Video signals

FEATURES EXTRACTION

Linear Independence

Different direction vectors are independent.

Parallel vectors are dependent. (+ve/-ve directions are parallel)



Basis & Orthonormal Bases

Basis (or axes): frame of reference



Basis & Orthonormal Bases

$$\begin{bmatrix} 2\\3 \end{bmatrix} = 2 \cdot \begin{bmatrix} 1\\0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0\\1 \end{bmatrix} = \begin{bmatrix} 1&0\\0&1 \end{bmatrix} \begin{bmatrix} 2\\3 \end{bmatrix};$$

$$\begin{bmatrix} -9\\6 \end{bmatrix} = -9 \cdot \begin{bmatrix} 1\\0 \end{bmatrix} + 6 \cdot \begin{bmatrix} 0\\1 \end{bmatrix} = \begin{bmatrix} 1&0\\0&1 \end{bmatrix} \begin{bmatrix} -9\\6 \end{bmatrix};$$

$$\begin{bmatrix} 2\\3 \end{bmatrix} = \frac{9}{7} \cdot \begin{bmatrix} 2\\1 \end{bmatrix} + \frac{4}{7} \cdot \begin{bmatrix} -1\\3 \end{bmatrix} = \begin{bmatrix} 2&-1\\1&3 \end{bmatrix} \begin{bmatrix} 9/7\\4/7 \end{bmatrix};$$

$$\begin{bmatrix} -9\\6 \end{bmatrix} = -3 \cdot \begin{bmatrix} 2\\1 \end{bmatrix} + 3 \cdot \begin{bmatrix} -1\\3 \end{bmatrix} = \begin{bmatrix} 2&-1\\1&3 \end{bmatrix} \begin{bmatrix} -3\\3 \end{bmatrix};$$

Basis & Orthonormal Bases

<u>Basis</u>: a space is totally defined by a set of vectors – any vector is a *linear combination* of the basis

Ortho-Normal: orthogonal + normal

Orthogonal: dot product is zero Normal: magnitude is one

$$\vec{x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \qquad \vec{x} \cdot \vec{y} = 0 \\ \vec{y} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \qquad \vec{x} \cdot \vec{z} = 0 \\ \vec{z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \qquad \vec{y} \cdot \vec{z} = 0$$

Projection: Using Inner Products



Marco Marcon

Eigenvalues & Eigenvectors

Eigenvectors (for a square *m*×*m* matrix **S**)





Video Signals

Eigenvalues & Eigenvectors (Properties)

The eigenvalues of real symmetric matrices are real.

$$A\vec{v}=\lambda\vec{v}$$

- > Then we can conjugate to get $\bar{A}\bar{\vec{v}}=\bar{\lambda}\bar{\vec{v}}$
- > If the entries of A are real, this becomes $A\vec{\vec{v}}=\bar{\lambda}\vec{\vec{v}}$
- > This proves that complex eigenvalues of real valued matrices come in conjugate pairs
- > Now transpose to get $\overline{\vec{v}}^t A^T = \overline{\vec{v}}^t \overline{\lambda}$. Because A is symmetric matrix we now have $\overline{\vec{v}}^t A = \overline{\vec{v}}^t \overline{\lambda}$.
- > Multiply both sides of this equation on the right with \vec{v} , i.e. $\vec{v}^t A \vec{v} = \vec{v}^t \bar{\lambda} \vec{v}$
- > On the other hand multiply $A\vec{v} = \lambda\vec{v}$ on the left by $\bar{\vec{v}}^t$ to get $\bar{\vec{v}}^t A\vec{v} = \bar{\vec{v}}^t \lambda \vec{v}$

$$\Rightarrow \bar{\vec{v}}^t \bar{\lambda} \vec{v} = \bar{\vec{v}}^t \lambda \vec{v} \Rightarrow \bar{\lambda} = \lambda \Rightarrow \lambda \text{ is real}$$

Eigenvalues & Eigenvectors (Properties)

If A is an n x n symmetric matrix, then any two eigenvectors that come from distinct eigenvalues are orthogonal.

$$A\vec{v}_i = \lambda_i \vec{v}_i$$

► Left multiply with \vec{v}_j^T to $A\vec{v}_i = \lambda_i \vec{v}_i \Rightarrow \vec{v}_j^T A \vec{v}_i = \vec{v}_j^T \lambda_i \vec{v}_i$

Similarly $\vec{v}_i^T A \vec{v}_j = \vec{v}_i^T \lambda_j \vec{v}_j$

From the above two equations $(\lambda_j - \lambda_i) \vec{v}_i^T \vec{v}_j = 0$

 $\Longrightarrow \vec{v}_i^T \vec{v}_j = 0$

 $\therefore \vec{v}_i$ and \vec{v}_i are perpendicular

Diagonalization

Stack up all the eigen vectors to get

 $AV = V\Lambda$

Where $V = [\vec{v}_1 \quad \vec{v}_2 \quad \dots \quad \vec{v}_n]; \Lambda = diag(\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n)$

If all eigenvectors are linearly independent, V is invertible.

 $A = V\Lambda V^{-1}$

Suppose A is symmetric matrix, then $V^T = V^{-1}$

Therefore $A = V\Lambda V^T$

Variance Formula



Standard Deviation

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

[standard deviation = square root of the variance]











Video Signals

Variance(x) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

= $\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}) (x_i - \bar{x})$

Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$





Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance(x, y) =
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$



Covariance Matrix

$$Cov(\Sigma) = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_m) \\ cov(x_2, x_1) & cov(x_2, x_2) & \cdots & cov(x_2, x_m) \\ \vdots & \vdots & \vdots & \vdots \\ cov(x_m, x_1) & cov(x_m, x_2) & \cdots & cov(x_m, x_m) \end{bmatrix}$$

$$Cov\left(\Sigma\right) = \frac{1}{n}(X - \bar{X})(X - \bar{X})^{T}; where \ X = \begin{bmatrix} x_{1} \\ x_{2} \\ \vdots \\ x_{m} \end{bmatrix}$$

Covariance Matrix

$$Cov(\Sigma) = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_m) \\ cov(x_2, x_1) & cov(x_2, x_2) & \cdots & cov(x_2, x_m) \\ \vdots & \vdots & \vdots & \vdots \\ cov(x_m, x_1) & cov(x_m, x_2) & \cdots & cov(x_m, x_m) \end{bmatrix}$$

- > Diagonal elements are variances, i.e. Cov(x, x)=var(x).
- Covariance Matrix is symmetric.
- It is a positive semi-definite matrix.



Covariance Matrix

- Covariance is a real symmetric positive semi-definite matrix.
 - ✤ All eigenvalues must be real
 - Eigenvectors corresponding to different eigenvalues are orthogonal
 - All eigenvalues are greater than or equal to zero
 - Covariance matrix can be diagonalized,

i.e. $Cov = PDP^T$



Correlation



- Covariance determines whether relation is positive or negative, but it was impossible to measure the degree to which the variables are related.
- Correlation is another way to determine how two variables are related.
- In addition to whether variables are positively or negatively related, correlation also tells the degree to which the variables are related each other.



Correlation

$$\rho_{xy} = Correlation(x, y) = \frac{cov(x, y)}{\sqrt{var(x)}\sqrt{var(y)}}.$$

$-1 \leq Correlation(x, y) \leq +1$



Eivectors and Eigenvalues

We can interpret this correlation as an ellipse whose major axis is one eigenvalue and the minor axis length is the other:

No correlation yields a circle, and perfect correlation yields a line.



The Principal Components

All principal components (PCs) start at the origin of the ordinate axes.

First PC is direction of maximum variance from origin

Subsequent PCs are orthogonal to 1st PC and describe maximum residual variance



Algebraic Interpretation

Given m points in a n dimensional space, for large n, how does one project on to a low dimensional space while preserving broad trends in the data and allowing it to be visualized?

Algebraic Interpretation – 1D

Given m points in a n dimensional space, for large n, how does one project on to a 1 dimensional space?



Choose a line that fits the data so the points are spread out well along the line

Video Signals

Algebraic Interpretation – 1D

Formally, minimize sum of squares of distances to the line.



Why sum of squares? Because it allows fast minimization.



PCA: 2D representation



PCA Scores



PCA Eigenvalues



Harris Corner Detector

Many applications benefit from features localized in (x,y)



Edges well localized only in one direction -> detect corners

Desirable properties of corner detector

- Accurate localization
- Invariance against shift, rotation, scale, brightness change
- Robust against noise, high repeatability



What patterns can be localized most accurately?

Local displacement sensitivity

$$S(\Delta x, \Delta y) = \sum_{(x,y)\in window} \left[f(x,y) - f(x + \Delta x, y + \Delta y) \right]^2$$

Linear approximation for small $\Delta x, \Delta y$

$$f(x + \Delta x, y + \Delta y) \approx f(x, y) + f_x(x, y)\Delta x + f_y(x, y)\Delta y$$

$$S(\Delta x, \Delta y) \approx \sum_{(x,y)\in window} \left[\begin{pmatrix} f_x(x,y) & f_y(x,y) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right]^2$$

Iso-sensitivity curves are ellipses

$$S(\Delta x, \Delta y) \approx (\Delta x \quad \Delta y) \left(\sum_{(x,y) \in window} \begin{bmatrix} f_x^2(x,y) & f_x(x,y) f_y(x,y) \\ f_x(x,y) f_y(x,y) & f_y^2(x,y) \end{bmatrix} \right) \left(\begin{array}{c} \Delta x \\ \Delta y \end{array} \right) = \left(\Delta x \quad \Delta y \right) \mathbf{M} \left(\begin{array}{c} \Delta x \\ \Delta y \end{array} \right)$$

Harris criterium

Often based on eigenvalues λ_1 , λ_2 of "structure matrix" (or "normal matrix" or "second-moment matrix")

$$\mathbf{M} = \begin{bmatrix} \sum_{\substack{(x,y) \in window}} f_x^2(x,y) & \sum_{\substack{(x,y) \in window}} f_x(x,y) f_y(x,y) \\ \sum_{\substack{(x,y) \in window}} f_x(x,y) f_y(x,y) & \sum_{\substack{(x,y) \in window}} f_y^2(x,y) \end{bmatrix}$$

 $f_x(x,y)$ – horizontal image gradient $f_y(x,y)$ – vertical image gradient

Measure of "cornerness"

 $C(x, y) = \det(\mathbf{M}) - k \cdot (trace(\mathbf{M}))^{2}$ $= \lambda_{1}\lambda_{2} - k \cdot (\lambda_{1} + \lambda_{2})$





Harris corner values



$$= \lambda_1 \lambda_2 - k \cdot \left(\lambda_1 + \lambda_2\right)^2$$

Video Signals

Keypoint Detection: Input



Video Signals

Harris cornerness



Thresholded cornerness



Video Signals

Local maxima of cornerness



Superimposed keypoints



Video Signals

Robustness of Harris Corner Detector Invariant to brightness offset: $f(x,y) \rightarrow f(x,y) + c$

Invariant to shift and rotation



High-dimensional data in computer vision





Face images

Handwritten digits

Eigenfaces

Images are converted into vectors:

Then all training images are user to build the average face and the covariance matrix, whose eigenvectors are called eigenfaces.





Marco Marcon

Eigenfaces

Each new face can then be assumed as a weighted sum of the eigenfaces.



The weights of each eigenface represent a possible signature of a face for face-recognition tasks.

PCA for image compression





Video Signals