{-1/2} U^T X$
-
- We can check:
- $X_W X_W^T = \Lambda^{-1/2} U^T X \; X^T U \Lambda^{-1/2}$
-
- Substituting XX^T
-
- $\Lambda^{-1/2} U^T \; U\Lambda U^T \; U \Lambda^{-1/2} = I$

On Whitening the Data

- Whitening: $X_W = \Lambda^{-1/2}\, U^T X$

- *Regularization:*
- $\Lambda^{-1/2}$ is a diagonal matrix with 1/(sqrt λi) on the diagonal
- This is regularized to 1/(sqrt $\lambda_i + \varepsilon$)

- *ZCA (zero-phase whitening)*
-
- Whitening is non-unique.
- Any rotation will leave it whitened (next slide)
-
- Taking in particular U from the data matrix:
-
- $X_{ZCA} = U\, \Lambda^{-1/2}\, U^T X$
-
- From all whitened X_W, this is the closest to the original X.



After whitening, added rotation leaves the data whitened

Next: Performing the ICA on image patches:

- **The “independent components” of natural scenes are edge filters**
- Bell and Sejnowski Vision Research 1997