CSE 152: Computer Vision

Hao Su

Lecture 4: Linear Algebra Applications (Least Square, Harris Corner)

Agenda

- Least Square Problem
- Harris Corner Detector

Agenda

- Least Square Problem
- Harris Corner Detector

Simple Linear Regression

Response Variable

Covariate

Linear Model: $$Y = mX + b$$

Slope

Intercept (bias)

Motivation

- One of the most widely used techniques
  - We will use it a few times in the class
- Easy to interpret
- Efficient to solve

Multiple Linear Regression

Rewriting with Matrix Notation

- Represent data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ as:

Covariate (Design) Matrix

Response Vector

$$X = \underbrace{\left[\begin{pmatrix} — & x_1 & — \\ — & x_2 & — \\ & \cdots & \\ — & x_n & — \end{pmatrix}\right.}_{\mathbf{p}}\left.\vphantom{\begin{pmatrix} x \\ x \\ \cdots \\ x \end{pmatrix}}\right\}\mathbf{n} \in \mathbb{R}^{np} \qquad Y = \underbrace{\left[\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}\right.}_{\mathbf{1}}\left.\vphantom{\begin{pmatrix} x \\ x \\ \cdots \\ x \end{pmatrix}}\right\}\mathbf{n} \in \mathbb{R}^n$$

Assume *X* has rank p
(not degenerate)

Rewriting with Matrix Notation

- Rewriting the model using matrix operations:

$$Y = X\theta + \epsilon$$

- Dropping the sign and flipping from maximization to minimization:

$$\mathcal{L}(\theta) = \arg\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} \underbrace{(y_i - \theta^T x_i)^2}$$

Minimize Sum (Error)^2

- Squared Loss
  - Least Squares Regression

Pictorial Interpretation of Squared Error

Minimizing the Squared Error

$$\hat{\theta}_{\mathrm{MLE}} = \arg\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2$$

- Take the gradient and set it equal to zero

Minimizing the Squared Error

$$\hat{\theta}_{\mathrm{MLE}} = \arg\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2$$

- Taking the gradient

$$-\nabla_\theta \log \mathcal{L}(\theta) = \nabla_\theta \sum_{i=1}^{n} (y_i - \theta^T x_i)^2$$

Chain Rule → $$= -2 \sum_{i=1}^{n} (y_i - \theta^T x_i) x_i$$

$$= -2 \sum_{i=1}^{n} y_i x_i + 2 \sum_{i=1}^{n} (\theta^T x_i) x_i$$

- Rewriting the gradient in matrix form:

$$-\nabla_\theta \log \mathcal{L}(\theta) = -2\sum_{i=1}^{n} y_i x_i + 2\sum_{i=1}^{n}(\theta^T x_i)x_i$$
$$= -2X^T Y + 2X^T X\theta$$

- To make sure the log-likelihood is convex compute the second derivative (Hessian)

$$-\nabla^2 \log \mathcal{L}(\theta) = 2X^T X$$

- If X is full rank then $X^T X$ is positive definite and therefore θ_{MLE} is the minimum
  - Address the degenerate cases with regularization

$$-\nabla_\theta \log \mathcal{L}(\theta) = -2X^T y + 2X^T X\theta = 0$$

- Setting gradient equal to 0 and solve for θ_{MLE}:

$$(X^T X)\hat{\theta}_{\text{MLE}} = X^T Y$$

$$\hat{\theta}_{\text{MLE}} = (X^T X)^{-1} X^T Y$$

Normal Equations (Write on board)

Connection to Pseudo-Inverse

$$\hat{\theta}_{\mathrm{MLE}} = \underbrace{(X^T X)^{-1} X^T}_{X^\dagger} Y$$

Moore-Penrose Psuedoinverse

- Generalization of the inverse:
  - Consider the case when X is square and invertible:

$$X^\dagger = (X^T X)^{-1} X^T = X^{-1} (X^T)^{-1} X^T = X^{-1}$$

  - Which implies $\theta_{\mathrm{MLE}} = X^{-1} Y$ the solution to $X\,\theta = Y$ when X is square and invertible

Agenda

- Least Square Problem
- Harris Corner Detector

Example application

- Panorama stitching
  - We have two images – how do we combine them?

Goal: Repeatability

- We want to detect (at least some of) the same points in both images.

With these points, there's no chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.

Detection: Basic Idea

- We do not know which other image locations the feature will end up being matched against.
- But we can compute **how stable** a location is in appearance with respect to small variations in position *u*.

- *Compare image patch against local neighbors.*

Corner Detection: Basic Idea

- We might recognize the point by looking through a small window.
- We want a window shift in *any direction* to give *a large change* in intensity.

"Flat" region:
no change in all directions

"Edge":
no change along the edge direction

"Corner":
significant change in all directions

A. Efros

Corner Detection by Auto-correlation

Change in appearance of window *w*(*x*,*y*) for shift [*u,v*]:

$$E(u,v) = \sum_{(x,y)\in\mathcal{W}} [I(x+u, y+v) - I(x,y)]^2$$

***I*(*x*, *y*)**

***E*(*u*, *v*)**

$$E(u,v) = \sum_{(x,y)\in\mathcal{W}} [I(x+u, y+v) - I(x,y)]^2$$

Think-Pair-Share:

Correspond the three red crosses

$E(u,v)$

$E(u,v)$

As a surface

Corner Detection by Auto-correlation

Change in appearance of window *w*(*x*,*y*) for shift [*u,v*]:

$$E(u,v) = \sum_{(x,y)\in\mathcal{W}} [I(x+u, y+v) - I(x,y)]^2$$

We want to discover how E behaves for small shifts

But this is very slow to compute naively.
O(window_width^2 * shift_range^2 * image_width^2)

O(11^2 * 11^2 * 600^2) = 5.2 billion of these
14.6 thousand per pixel in your image

Corner Detection by Auto-correlation

Change in appearance of window *w*(*x*,*y*) for shift [*u,v*]:

$$E(u,v) = \sum_{(x,y)\in\mathcal{W}} [I(x+u, y+v) - I(x,y)]^2$$

We want to discover how E behaves for small shifts

But we know the response in *E* that we are looking for – strong peak.

Can we just approximate $E(u,v)$ locally by a quadratic surface?

$\approx$

Recall: Taylor series expansion

A function f can be represented by an infinite series of its derivatives at a single point *a*:

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots.$$

As we care about window centered, we set *a = 0* (MacLaurin series)

Approximation of $f(x) = e^x$ centered at f(0)

Local quadratic approximation of $E(u,v)$ in the neighborhood of (0,0) is given by the *second-order Taylor expansion*:

$$E(u,v) \approx E(0,0) + [u \quad v]\begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2}[u \quad v]\begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

Notation: partial derivative

Corner Detection: Mathematics

$$E(u,v) = \sum_{x,y} \left[I(x+u, y+v) - I(x,y) \right]^2$$

Second-order Taylor expansion of $E(u,v)$ about (0,0):

$$E(u,v) \approx E(0,0) + [u \;\; v] \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} [u \;\; v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E_u(u,v) = \sum_{x,y} 2 \left[I(x+u, y+v) - I(x,y) \right] I_x(x+u, y+v)$$

$$E_{uu}(u,v) = \sum_{x,y} 2 \, I_x(x+u, y+v) I_x(x+u, y+v)$$

$$+ \sum_{x,y} 2 \left[I(x+u, y+v) - I(x,y) \right] I_{xx}(x+u, y+v)$$

$$E_{uv}(u,v) = \sum_{x,y} 2 \, I_y(x+u, y+v) I_x(x+u, y+v)$$

$$+ \sum_{x,y} 2 \left[I(x+u, y+v) - I(x,y) \right] I_{xy}(x+u, y+v)$$

Corner Detection: Mathematics

$$E(u,v) = \sum_{x,y} \left[I(x+u, y+v) - I(x,y) \right]^2$$

Second-order Taylor expansion of $E(u,v)$ about (0,0):

$$E(u,v) \approx E(0,0) + [u \;\; v] \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} [u \;\; v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(0,0) = 0$$

$$E_u(0,0) = 0$$

$$E_v(0,0) = 0$$

$$E_{uu}(0,0) = \sum_{x,y} 2 \qquad I_x(x,y) I_x(x,y)$$

$$E_{vv}(0,0) = \sum_{x,y} 2 \qquad I_y(x,y) I_y(x,y)$$

$$E_{uv}(0,0) = \sum_{x,y} 2 \qquad I_x(x,y) I_y(x,y)$$

Corner Detection: Mathematics

The quadratic approximation simplifies to

$$E(u,v) \approx [u \;\; v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where *M* is a *second moment matrix* computed from image derivatives:

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Corners as distinctive interest points

$$M = \sum \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

2 x 2 matrix of image derivatives (averaged in neighborhood of a point)

Notation:

$$I_x \Leftrightarrow \frac{\partial I}{\partial x} \qquad I_y \Leftrightarrow \frac{\partial I}{\partial y} \qquad I_x I_y \Leftrightarrow \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}$$

Interpreting the second moment matrix

The surface $E(u,v)$ is locally approximated by a quadratic form. Let's try to understand its shape.

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Interpreting the second moment matrix

Consider a horizontal "slice" of $E(u, v)$: $$[u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}$$

This is the equation of an ellipse.

Diagonalization of M: $$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

The axis lengths of the ellipse are determined by the eigenvalues, and the orientation is determined by a rotation matrix R.

Classification of image points using eigenvalues of *M*

Classification of image points using eigenvalues of *M*

Cornerness

$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

Classification of image points using eigenvalues of *M*

Cornerness

$$C = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

Remember your linear algebra:

Determinant: $\det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \cdots \lambda_n$.

Trace: $\mathrm{tr}(A) = \sum_i \lambda_i$.

$$C = \det(M) - \alpha \, \mathrm{trace}(M)^2$$

Harris corner detector

1) Compute *M* matrix for each window to recover a *cornerness* score C.
   - Note: We can find *M* purely from the per-pixel image derivatives!

2) Threshold to find pixels which give large corner response (C > threshold).

3) Find the local maxima pixels, i.e., suppress non-maxima.

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

Harris Detector: Steps

Harris Detector: Steps

Compute corner response C

Harris Detector: Steps

Find points with large corner response: C > threshold

Harris Detector: Steps

Take only the points of local maxima of C

Harris Detector: Steps

Invariance and covariance

- Are locations *invariant* to photometric transformations
- and *covariant* to geometric transformations?
  - **Invariance:** image is transformed and corner locations do not change
  - **Covariance:** if we have two transformed versions of the same image, features should be detected in corresponding locations