Introduction to Computer Vision

Michael J. Black

Oct 2009

Probability, PCA, covariance and classification

Reading

Szeliski

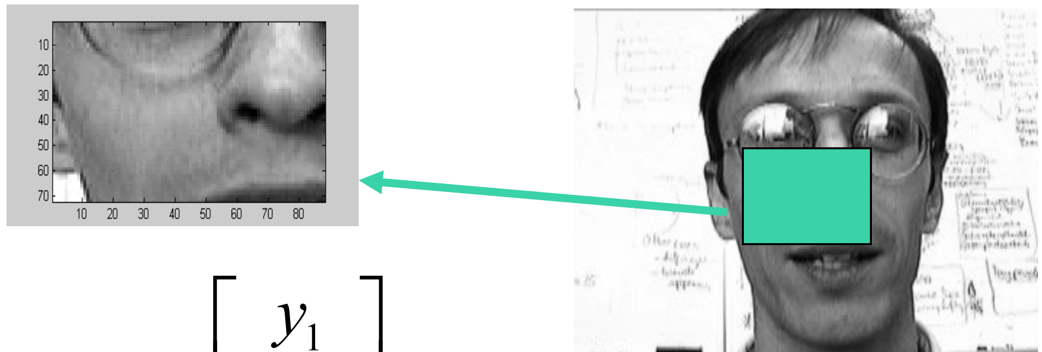
14.1, Face Recognition (including PCA)

A1.1 and 1.2, SVD and PCA

Goals

- Finish probability and classification
  - Everything you need assignment 2
- Wed/Fri: Motion and prep for assign 3

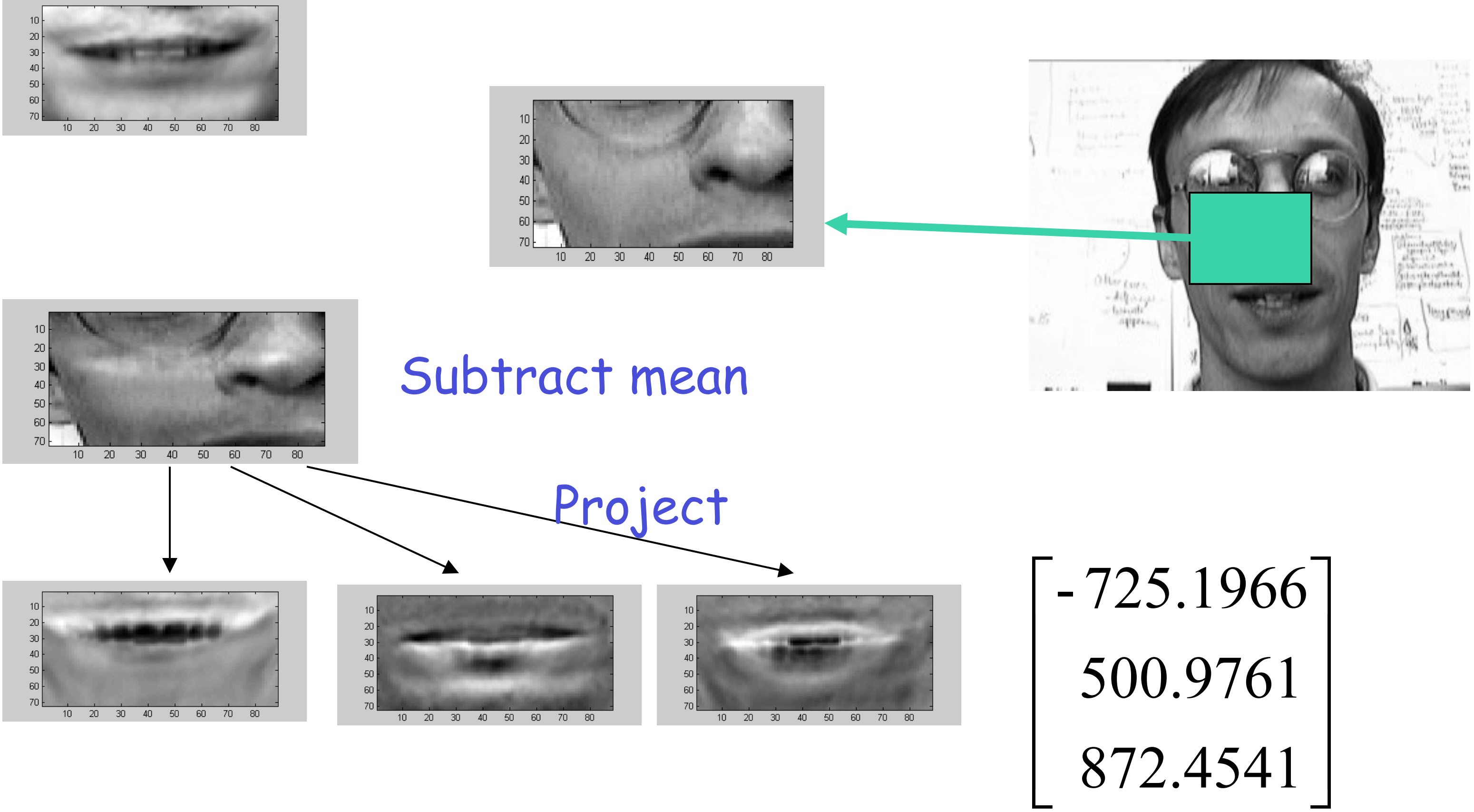
Images as Vectors



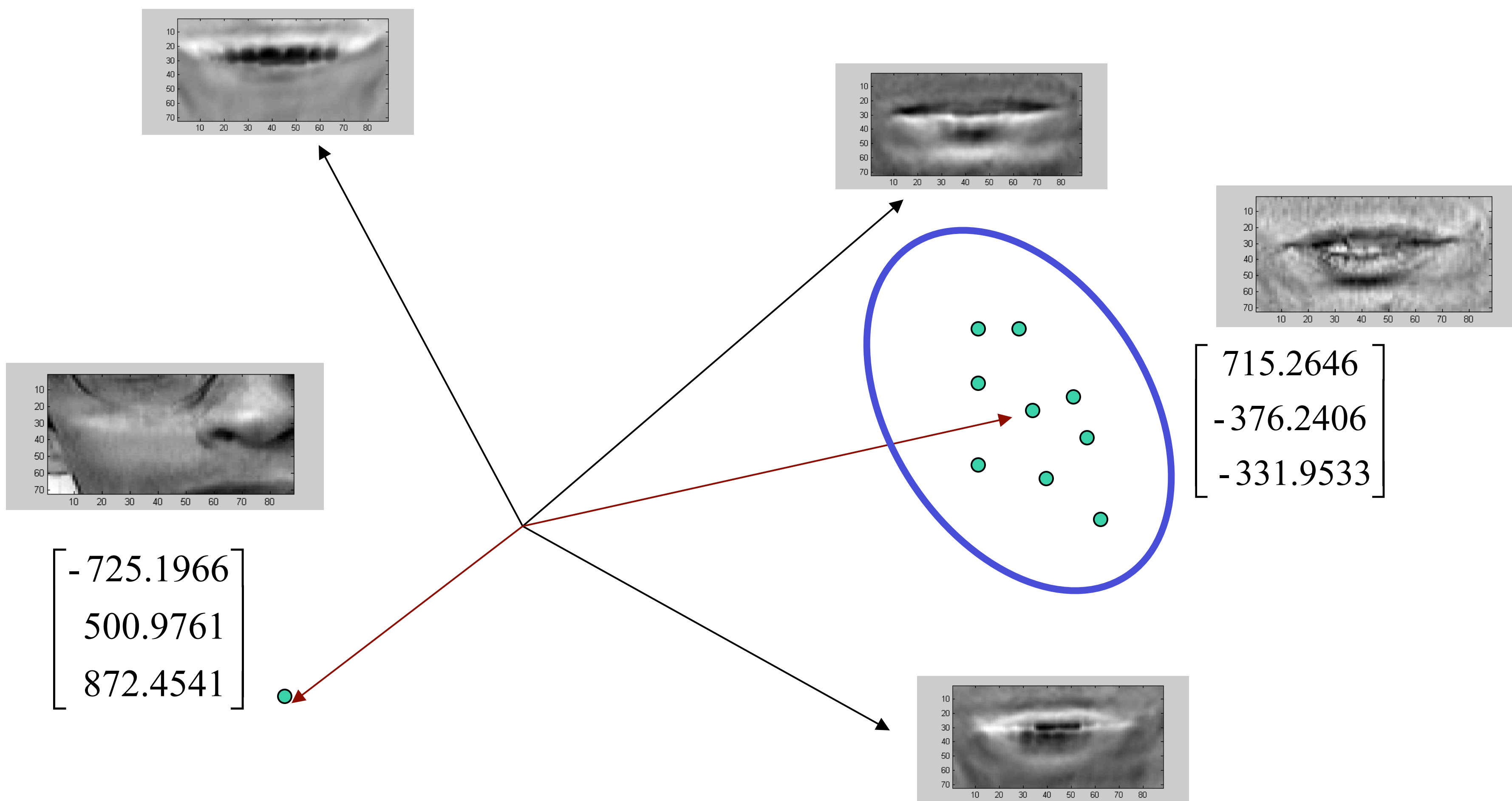
$$= \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n \times m} \end{bmatrix}$$

Is it a mouth?

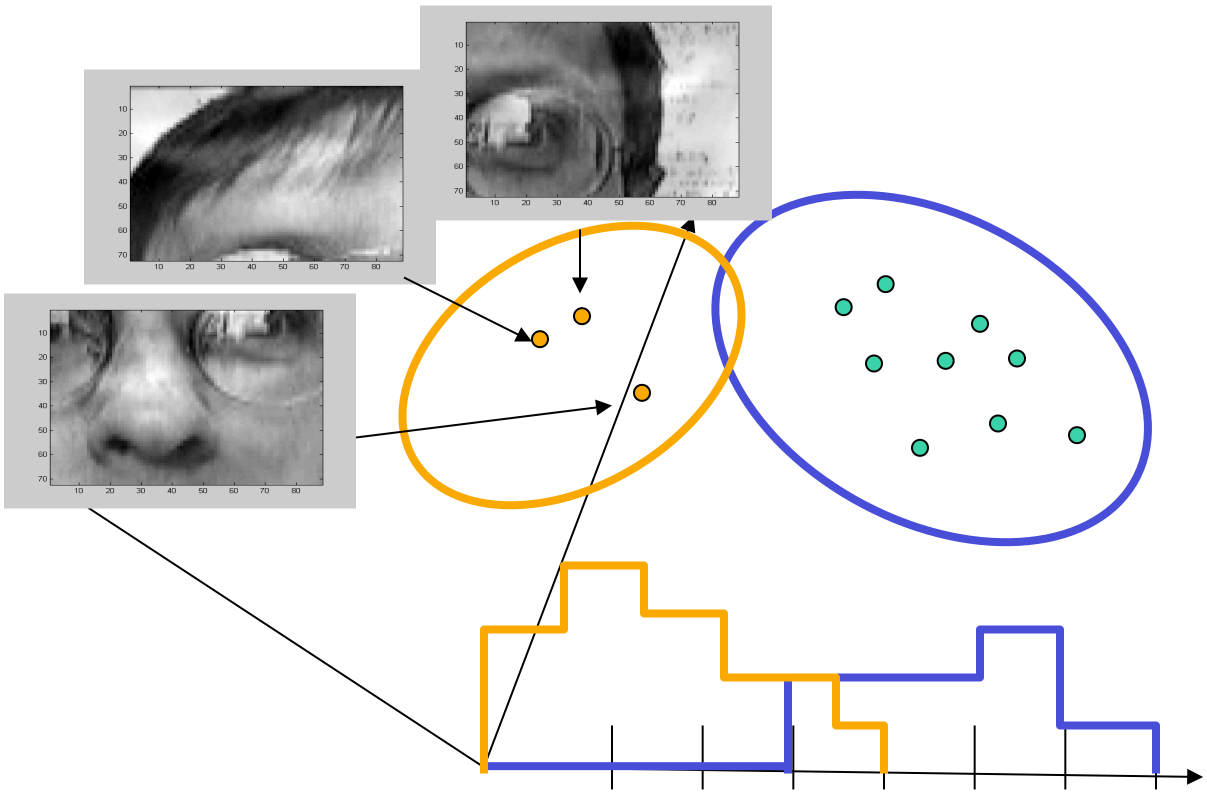
Images as Vectors



Mouth Space

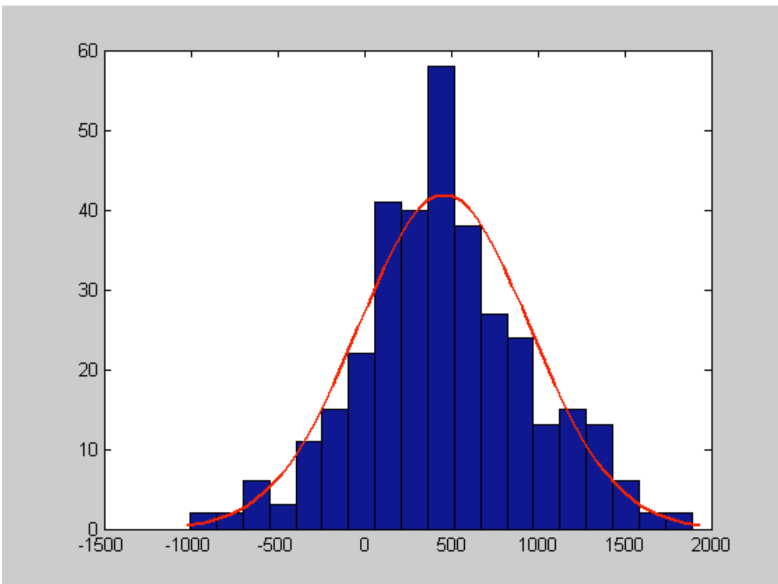


Classification

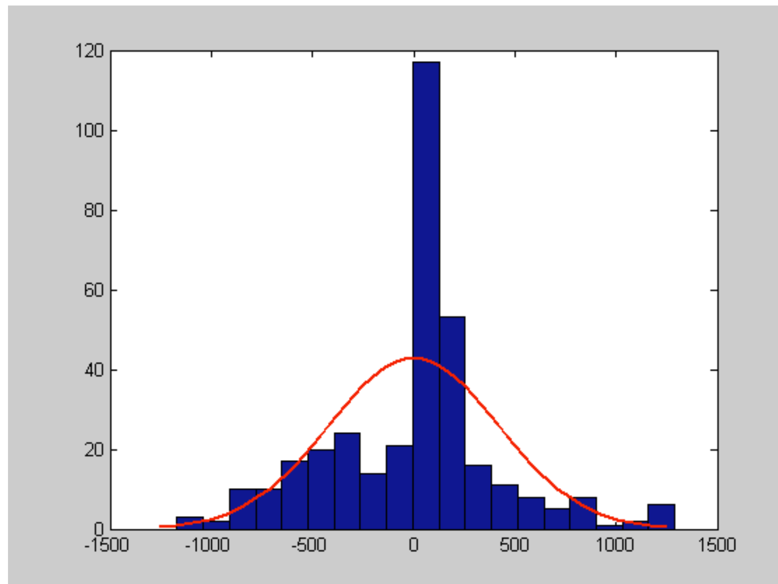


Classification

Imagine we just consider one dimension (one linear coefficient).

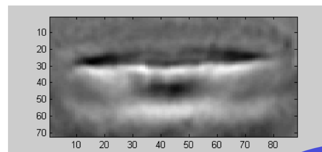


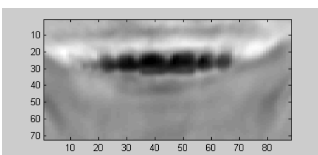
$p(a_3 \mid \neg \text{mouth})$

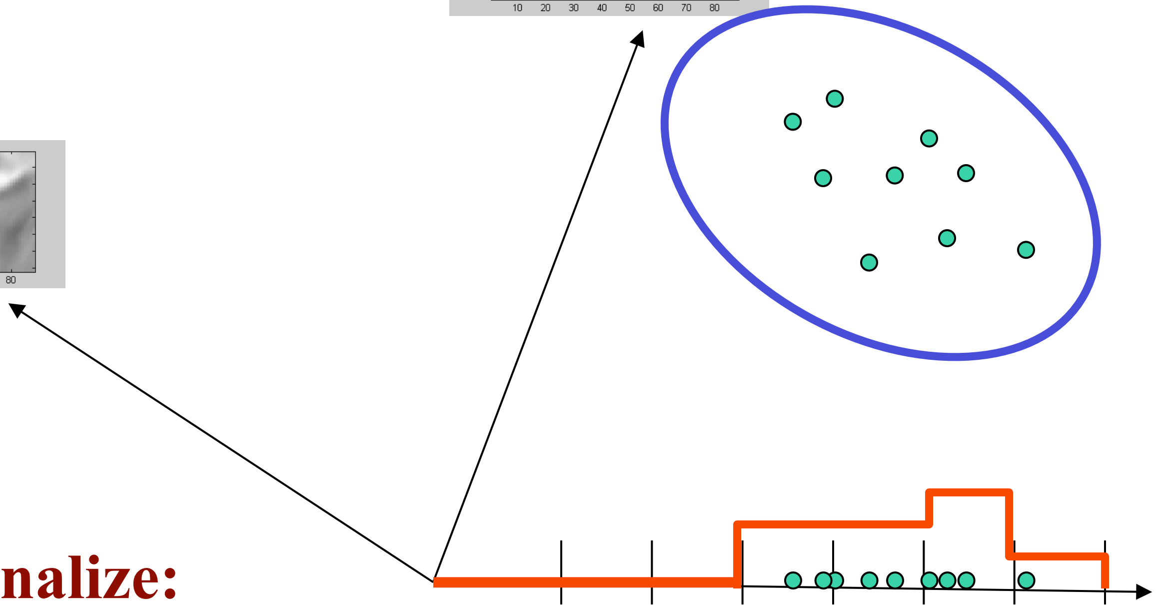


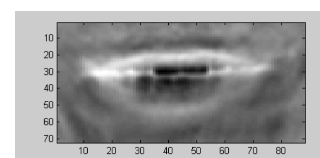
$p(a_3 \mid \text{mouth})$

Probabilistic Model









Marginalize:

$$\sum_{a_1, a_2} p(a_1, a_2, a_3 \mid \text{mouth}) = p(a_3 \mid \text{mouth})$$

Marginalization

$$p(a,b) = p(a \mid b)\, p(b)$$

$$p(a) = \sum_b p(a \mid b)\, p(b) = \sum_b p(a,b)$$

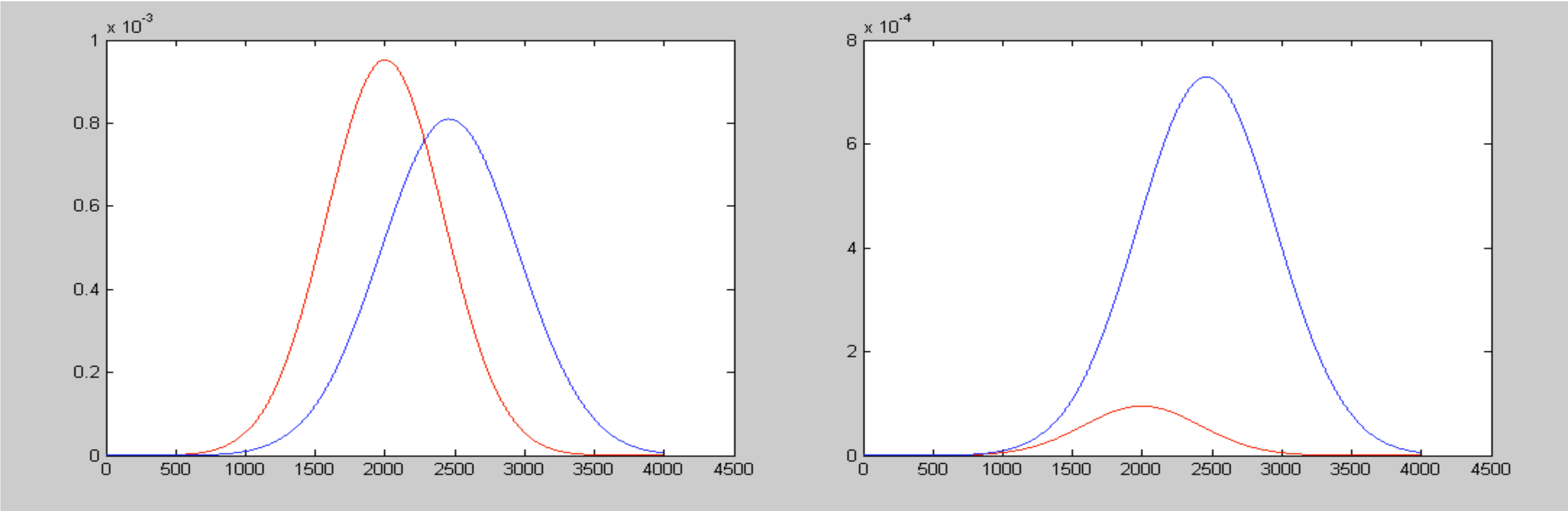
Posterior Probability

likelihood prior

$$p(\text{mouth} \mid a_3) = \frac{p(a_3 \mid \text{mouth})\, p(\text{mouth})}{p(a_3)}$$

normalization constant
(independent of mouth)

Maximum A Posteriori Classification



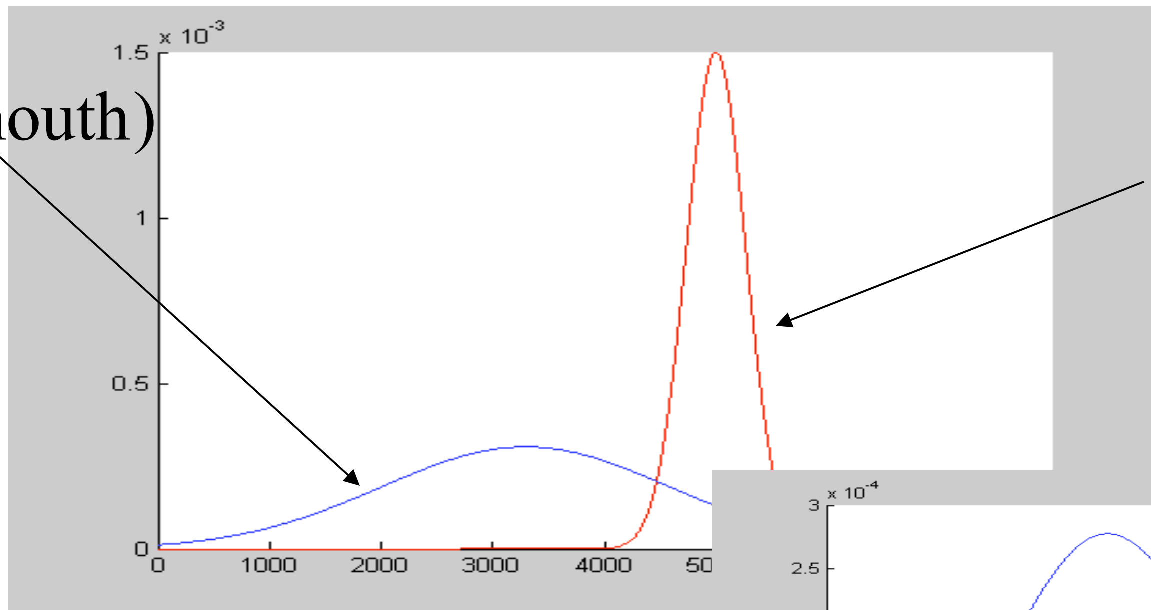
likelihood posterior

From a_3 alone, it looks like MAP classification will always prefer the not-mouth interpretation.

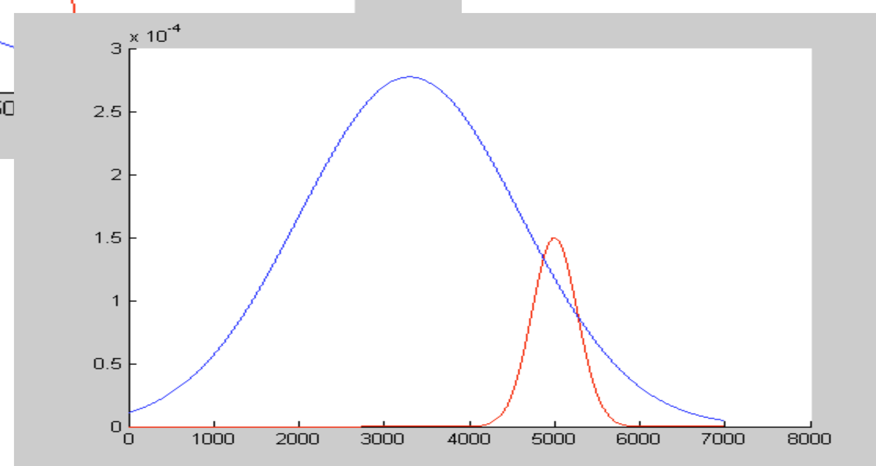
What about the other coefficients?

$p(a_6 \mid \neg \text{mouth})$

$p(a_6 \mid \text{mouth})$



posterior:



Conditional Independence

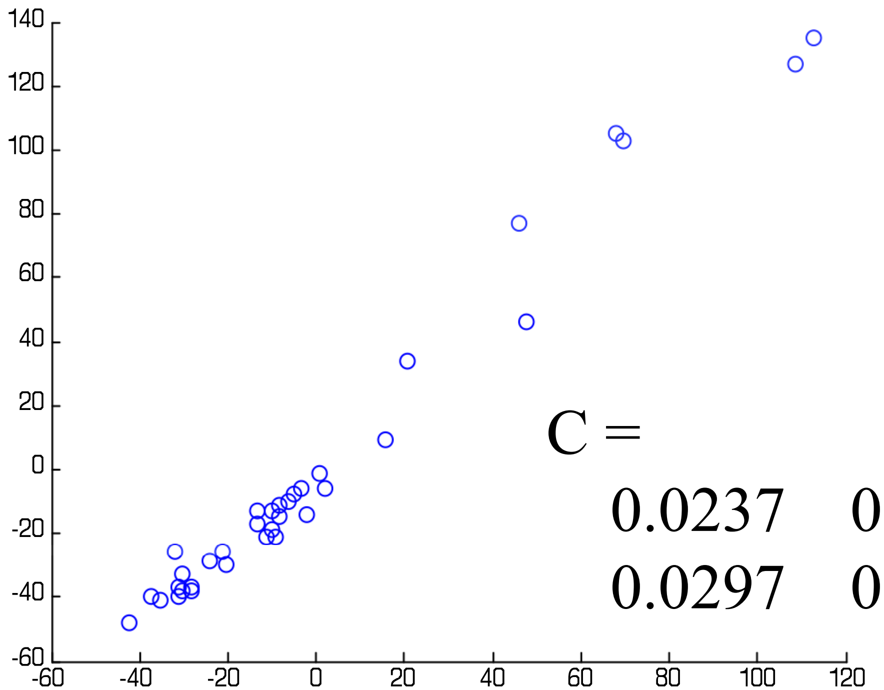
$$p(a_1, a_2, \ldots, a_M \mid \text{mouth}) = \prod_{i=1}^{M} p(a_i \mid \text{mouth})$$

Conditional Independence

$$p(a_1, a_2, \ldots, a_M \mid \text{mouth}) = \prod_{i=1}^{M} p(a_i \mid \text{mouth})$$

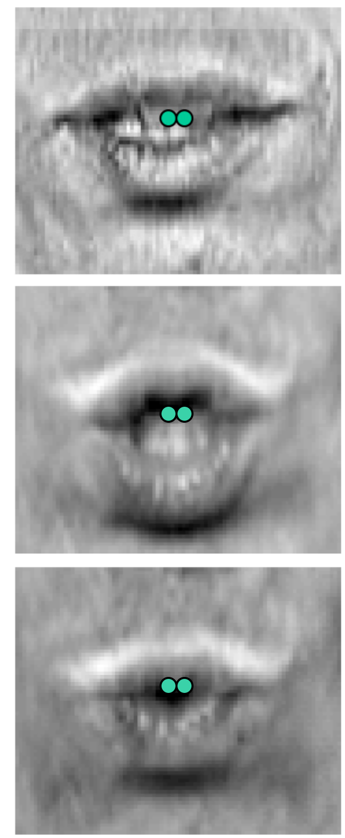
Where does this break?

Example: Covariance



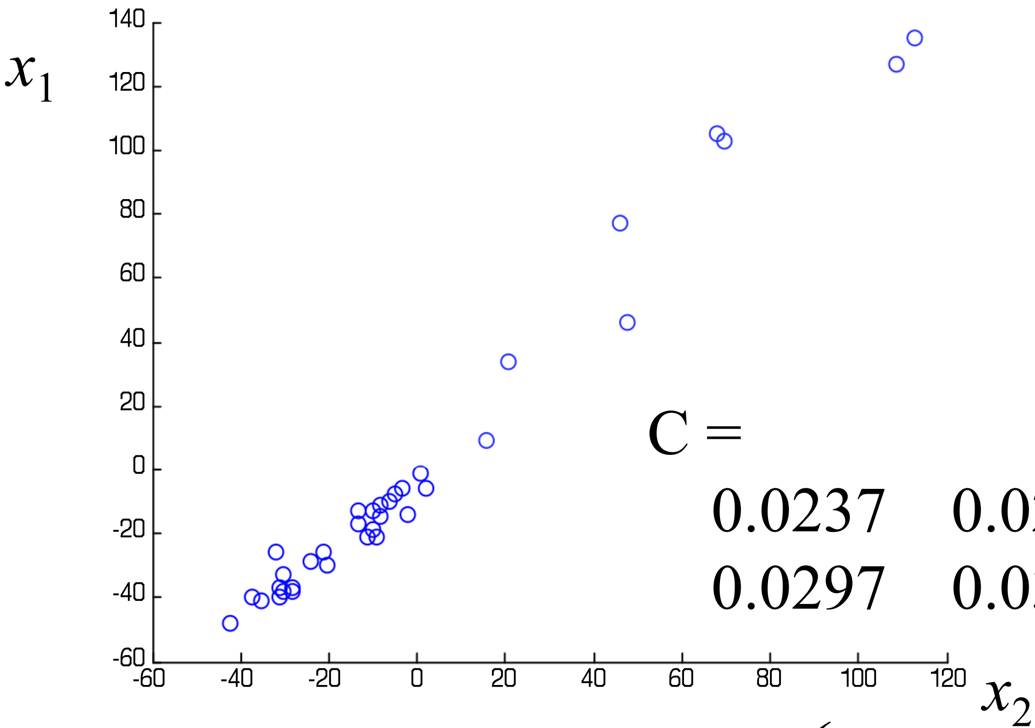
C =

0.0237	0.0297
0.0297	0.0384

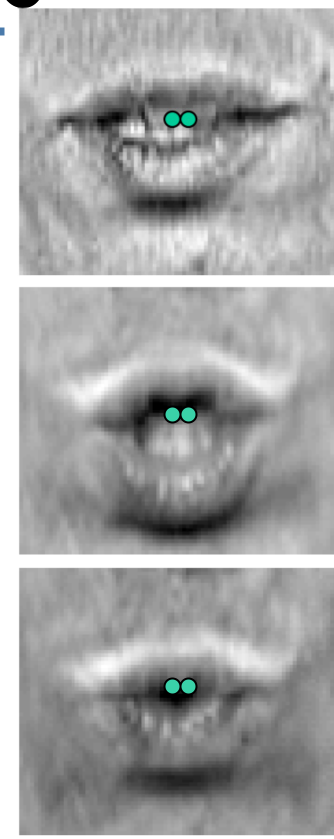


C = cov(A(:,30*88+46)/255, A(:,30*88+47)/255)

Example: Covariance



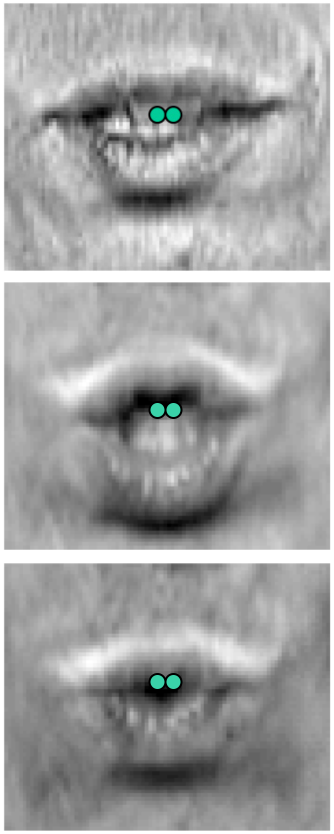
$$C = \begin{matrix} 0.0237 & 0.0297 \\ 0.0297 & 0.0384 \end{matrix}$$



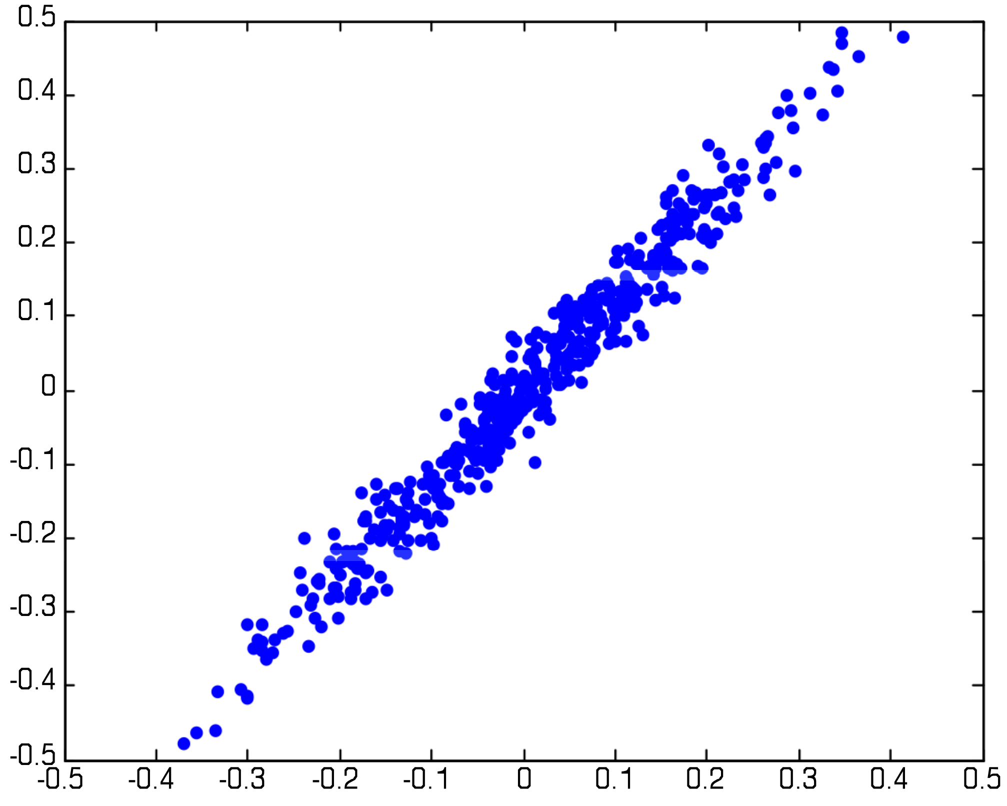
$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, |C|^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

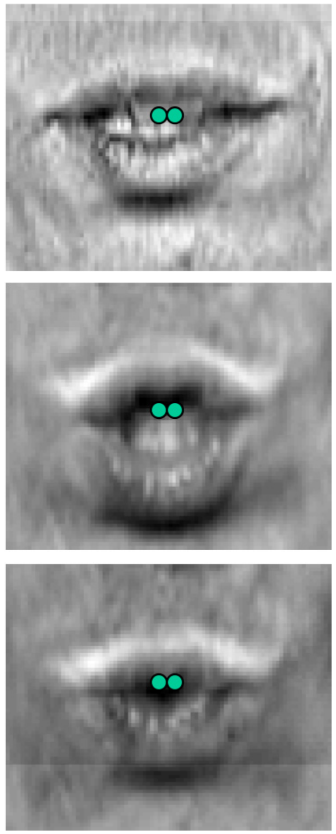
Example: Covariance

```
mu=[0 0];

% draw 500 samples from a multivariate
% Gaussian
r = mvnrnd(mu, C, 500);

plot(r(:,1), r(:,2), '.');
axis([-0.5 0.5 -0.5 0.5])
```



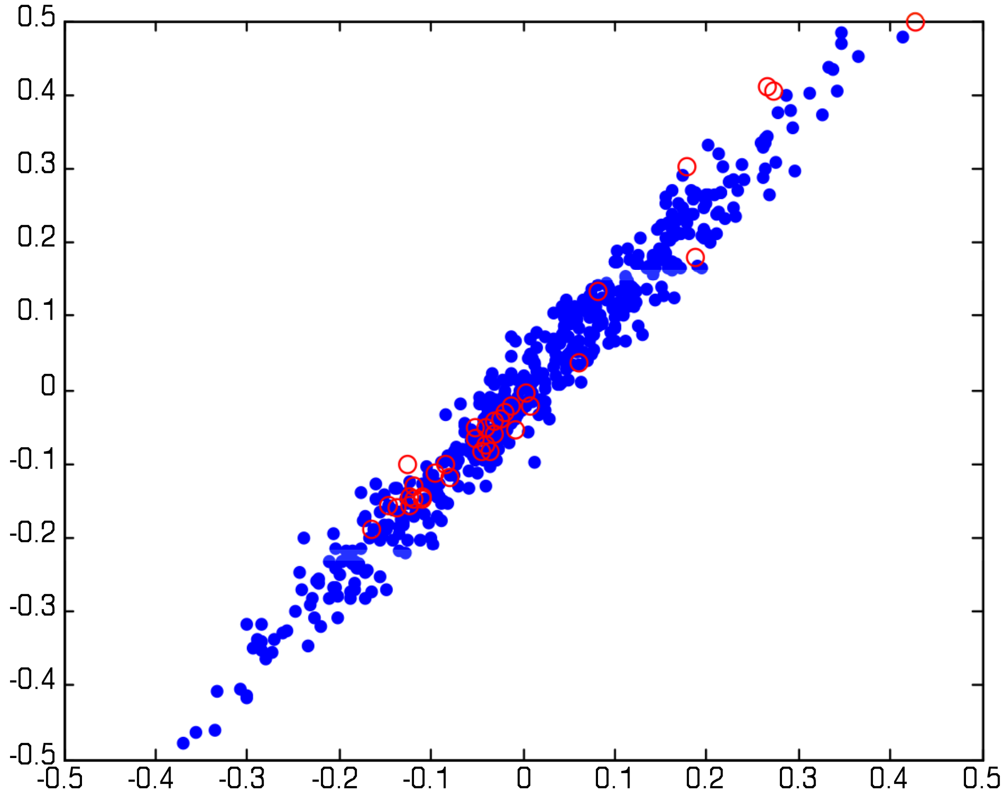
Example: Covariance

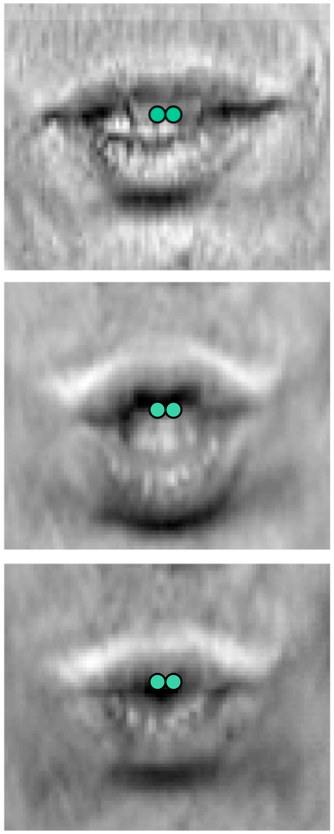




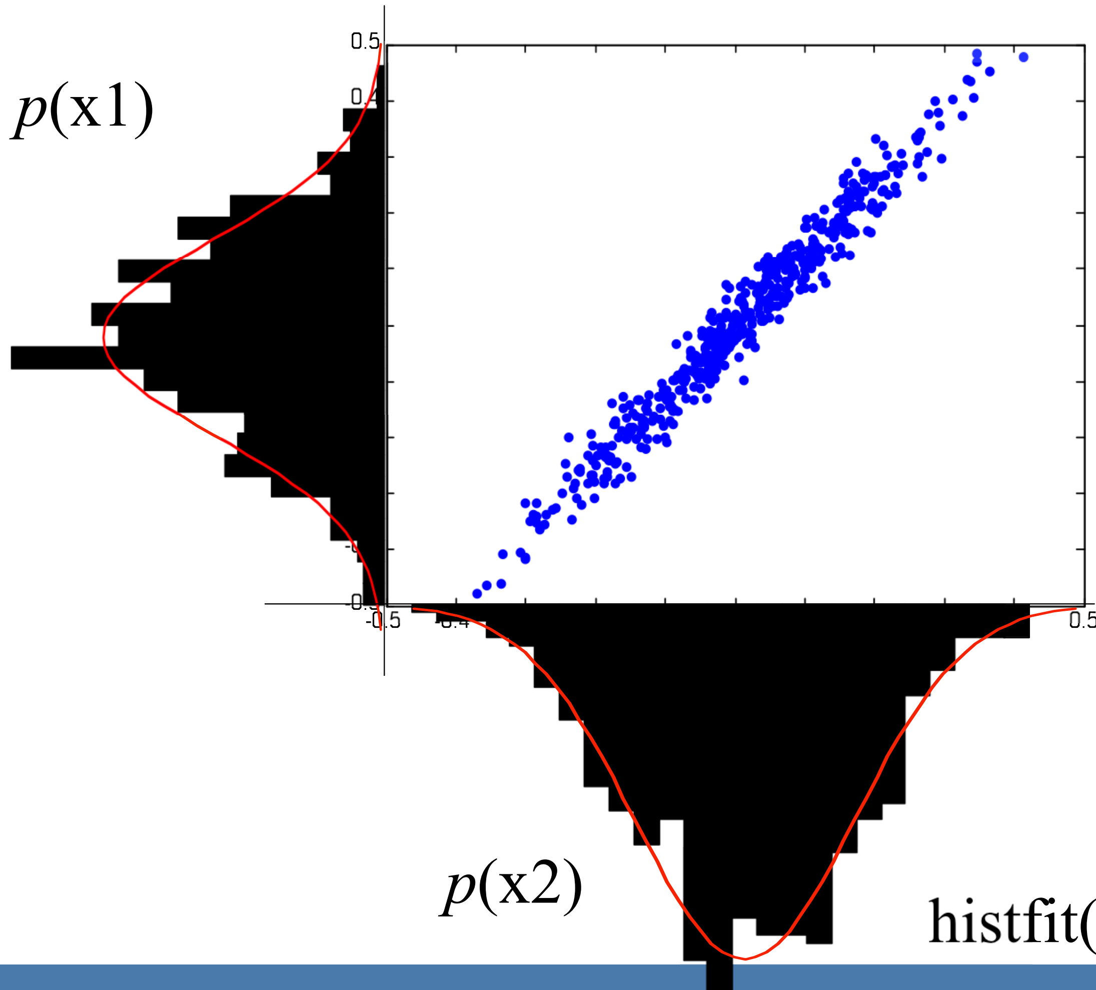
Samples from $p(x1, x2)$

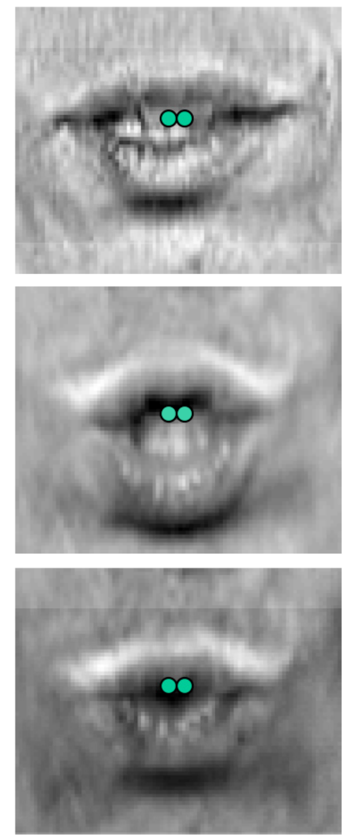
Example: Covariance





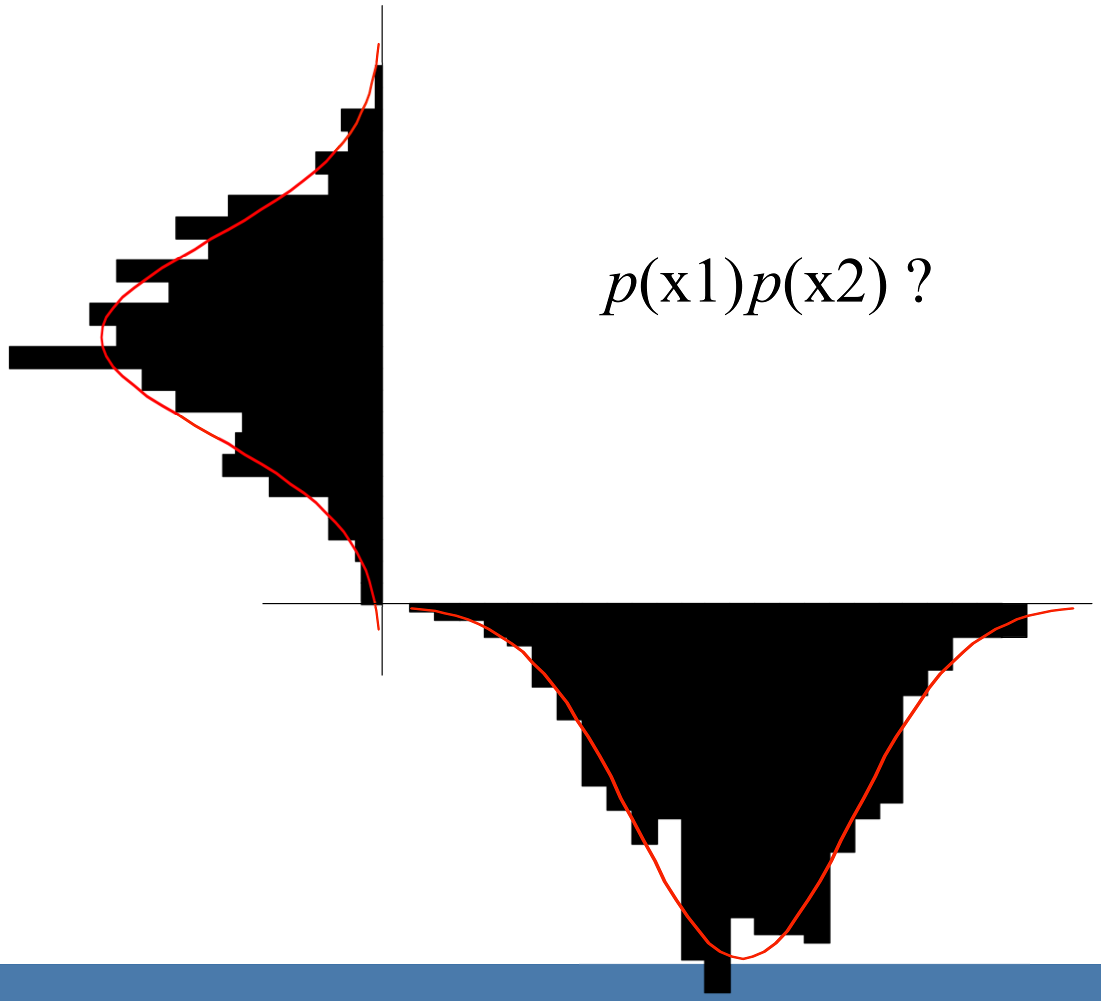
Marginals



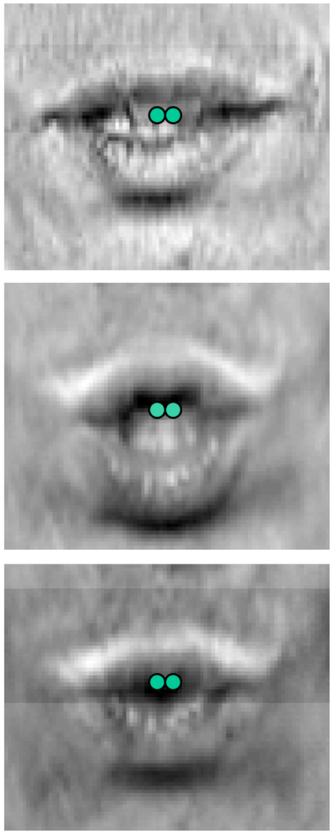


histfit(r(:,2),25)

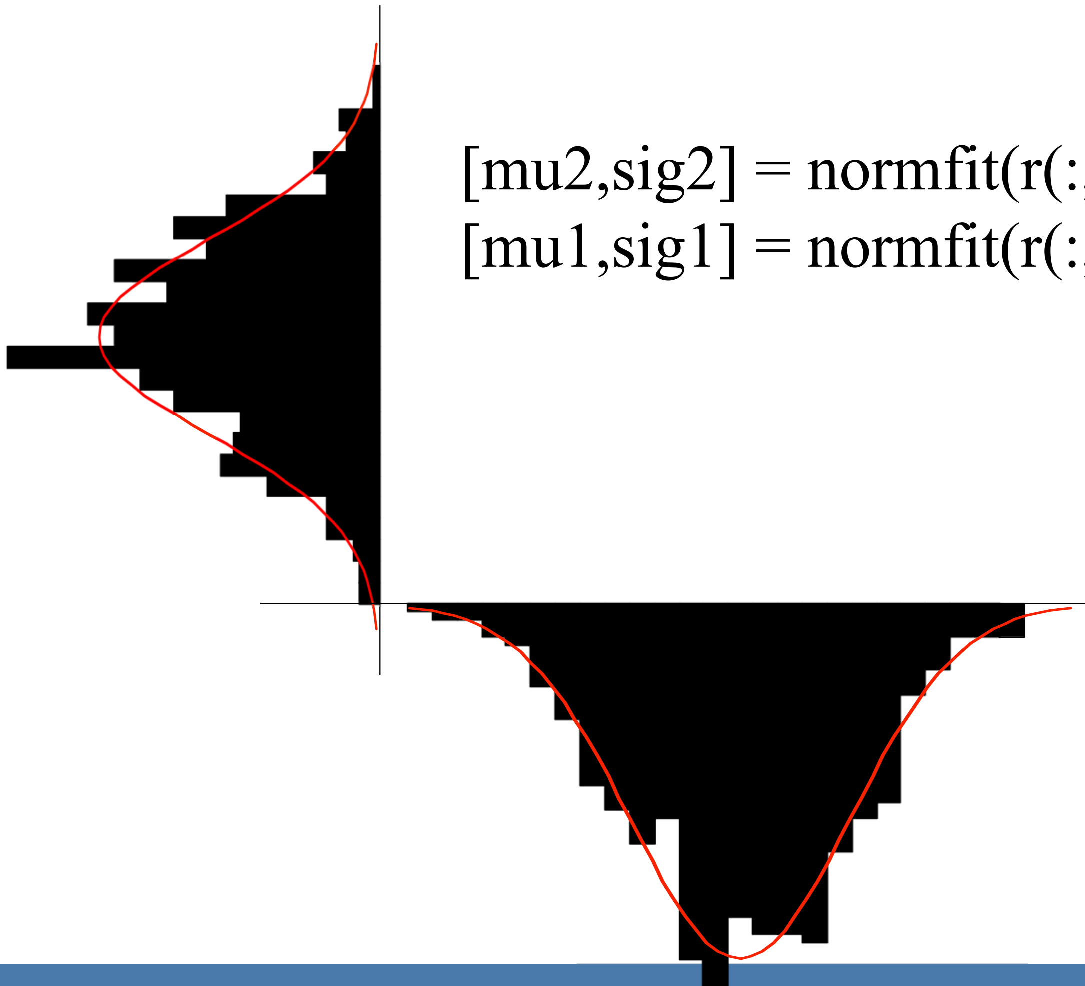
Independence



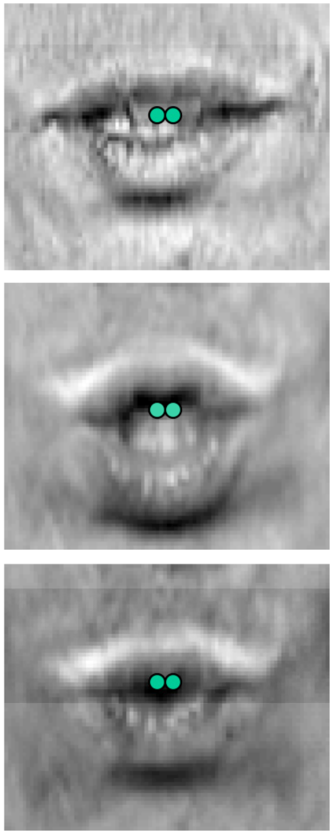
$p(\mathrm{x1})p(\mathrm{x2})$?



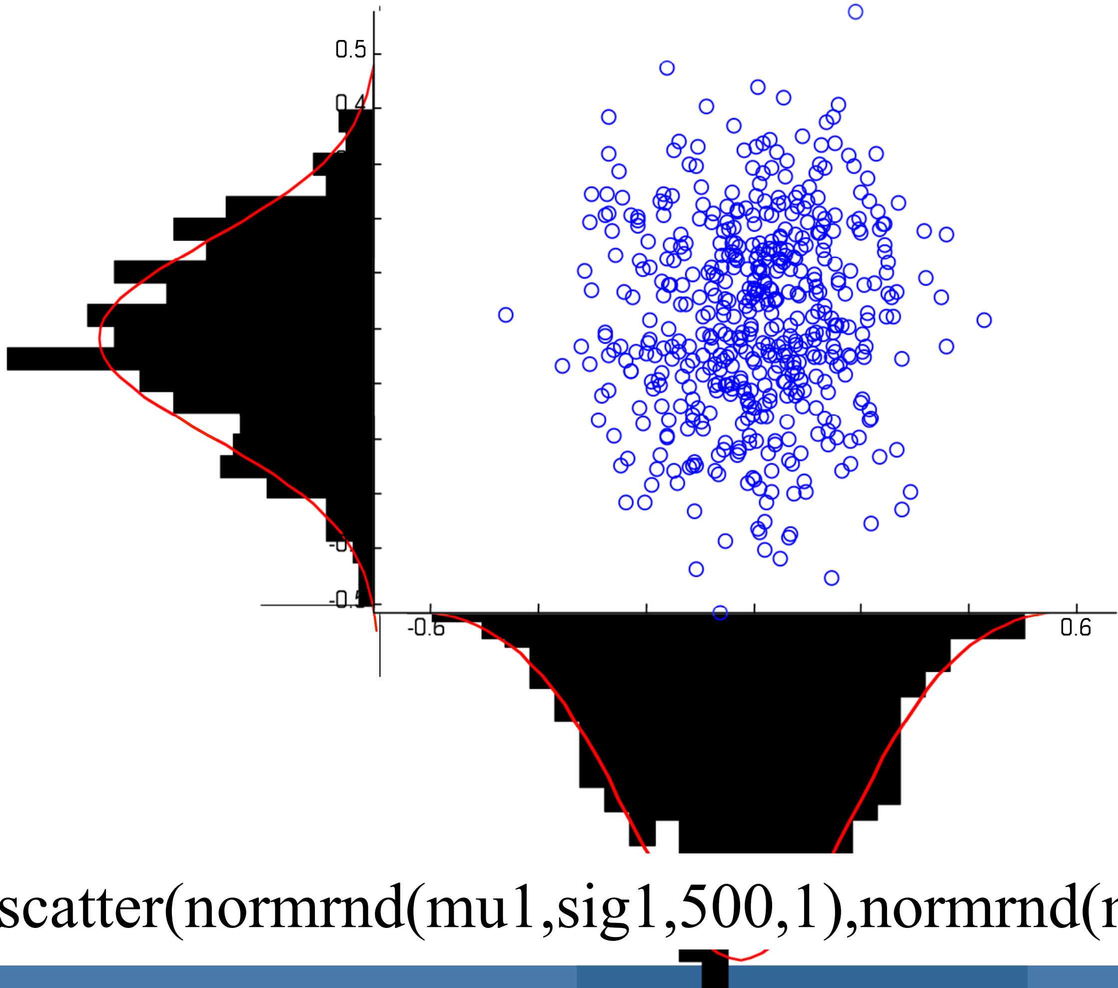
Independence

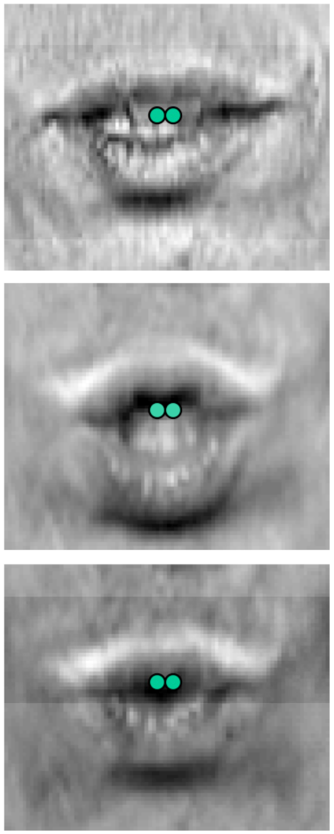


[mu2,sig2] = normfit(r(:,2))
[mu1,sig1] = normfit(r(:,1))



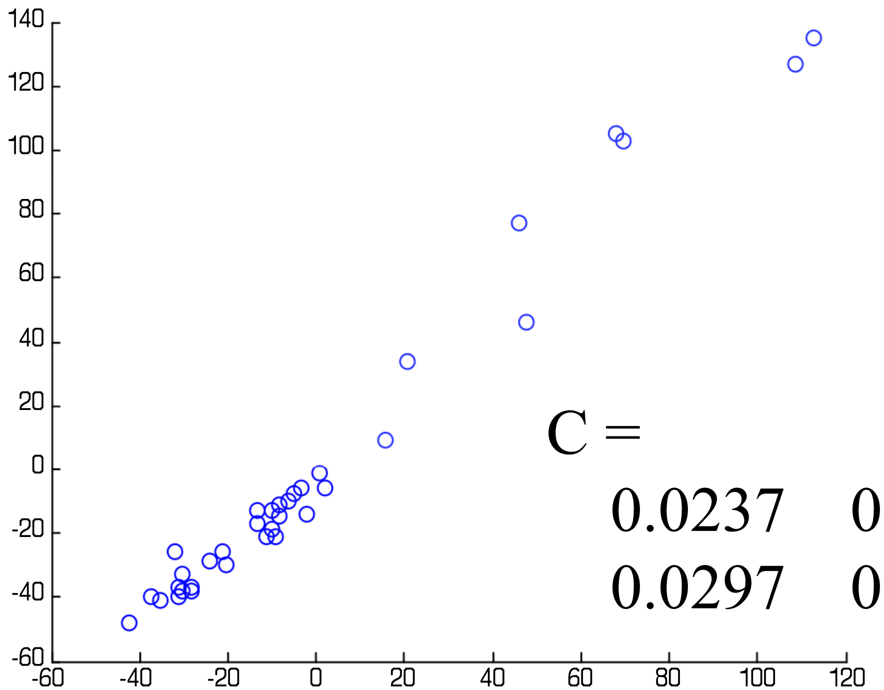
Independence





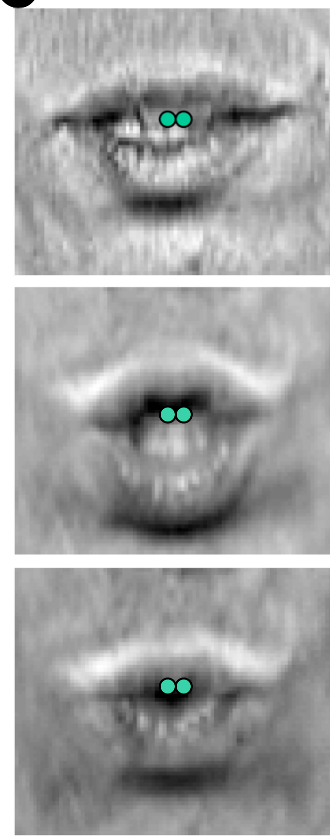
scatter(normrnd(mu1,sig1,500,1),normrnd(mu2,sig2,500,1))

Example: Covariance



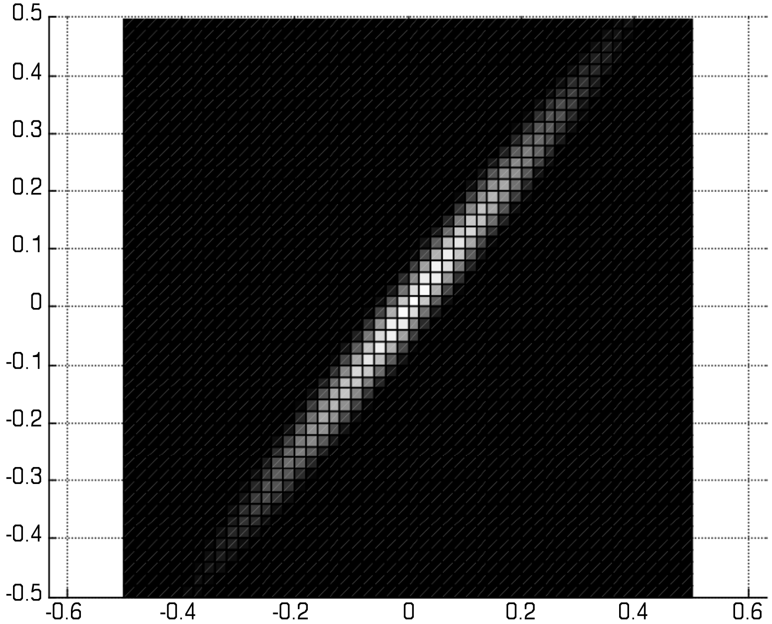
C =

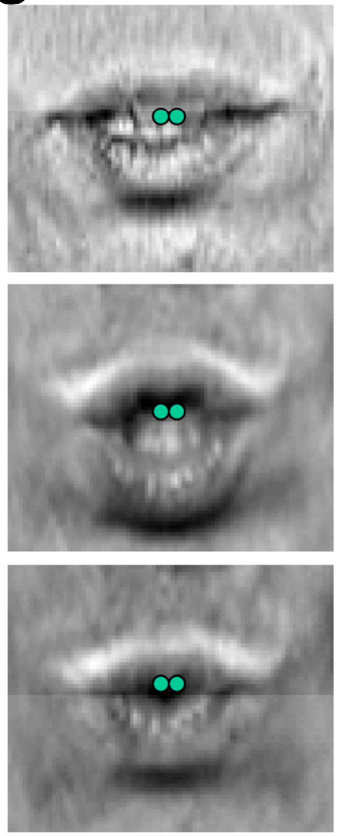
$$\begin{matrix} 0.0237 & 0.0297 \\ 0.0297 & 0.0384 \end{matrix}$$



$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, | C |^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

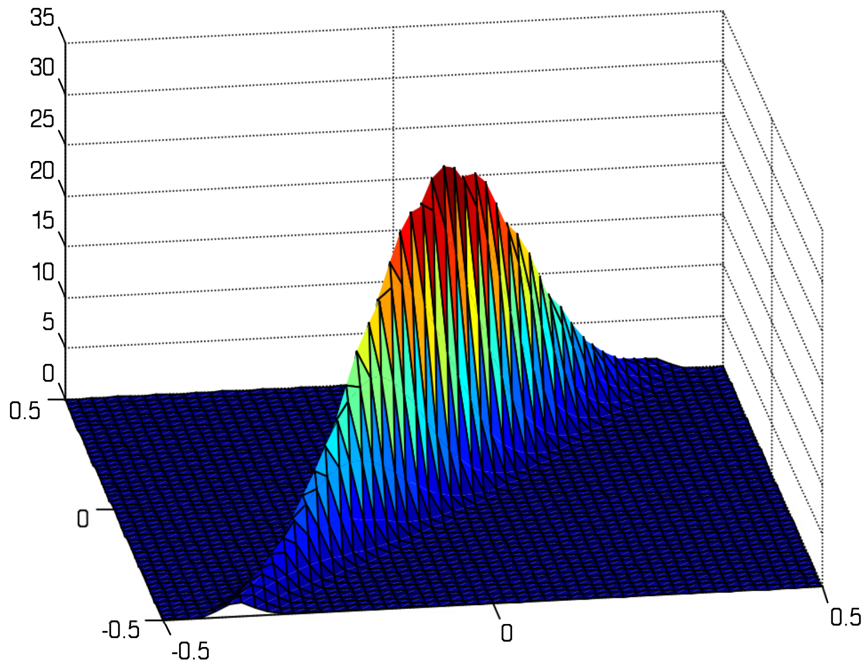
Example: Covariance

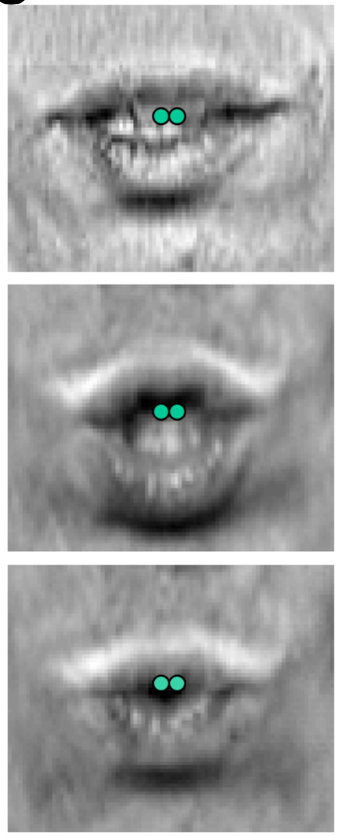




```
[X1, X2]=meshgrid(-0.5:0.02:0.5, -0.5:0.02:0.5);
X = [X1(:) X2(:)];
p=mvnpdf(X, mu, C);
surf(X1, X2, reshape(p,size(X1,1), size(X1,2)));
```

Example: Covariance





```
[X1, X2]=meshgrid(-0.5:0.02:0.5, -0.5:0.02:0.5);
X = [X1(:) X2(:)];
p=mvnpdf(X, mu, C);
surf(X1, X2, reshape(p,size(X1,1), size(X1,2))); colormap default;
```

Whitening

[U,D]=eig(C)

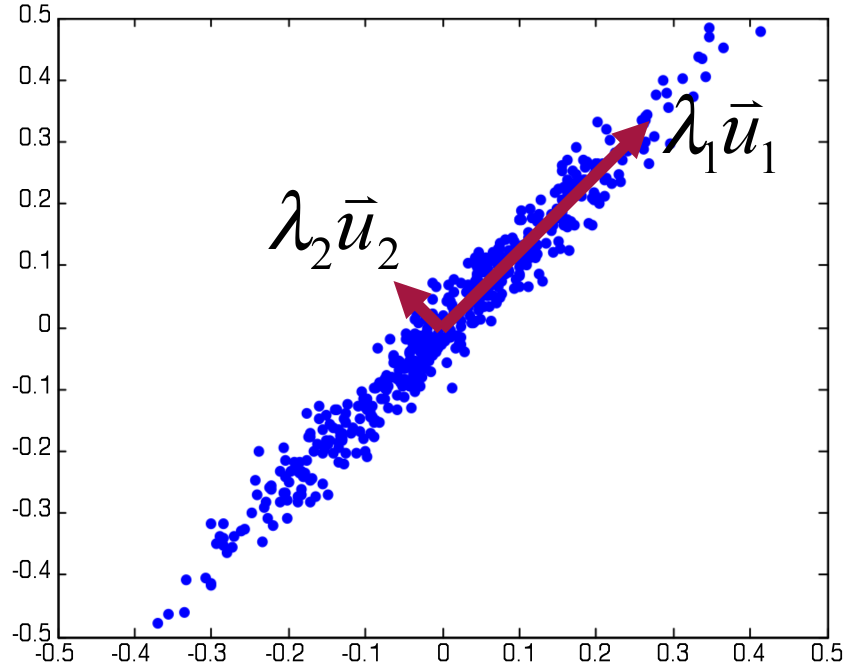
U =

-0.7876	0.6162
0.6162	0.7876

D =

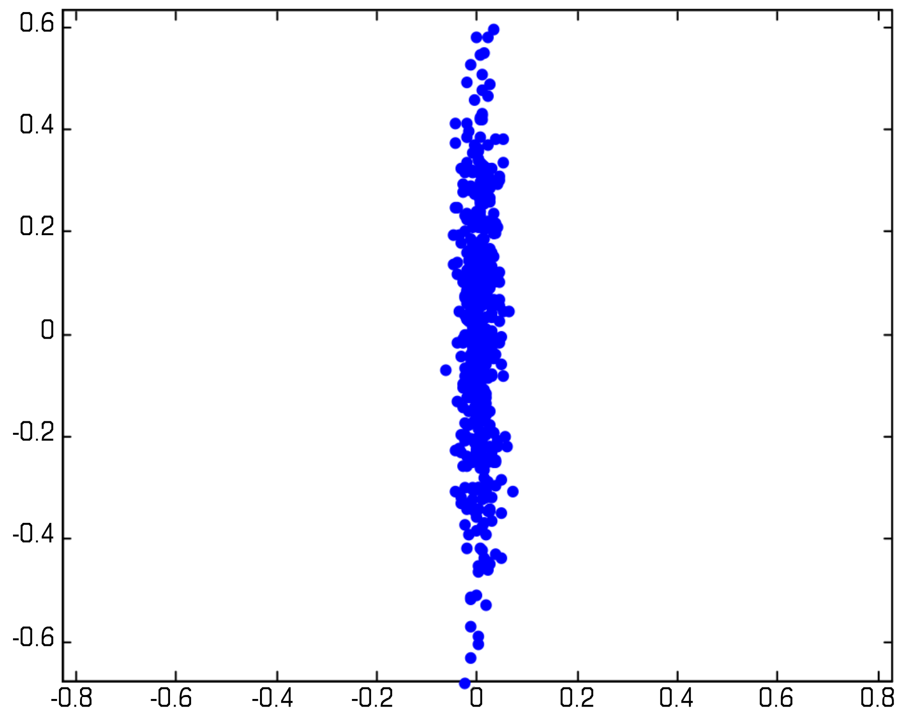
0.0004	0
0	0.0617

```
plot(r(:,1), r(:,2), '.');
```



Whitening

```
[U, D]=eig(C)
% project points onto basis
coeffs = r*U;
plot(coeffs(:,1), coeffs(:,2), '.');
axis true
```

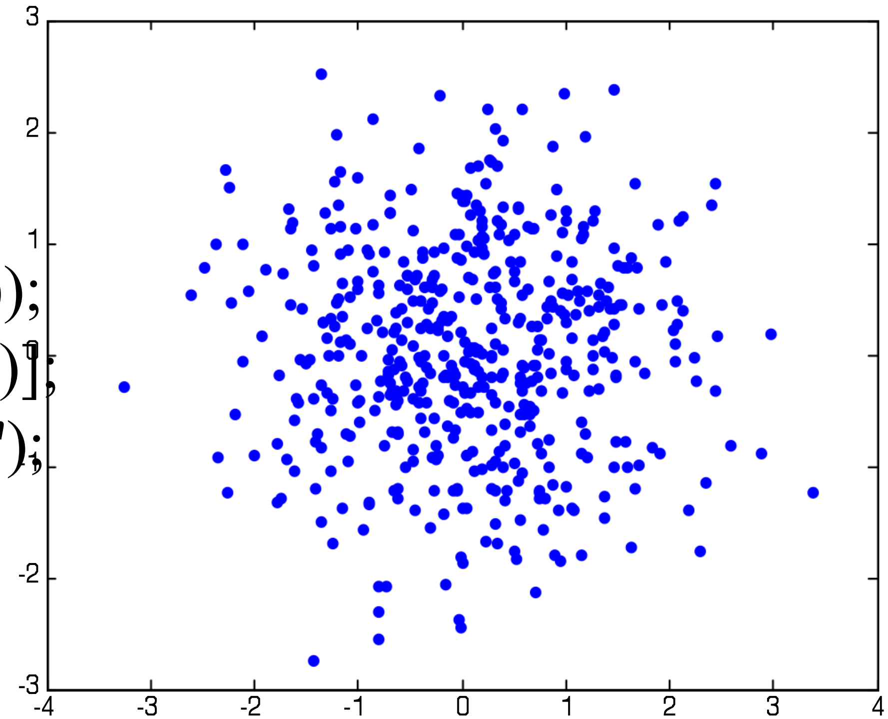


$$\tilde{x}^n = \bar{x}^n U$$

$\bar{x}^n$ zero mean $\qquad \mathrm{cov}(\tilde{x}) = D$

Whitening

```
[U, D]=eig(C)
% project points onto basis
coeffs = r*U;
coeffs2=[coeffs(1,:)/sqrt(D(1,1));
         coeffs(2,:)/sqrt(D(2,2))];
plot(coeffs2(:,1), coeffs2(:,2) ,'.');
axis true
```



$$\tilde{x}^n = \overline{x}^n U D^{-1/2}$$

$\overline{x}^n$ zero mean $\quad \text{cov}(\tilde{x}) = I$

Diagonal Covariance

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, | C |^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

Determinant is just the product of the diagonals (ie variances).

$$p(\vec{x}) = \prod_{i=1:D} p(x_i) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} (\vec{x}_i - \vec{\mu}_i)^2 / \sigma_i{}^2 \right)$$

Some Facts

$$C = \mathrm{E}[(\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T] = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}$$

If x and y are statistically independent then s_{xy}=0.

If s_{xy}=0, then x and y are uncorrelated.

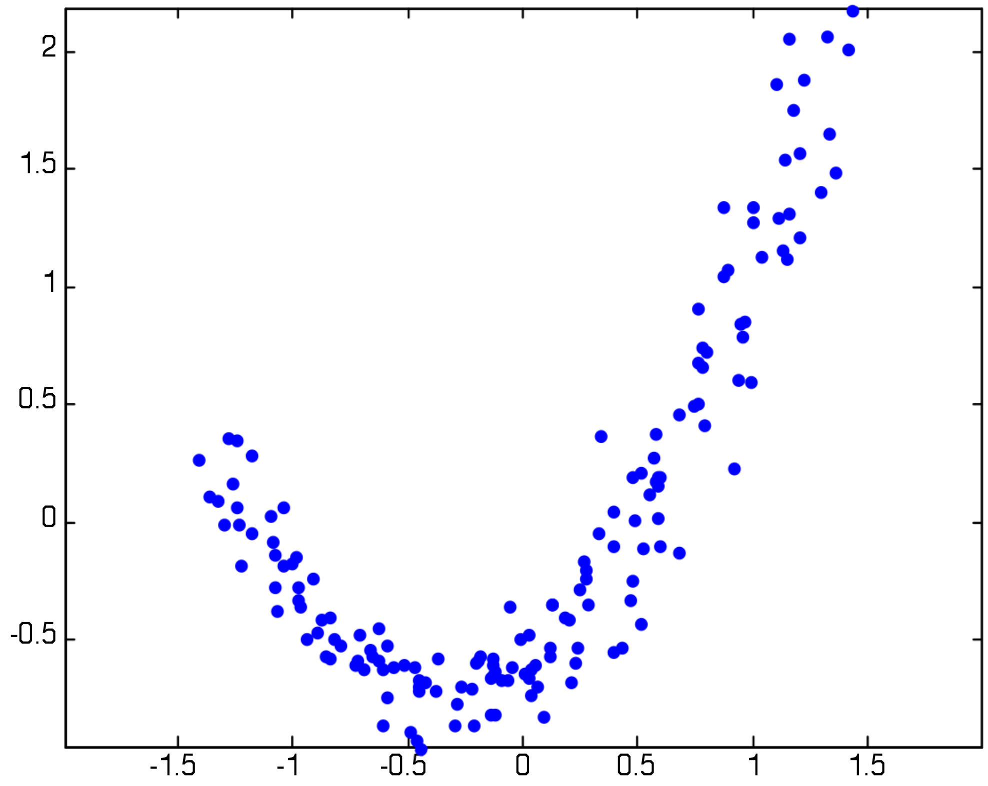
Uncorrelated does not imply statistically independent.
Uncorrelated and Gaussian does.

PCA de-correlates the directions but unless the data is Gaussian, the coefficients are not statistically independent.

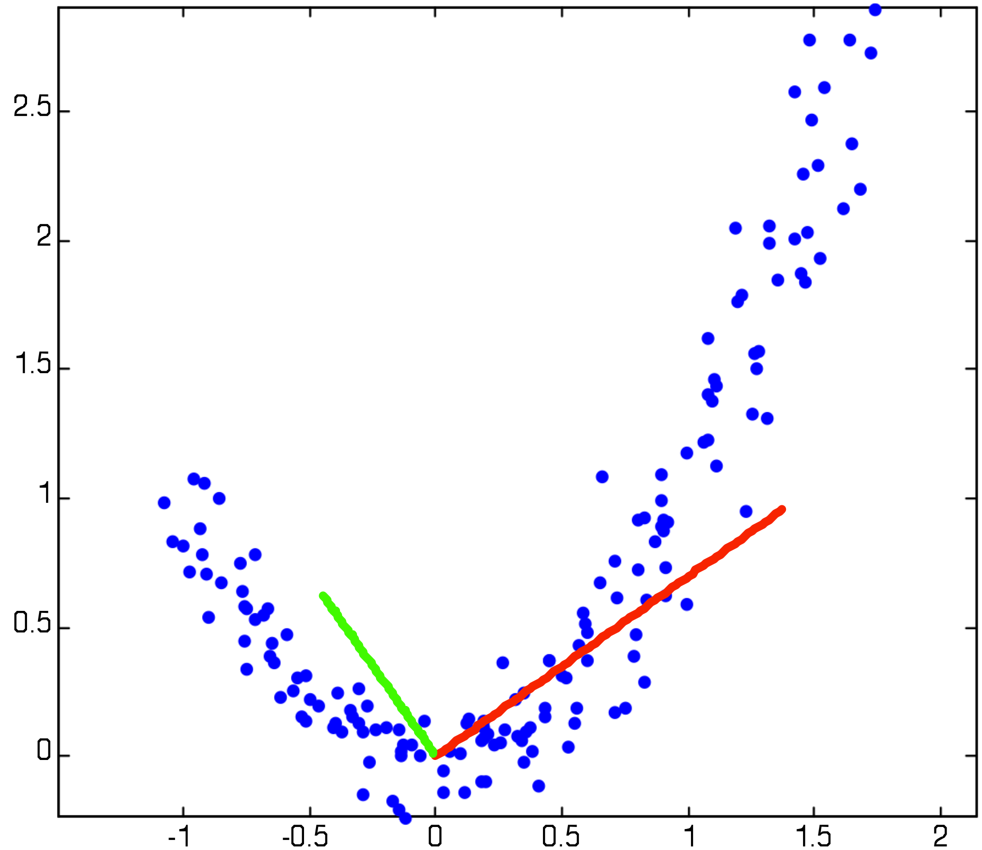
Why does decorrelated not imply statistically independent?

- PCA takes into account the second-order statistics in the data (in the covariance matrix).
- The covariance matrix captures correlation.
- PCA decorrelates the data.
- But covariance is only a second order statistic.
- Gaussians are fully described by their first and second order statistics (mean and covariance) –decorrelating then results in statistical independence
- But if the data has non-zero higher order statistics, decorrelating will not make the dimensions statistically independent.

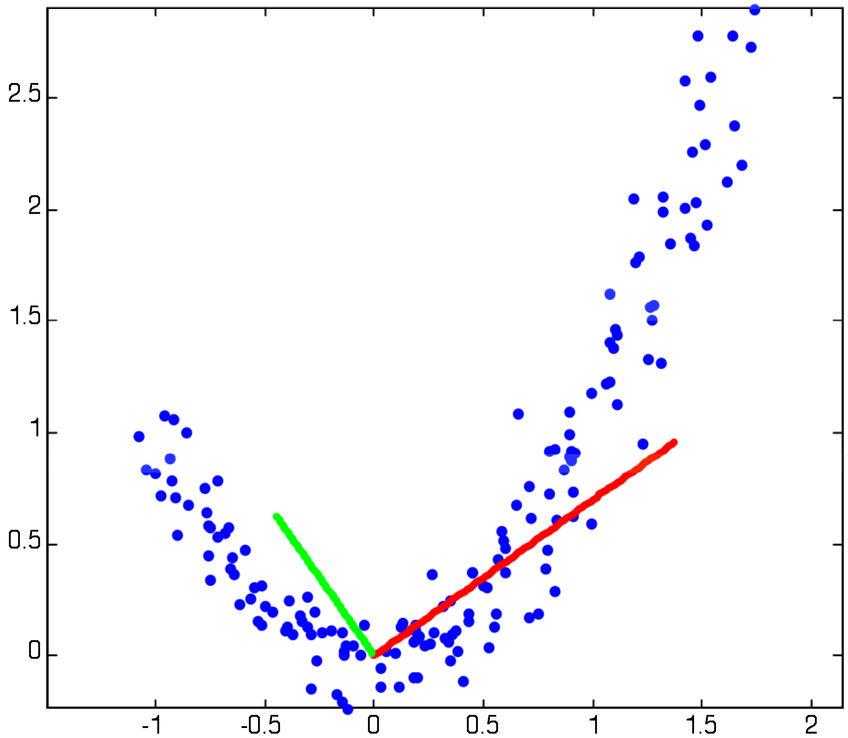
PCA and non-Gaussian data

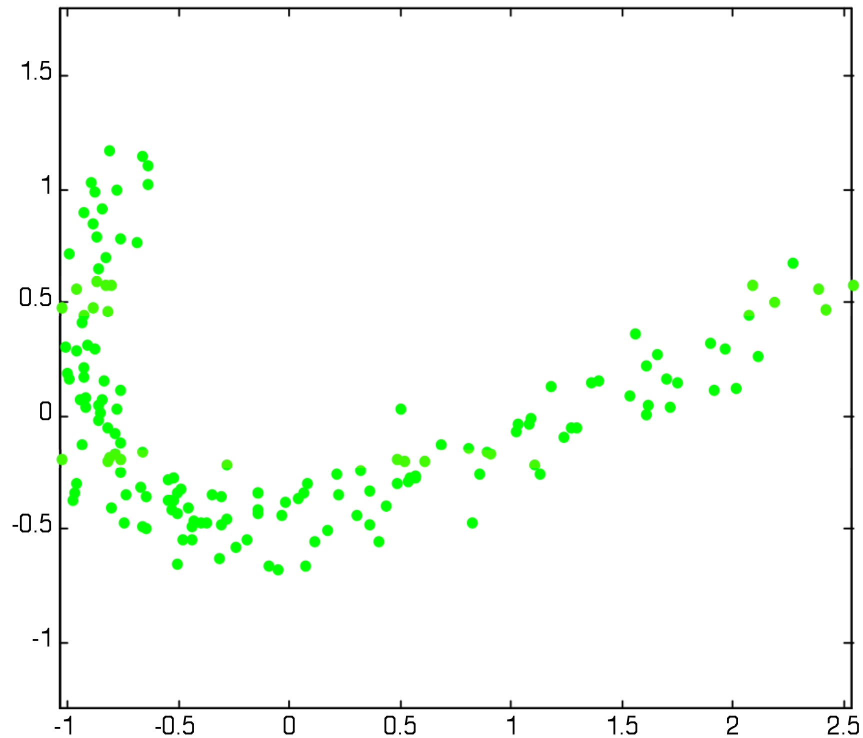


PCA and non-Gaussian data



PCA and non-Gaussian data





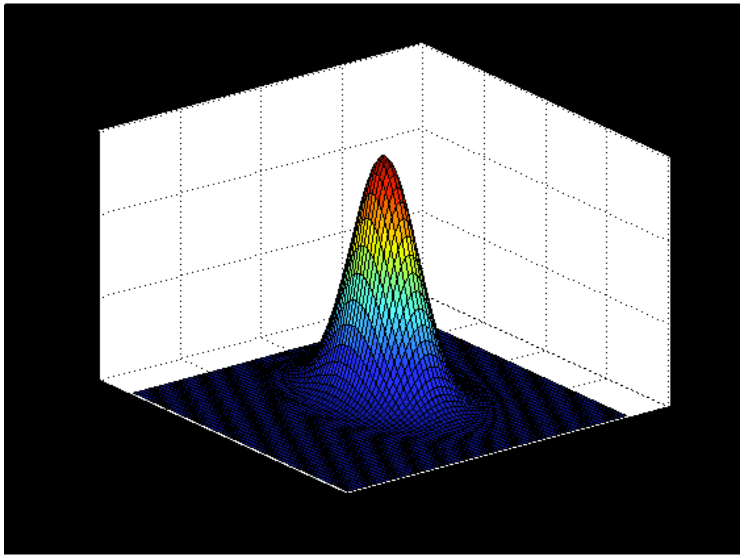
Decorrelated: corr(X'*u1,X'*u2)
ans = -5.8981e-017

cov(X'*u1,X'*u2)

0.9834	-0.0000
-0.0000	0.2010

Not statistically independent.

PCA and Covariance



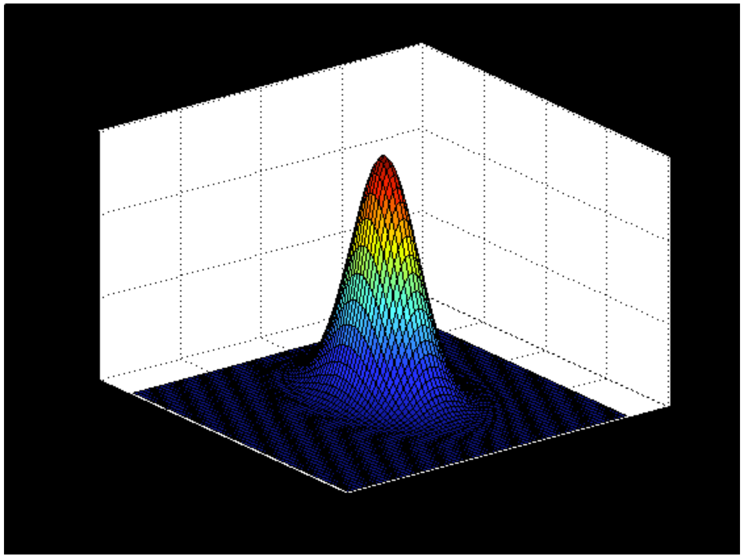
surf(m36)

An=A-meanmouth
U=eigenvectors
U36=matrix of linear coeffs

Let's look at how a_3 and a_6 co-vary.

```
U36=[An*U(:,3) An*U(:,6)];
C36=cov(U36)
mu36=[mean(An*U(:,3)) mean(An*U(:,6))]
 m36=mvnpdf(X, mu36, C36);
```

Covariance



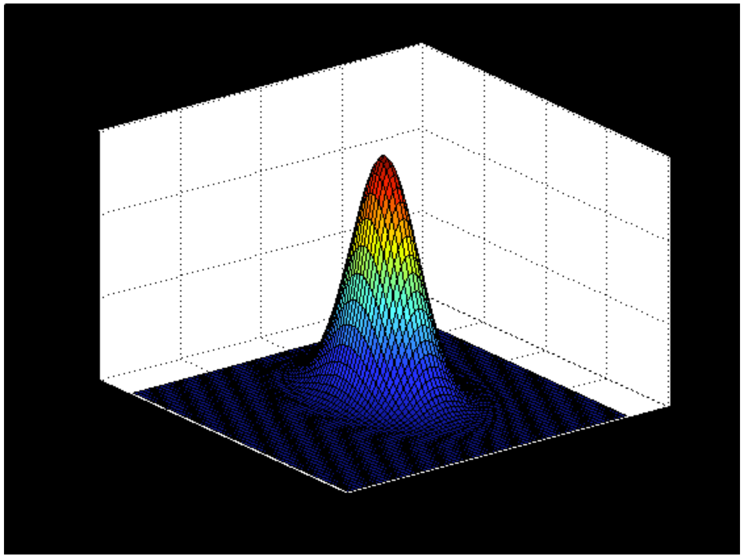
Multivariate Gaussian (Normal)

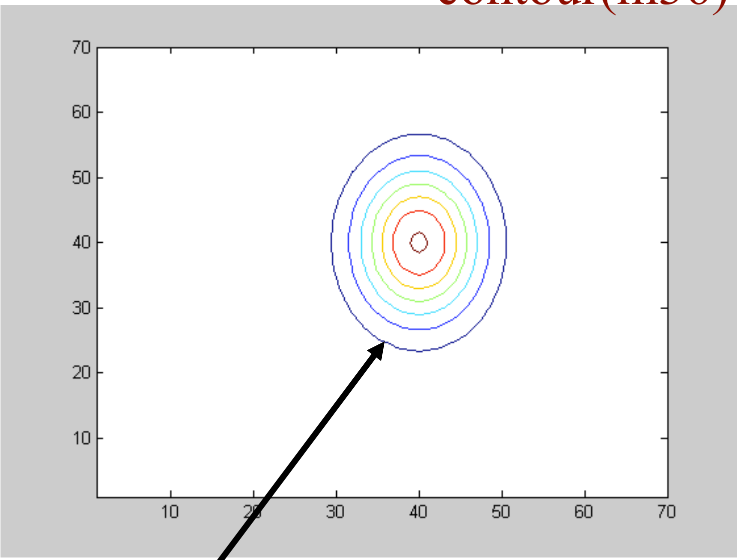
Mahalanobis distance Δ^2

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, |C|^{1/2}} \exp\left(-\frac{1}{2} \underbrace{(\vec{x}-\vec{\mu})^T C^{-1} (\vec{x}-\vec{\mu})}_{\Delta^2} \right)$$

Covariance Ellipse

contour(m36)





hyperellipsoids of constant Mahalanobis distance Δ^2

Note the ellipse is axis-aligned. Why?

Mahalanobis distance

$$p(\vec{x}) = \frac{1}{(2\pi)^{D/2} \, |\, C \,|^{1/2}} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu}) \right)$$

$$\widetilde{x} = \vec{x} - \vec{\mu}$$

$$d(\widetilde{x}) = \left(\widetilde{x}^T C^{-1} \widetilde{x} \right)$$

$$C = USU^T$$

Mahalanobis Distance

$$d(\tilde{x}) = \left(\tilde{x}^T C^{-1} \tilde{x}\right)$$

$$= \tilde{x}^T (USU^T)^{-1} \tilde{x}$$

$$= \tilde{x}^T U S^{-1} U^T \tilde{x}$$

$$= y^T S^{-1} y$$

$$= \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$\approx \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}$$

Linear coefficients

$$y = U^T \tilde{x}$$

Error in approximation?

- Above measures “distance in feature space”.
- Residual error is “distance from feature space”. This can be approximated.
- See Moghaddam & Pentland paper on website.
- Taking approximate error into account improves detection for problem 3.