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

- Subtract mean
- Project

\[
\begin{bmatrix}
-725.1966 \\
500.9761 \\
872.4541
\end{bmatrix}
\]
Mouth Space

\[
\begin{bmatrix}
-725.1966 \\
500.9761 \\
872.4541
\end{bmatrix}

\begin{bmatrix}
715.2646 \\
-376.2406 \\
-331.9533
\end{bmatrix}
\]
Classification
Classification

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

\[ p(a_3 \mid \neg\text{mouth}) \quad \text{and} \quad p(a_3 \mid \text{mouth}) \]
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

\[ p(\text{mouth} \mid a_3) = \frac{p(a_3 \mid \text{mouth}) p(\text{mouth})}{p(a_3)} \]

likelihood \quad prior

normalization constant
(independent of mouth)
Maximum A Posteriori Classification

From \( a_3 \) alone, it looks like MAP classification will always prefer the not-mouth interpretation.
What about the other coefficients?

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

posterior:

$p(a_6 | \text{mouth})$
Conditional Independence

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

\[ C = \begin{bmatrix} 0.0237 & 0.0297 \\ 0.0297 & 0.0384 \end{bmatrix} \]

\[ C = \text{cov} \left( \frac{A(:,30*88+46)}{255}, \frac{A(:,30*88+47)}{255} \right) \]
Example: Covariance

\[
P(\bar{x}) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp \left( -\frac{1}{2} (\bar{x} - \bar{\mu})^T C^{-1} (\bar{x} - \bar{\mu}) \right)
\]

\[
C = \begin{pmatrix}
0.0237 & 0.0297 \\
0.0297 & 0.0384
\end{pmatrix}
\]
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 $p(x_1, x_2)$
Example: Covariance
Marginals

$p(x_1)$

$p(x_2)$

$\text{histfit}(r(:,2),25)$
Independence

\[ p(x_1)p(x_2) ? \]
Independence

\[
[\mu_2, \sigma_2] = \text{normfit}(r(:,2))
\]

\[
[\mu_1, \sigma_1] = \text{normfit}(r(:,1))
\]
Independence

```
scatter(normrnd(mu1,sig1,500,1),normrnd(mu2,sig2,500,1))
```
Example: Covariance

\[
p(\bar{x}) = \frac{1}{(2\pi)^{D/2} \sqrt{|C|}} \exp\left(-\frac{1}{2} (\bar{x} - \bar{\mu})^T C^{-1} (\bar{x} - \bar{\mu})\right)
\]

\[
C = \\
\begin{pmatrix}
0.0237 & 0.0297 \\
0.0297 & 0.0384
\end{pmatrix}
\]
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

\[
[X_1, X_2] = \text{meshgrid}(-0.5:0.02:0.5, -0.5:0.02:0.5);
X = [X_1(:) X_2(:)];
p = \text{mvnpdf}(X, \mu, C);
surf(X_1, X_2, \text{reshape}(p, \text{size}(X_1,1), \text{size}(X_1,2))); \text{colormap default};
\]
Whitening

\[[U, D] = \text{eig}(C)\]

\[
U = 
\begin{bmatrix}
-0.7876 & 0.6162 \\
0.6162 & 0.7876 \\
\end{bmatrix}
\]

\[
D = 
\begin{bmatrix}
0.0004 & 0 \\
0 & 0.0617 \\
\end{bmatrix}
\]

\[\text{plot}(r(:,1), r(:,2), '.');\]
Whitening

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

\[ \tilde{x}^n = \bar{x}^n U \]
\[ \bar{x}^n \text{ zero mean} \quad \text{cov}(\tilde{x}) = D \]
Whitening

\[ [U, D] = \text{eig}(C) \]
\% project points onto basis
\[
\text{coeffs} = r \times U;
\]
\[
\text{coeffs2} = \left[ \frac{\text{coeffs}(1,:)}{\sqrt{D(1,1)}}; \frac{\text{coeffs}(2,:)}{\sqrt{D(2,2)}} \right];
\]
\[
\text{plot(coeffs2(:,1), coeffs2(:,2), '.'); axis true}
\]

\[
\tilde{x}^n = \bar{x}^n U D^{-1/2}
\]
\[
\bar{x}^n \text{ zero mean} \quad \text{cov} (\tilde{x}) = I
\]
Diagonal Covariance

\[
p(\bar{x}) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp\left( -\frac{1}{2} (\bar{x} - \bar{\mu})^T C^{-1} (\bar{x} - \bar{\mu}) \right)
\]

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

\[
p(\bar{x}) = \prod_{i=1:D} p(x_i) = \prod_{i} \frac{1}{\sqrt{2\pi \sigma_i}} \exp\left( -\frac{1}{2} (\bar{x}_i - \bar{\mu}_i)^2 / \sigma_i^2 \right)
\]
Some Facts

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

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: \( \text{corr}(X'u_1, X'u_2) \)
\[
\begin{align*}
\text{ans} &= -5.8981 \times 10^{-17} \\
\text{cov}(X'u_1, X'u_2) &= \begin{bmatrix} 0.9834 & -0.0000 \\ -0.0000 & 0.2010 \end{bmatrix}
\end{align*}
\]

Not statistically independent.
PCA and Covariance

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

$$U_{36} = [\text{An} \times U(:,3), \text{An} \times U(:,6)]$$

$$C_{36} = \text{cov}(U_{36})$$

$$\mu_{36} = [\text{mean}(\text{An} \times U(:,3)), \text{mean}(\text{An} \times U(:,6))]$$

$$m_{36} = \text{mvnpdf}(X, \mu_{36}, C_{36})$$

$\text{An} = A - \text{meanmouth}$

$U = \text{eigenvectors}$

$U_{36} = \text{matrix of linear coeffs}$
Covariance

Multivariate Gaussian (Normal)

\[ p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{x} - \mathbf{\mu})^T C^{-1} (\mathbf{x} - \mathbf{\mu}) \right) \]

Mahalanobis distance \( \Delta^2 \)
Covariance Ellipse

hyperellipsoids of constant Mahalanobis distance $\Delta^2$

Note the ellipse is axis-aligned. Why?
Mahalanobis distance

\[ p(\tilde{x}) = \frac{1}{(2\pi)^{D/2} |C|^{1/2}} \exp\left( -\frac{1}{2} (\tilde{x} - \tilde{\mu})^T C^{-1} (\tilde{x} - \tilde{\mu}) \right) \]

\[ \tilde{x} = \tilde{x} - \tilde{\mu} \]

\[ d(\tilde{x}) = (\tilde{x}^T C^{-1} \tilde{x}) \]

\[ C = USU^T \]
Mahalanobis Distance

\[
d(\tilde{x}) = (\tilde{x}^T C^{-1} \tilde{x})
= \tilde{x}^T (USU^T)^{-1} \tilde{x}
= \tilde{x}^T US^{-1} U^T \tilde{x}
= y^T S^{-1} y
\]

Linear coefficients

\[
y = U^T \tilde{x}
\]

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

\[
\approx \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}
\]
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.