Pattern Recognition

Lecture 4

More on Invariants

PCA

Thomas M. Breuel

News

please remember to hand in your exercise sheets

you can now run Sage on tux1 etc.

if worksheets are not available from home page, there may be PDF files at the bottom of the page

Last Time

Looked extensively at ellipses, matrices, distances, second order moments, eigenvectors.

We will be using this again and again (a little in this lecture).

Today:

- more on invariant representations (Fourier Transform, Moments)
- first steps on PCA

Moments

Consider a distribution

$$I = I(x, y)$$

where $\int I(x, y)\, dx\, dy = 1$

$$v = (x, y) = (v_{0,} v_1)$$

First order moments:

$$\mu = E_I[v]$$
$$\mu_x = E_I[x]$$
$$\mu_y = E_I[y]$$
$$\mu_i = E_I[v_i]$$

Second Moments

Second order moments:

$$\sigma_{ij} = E_I[v_i v_j]$$

Second order centralized moments:

$$\sigma_{ij} = E_I[(v_i - \mu_i)(v_j - \mu_j)]$$

Second order centralized moments are invariant under translation.

Second Moments

recall: centroid

$$\bar{x} = \sum I \cdot x$$
$$\bar{y} = \sum I \cdot y$$

centralized second order moments

$$\sigma_{xx} = \sum I (x - \bar{x})^2$$
$$\sigma_{yy} = \sum I (y - \bar{y})^2$$
$$\sigma_{xy} = \sum I (x - \bar{x})(y - \bar{y})$$

matrix of centralized second moments

$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix}$$

Translation Invariant Recognition

Idea 1 – Canonicalization

- compute first order moments
- canonicalize image
- nearest neighbor on canonicalized image

Idea 2 – Invariant Moment Recognition

- compute centralized second and higher order moments
- compute vector of moments $v=(\sigma_{11,} \sigma_{12,} \ldots)$
- nearest neighbors on moment vector

Moments are used differently in the two cases.

Translation Invariant Recognition — FT

Centralized moments are invariant... anything else?

The Fourier Spectrum!

$$F(u,v)=\int f(x,y)e^{-2\pi i(xu+yv)}dx\,dy$$

$$S(u,v)=|F(u,v)|^2$$

Translation Invariant Recognition — FT

Under translation...

Let $f'(x,y) = f(x+r, y+s)$

Then $F'(u,v) = c \cdot F(u,v)$ where $|c| = 1$

$$S'(u,v) = |F'(u,v)|^2 = |c \cdot F(u,v)|^2 = |c|^2 |F(u,v)|^2 = |F(u,v)|^2 = S(u,v)$$

Translation-Invariant Recognition

- translation invariant distance measure (last time)
- canonicalization by first moment
- translation invariant moments
- Fourier Spectrum

Completeness of Representation?

We compute...

- translation invariant moments $\phi=(\sigma_{11,}...)$
- Fourier spectrum $\phi=(F_{00,}F_{01,}...)$

We have

- $x=y\Rightarrow\phi(x)=\phi(y)$

What about

- $\phi(x)=\phi(y)\Rightarrow x=y$?

Completeness of Representation

translation invariant moments

- second order moments not sufficient
- high order centralized moments?

Fourier spectrum

- Fourier *transform* is invertible
- Fourier *spectrum* ?

Count the number of parameters

Completeness of Representation

formula for σ_{12} (assume $\sum I(x, y) = 1$) using sums

$$\sigma_{12} = \sum_{xy} x\, y\, I(x, y) = \sum_{xy} m(x, y)\, I(x, y) = \sum_{i} m_i I_i$$

we see that this is just a dot product, considering m and I to be vectors:

$$\sigma_{12} = m \cdot I$$

Completeness of Representation

roughly...

- $m_{ab}(x, y) = x^a y^b$
- observe that the m_{ab} are linearly independent
- you can also orthonormalize them
- enough of them for a complete basis for I
- also kind-of works for centralized moments

Completeness of Representation

Fourier Transform also is a linear transform

Basis functions are $m_{uv}(x, y) = e^{-2\pi i (ux+vy)}$

Note that these are actually overcomplete (b/c complex)

Spectrum has same number of values as original image

Turns out image can be recovered

$$x = y \Leftrightarrow \phi(x) = \phi(y)$$

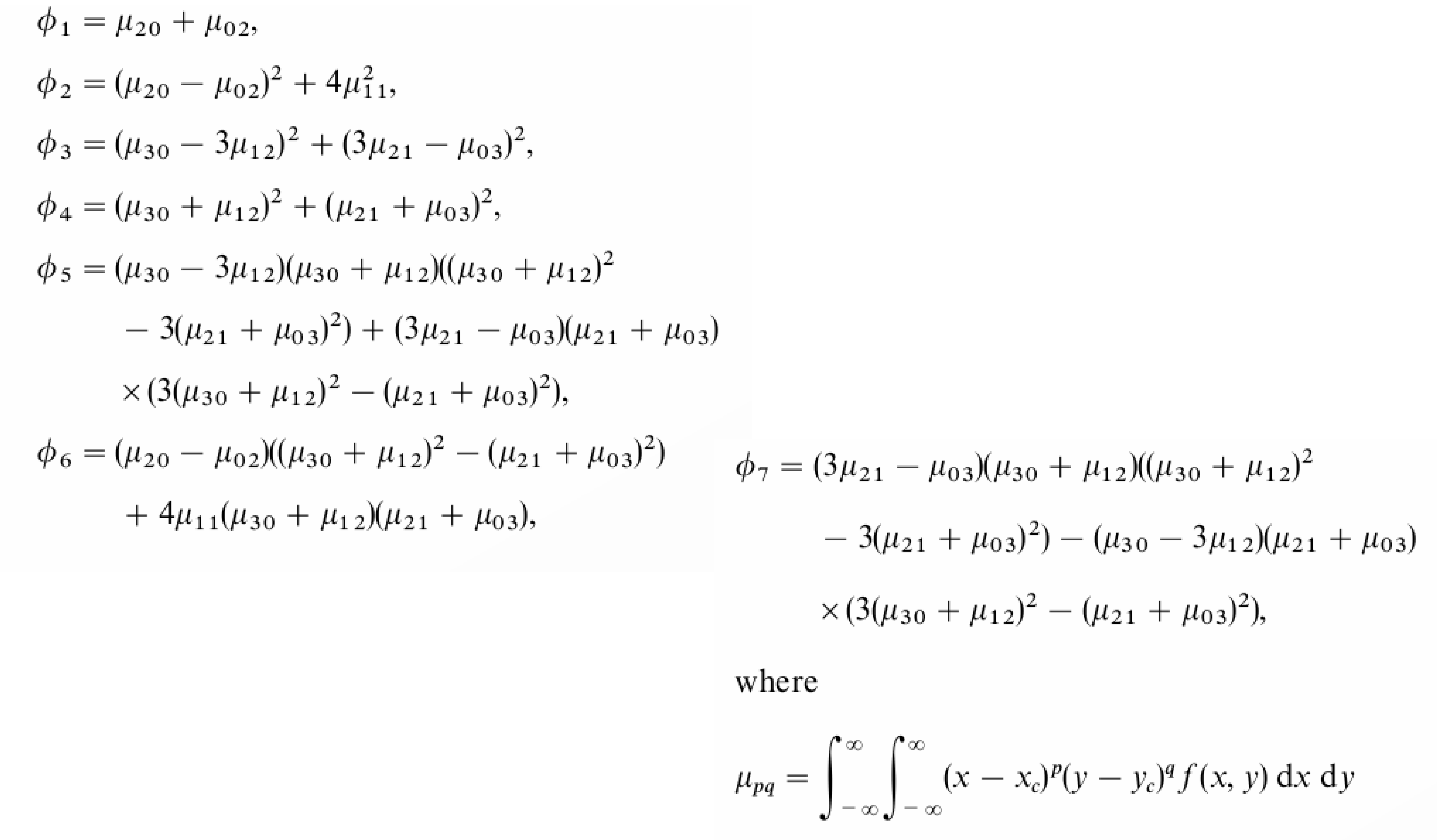
Rotation-Invariant Recognition?

how would you do it?

Rotation-Invariant Recognition

- rotation invariant distance measure (like last time)
- canonicalization by second moments (figure it out)
- rotation invariant moments
- Fourier-Mellin transformation

Hu Moments = Rotation Invariant Combinations of Centr. Moments



$$\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02},\\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2,\\
\phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2,\\
\phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2,\\
\phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2\\
&\qquad - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})\\
&\qquad \times (3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2),\\
\phi_6 &= (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)\\
&\qquad + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}),
\end{aligned}$$

$$\begin{aligned}
\phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2\\
&\qquad - 3(\mu_{21} + \mu_{03})^2) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})\\
&\qquad \times (3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2),
\end{aligned}$$

where

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y$$

Centralized Second Moments

think of

- moments as covariance matrix of a normal density
- centroid is the “mean” of the normal density
- normal density is the “best second order approximation” to the pattern
- actual normal density will give rise to this centroid and moments
- level set of covariance matrix is ellipse
- major and minor axis of this ellipse correspond to eigenvectors

canonicalizing transformation

- find a rotation matrix that diagonalizes Σ

Rotation vs Skew Correction

canonicalization by rotation

$$R\Sigma = \begin{pmatrix} \sigma'_{xx} & 0 \\ 0 & \sigma'_{yy} \end{pmatrix} \quad \text{where} \quad R = R(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

then $f(v) = Rv$

canonicalization by skew correction (shear transformation)

$$T = T(\alpha) = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix}$$

choose T such that one of the two eigenvectors of Σ becomes vertical

Summary of Correction / Invariants

two kinds

- correction: “undo” the transformation and compute canonical rep.
- invariants: compute a function that doesn't change under transform.

covered

- translation invariance: correction or moments
- rotation invariance: correction or moments
- skew invariance: correction (below)

more kinds of invariants

- scale invariance: correction or moments
- affine invariance: see literature
- blur invariance: see literature

PCA (Principal Component Analysis)

input data

- let $\{x_1, \ldots, x_N\}$ be samples from some pattern recognition problem
- let Σ be the covariance matrix (empirical or true)
- think of the x_i as coming from a normal density (even if they don't)

recall that

- a normal density with cov Σ can be derived by rotation and scaling from a spherical normal density
- $x \to S \cdot R \cdot x$ then $\Sigma^{-1} = R^T \cdot S^T \cdot S \cdot R$ (S diag scale, R rotation)
- assume that S is ordered in decreasing size

PCA

PCA = ...

- undo the rotation matrix R
- truncate the diagonal matrix (perform a projection) somewhere

PCA

input

- list of vectors $x_1, \ldots, x_N$

algorithm

- compute covariance matrix Σ
- compute the eigenvalues and eigenvectors
- sort eigenvalues largest to smallest
- the eigenvectors determine rotation matrix R
- the eigenvalues determine S^2
- take the input vectors x and transform them into $R^{-1} x$
- compute the projection $y = P_k R^{-1} x$
- perform classification with the $y_1, \ldots, y_N$

PCA — Why?

- the y have lower dimensionality than x
- the y are the best lsq approximation to x of dimension k
- the components of y are uncorrelated
- the variance of component j of y is S_{jj}
- the covariance matrix of y is S (it is diagonal)
- y contains the “non-noise portions” of x
- for any x, x' we have $d_2(y, y') \approx d_x(x, x')$

PCA — How to Compute?

obvious algorithm

- compute μ for the input data
- transform the data by $x_i \rightarrow x_i - \mu$
- compute data matrix X with rows equal to x_i
- compute $\hat{\Sigma} = X^T X / N$
- compute eigenvectors & eigenvalues, or compute svd

note

- 30 x 30 input image = 900 x 900 = 810000 element cov matrix

PCA — How to Compute?

iterative algorithm

- start with any unit vector v
- compute $v' = \Sigma v$
- normalize $v'' = v' / |v'|$
- iterate until convergence

this will usually give you the eigenvector corresponding to the largest eigenvalue

why? ... think about it in diagonalized form

PCA — How to Compute?

we have an algorithm for the “largest eigenvector”

how do we find the others?

iterative algorithm

- find the “largest eigenvector”, call it v_1
- project onto $v_1^{\perp}$
- repeat

projection... one of

- replace $x \rightarrow x - (x \cdot v_1) v_1$
- multiply $\hat{\Sigma}$ by a projection matrix

Summary

looked at invariants and canonicalization again

invariant feature extraction vs linear transforms

PCA — meaning and interpretation

PCA — “obvious” algorithm

PCA — iterative algorithm