Optimization - 1

CMSC828D
Fundamentals of Computer Vision

Outline - I

- Algebraic distance
  - Definition
  - Problems
  - Scaling and Normalization
- Different ways of computing the Cost function
  - Errors in both coordinates
  - Transfer Error and Reprojection Error
- “Physics/Geometry” based distances
  - General Examples
  - Examples in Vision
- Constraints
  - Equality constraints
    - Lagrange multipliers and Penalty function methods
  - Inequality Constraints

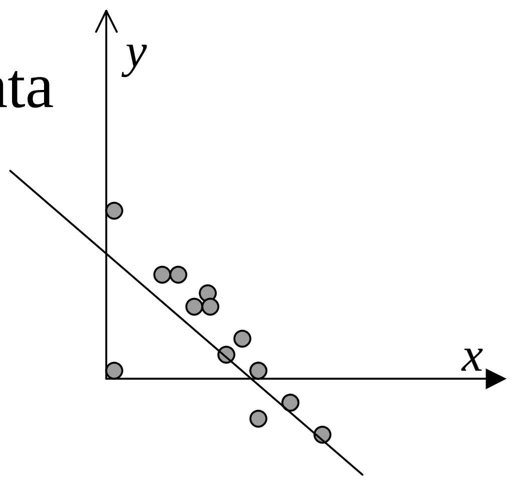
Outline - II

- Other Metrics
  - Riemann Lebesgue lemma
  - Sobolev norms
- Statistical Cost Functions
  - Mahalanobis distance
  - Maximum Likelihood (ML), Expectation Maximization (EM) and Maximum a Posteriori (MAP)
- Robust Estimation
  - Outliers and Inliers
  - Median Estimators
  - RANSAC

Typical Optimization Problems

- Model fitting
  - Fit a straight line or polynomial through data

$$y_i = \Sigma_j \, a_j \, x^j_i$$



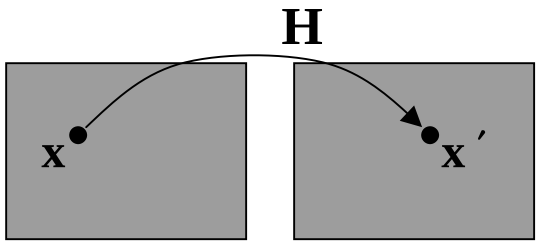
  - Fit a sum of cosines, exponentials etc.

$$y_i = \Sigma_j \, a_j \, \phi_j(x_i)$$

Model ϕ_j s, parameters a_js data (x_i,y_i)

- Determine a transformation
  - Determine a homography matrix

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$



  - Determine the fundamental matrix

$$\mathbf{x}'^{\mathbf{t}}\mathbf{F}\mathbf{x} = 0$$

Least Squares

- Look for a solution to a linear system of equations

$$\mathbf{Ax}=\mathbf{b}$$

- Number of equations and unknowns need not match
- Look for solution by minimizing ||**Ax - b**||
  - minimize the distance between the vectors **Ax** and **b**
- Differentiate $(A_{ij}x_j-b_i).(A_{ik}x_k-b_i)$ with respect to x_l
- Recall $\frac{\partial x_i}{\partial x_l}=\delta_{il}$

$$\frac{\partial}{\partial x_l}(A_{ij}x_j - b_i)\bullet(A_{ik}x_k - b_i) = 0$$

$$(A_{ij}\delta_{jl})\bullet(A_{ik}x_k - b_i)+(A_{ij}x_j - b_i)\bullet(A_{ik}\delta_{kl}) = 0$$

$$A_{il}\bullet(A_{ik}x_k - b_i)+(A_{ij}x_j - b_i)\bullet A_{il} = 2\left(A_{il}A_{ik}x_k - A_{il}b_i\right) = 0$$

$$A_{il}A_{ik}x_k = A_{il}b_i$$

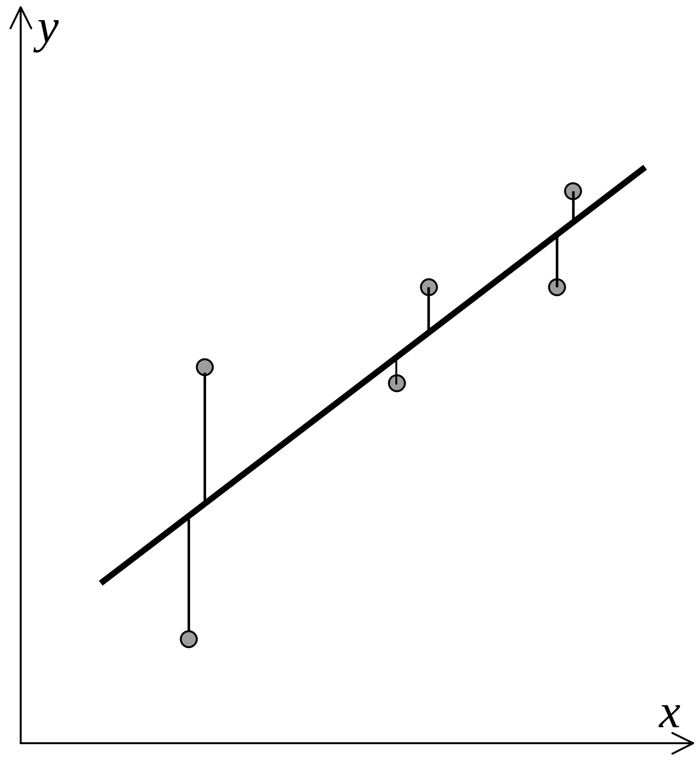
- Same as the solution of $\mathbf{A}^t\mathbf{Ax}=\mathbf{A}^t\mathbf{b}$

Optimization – Physical Cost Function

- Adjust parameters of a system or model to maximize or minimize something
  - $, Distance
- Ideally there is a real cost being minimized
  - E.g. Dollars or distance travelled
  - Then each equation makes sense
- Airlines: minimize costs, crew movement and plane takeoffs and landings, subject to regulatory constraints
- Traders: maximize returns for a given level of risk
- Some other physically measurable quantity

Algebraic Distance

- Algebraic system $\mathbf{Ax} = \mathbf{b}$
- Approximate solution $\mathbf{x}^{/}$
- Residual $\|\mathbf{A}\,\mathbf{x}^{/} - \mathbf{b}\|$
- Residual is also called algebraic distance
- Algorithms that seek to reduce the residual are called "minimum residual" algorithms



Properties of the Algebraic Distance

- Each row in a linear equation can be multiplied by an arbitrary number

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

is the same as

$$c(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n) = cb_1$$

However given an approximate solution $\tilde{\mathbf{x}}$

$$a_{11}\tilde{x}_1 + a_{12}\tilde{x}_2 + a_{13}\tilde{x}_3 + \cdots + a_{1n}\tilde{x}_n - b_1$$

is not the same as

$$c\left(a_{11}\tilde{x}_1 + a_{12}\tilde{x}_2 + a_{13}\tilde{x}_3 + \cdots + a_{1n}\tilde{x}_n - b_1\right)$$

Scaling

- Try to avoid anyone equation being overly represented.
- Scale each equation
  - Scale by largest coefficient so that it becomes 1

$$a_{i1}/a_{11}$$

  - Scale so that sum of coefficients is 1

$$a_{11}^2 + a_{12}^2 + \ldots + a_{1n}^2 = 1$$

- Scaling also has the benefit of avoiding round-off.

Weighted Least Squares

- Multiplying an equation by a number will increase its ***weight*** or ***influence*** in the cost function.
- Not always a bad thing
  - May want to weight different equations differently
- How to select weights?
  - Number of observations
  - Reliability of measurement
    - Measured variances
- How good is the least squares solution? How "probable" are the parameter estimates?
- ***Bring in notions of statistics***

Maximum Likelihood Parameter Estimation

- *likelihood* of the parameters given the data
- Least squares fit is a “maximum likelihood estimator”
- Assume
  - y_i has a measurement error that is normally distributed around true *y*(*x*). $\exp\left[-\frac{1}{2}\left(\frac{y_i - y(x_i)}{\sigma}\right)^2\right]$
  - Assume errors are independent, and standard deviations σ of all these normal distributions are the same.
  - Then probability that the data set and the model predictions are within Δ*y* the product of that of each other is

$$P \propto \prod_{i=1}^{N} \left\{ \exp\left[-\frac{1}{2}\left(\frac{y_i - y(x_i)}{\sigma}\right)^2\right] \Delta y \right\}$$

- Maximize likelihood that parameters are correct by maximizing *P* with respect to model parameters.

Least Squares = MLE

•Since logarithm is a monotonic increasing function maximum of log P is the maximum of P

$$\log P = \left[\sum_{i=1}^{N} \frac{[y_i - y(x_i)]^2}{2\sigma^2}\right] - N \log \Delta y$$

•Maximizing log *P* is equivalent to maximizing the least squares criterion $(y_i - y(x_i))^2$ since the other terms are constants

•What to do when variances are not all the same?

- •Maximize the Mahalanobis distance

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{y_i - y(x_i; a_1 \ldots a_M)}{\sigma_i} \right)^2$$

•Here the errors were just assumed to be in the measured *y*s

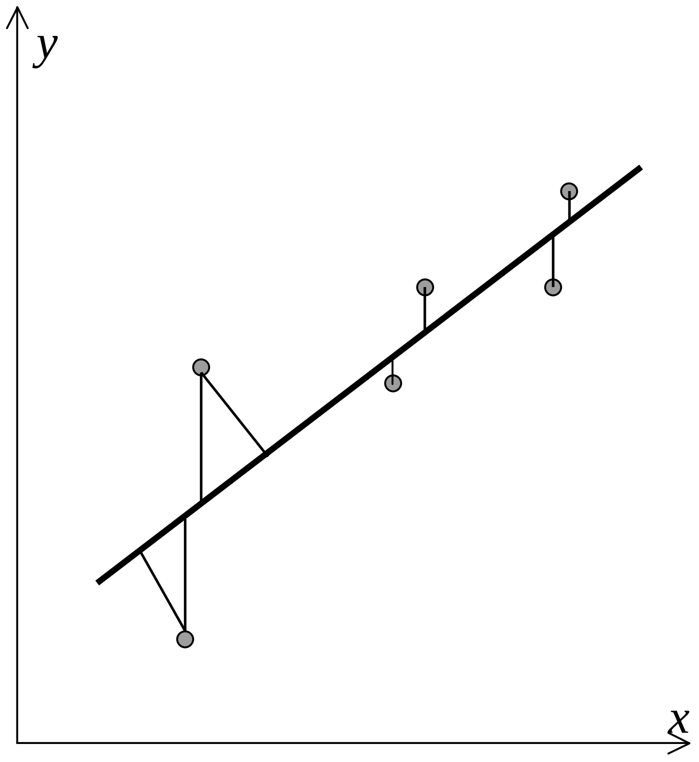
Errors in both coordinates

- Often in computer vision measurements are made in both images and a relationship between them must be deduced.
- Consider the line fit example again
  - Intuitively distances should be perpendicular to the line
- Perpendicular distance between point (x_i, y_i) and a line $y-a-bx=0$ is

$$d(x_i, y_i; a, b) = \left[\frac{(y_i - a - bx_i)^2}{1+b^2}\right]^{1/2}$$

- If x_i and y_i are distributed normally with standard deviations σ_{xi} and σ_{yi} we can show that c is

$$\chi^2(a,b) = \sum_{i=1}^{N} \frac{(y_i - a - bx_i)^2}{\sigma_{y\,i}^2 + b^2\sigma_{x\,i}^2}$$



- Makes the cost function nonlinear in parameters
  - Nonlinear in *b*. In general physical error functions are nonlinear.

Cost functions for *image* based data

- Notation
  - Measured value of a point $\mathbf{x}^{\sim}$
  - True value of a point $\mathbf{x}$
  - Estimated value of a point $\mathbf{x}^{\wedge}$
  - Transformation or model is denoted H
    - Model $\mathbf{y}=H(\mathbf{x})$ and $\mathbf{x}=H^{-1}(\mathbf{y})$
- Symmetric error functions
  - Case 1: Error only in one image
    - Could arise if we are imaging a calibration pattern with known coordinates and trying to determine camera calibration
  - Appropriate error function is

$$\text{Find } \mathbf{H}^{\wedge} \text{ that minimizes} \qquad \Sigma_j \, d(\mathbf{x}'^{\sim}{}_j, \mathbf{H}^{\wedge}\mathbf{x}_j^{\sim})^2$$

Cost functions for image data

- Errors in both images

  Find $\mathbf{H}^\wedge$ that minimizes $\Sigma_j\, d(\mathbf{x}'_j{}^\sim, \mathbf{H}^\wedge \mathbf{x}_j{}^\sim)^2 + d(\mathbf{H}^{\wedge\,-1}\mathbf{x}'_j{}^\sim, \mathbf{x}_j{}^\sim)^2$

- Reprojection error
  - Instead of determining parameters of a transformation that minimizes distances on erroneous data, find corrections to the wrong data and find the transformation that maps corrected data.
  - Get estimates $\mathbf{x}^\wedge$ and $\mathbf{x}'^\wedge$ such that

    $\Sigma_j\, d(\mathbf{x}_j{}^\sim, \mathbf{x}_j{}^\wedge)^2 + d(\mathbf{x}'_j{}^\sim, \mathbf{x}_j{}^\wedge)^2$ subject to $\mathbf{H}^\wedge \mathbf{x}_j{}^\wedge = \mathbf{x}'_j$

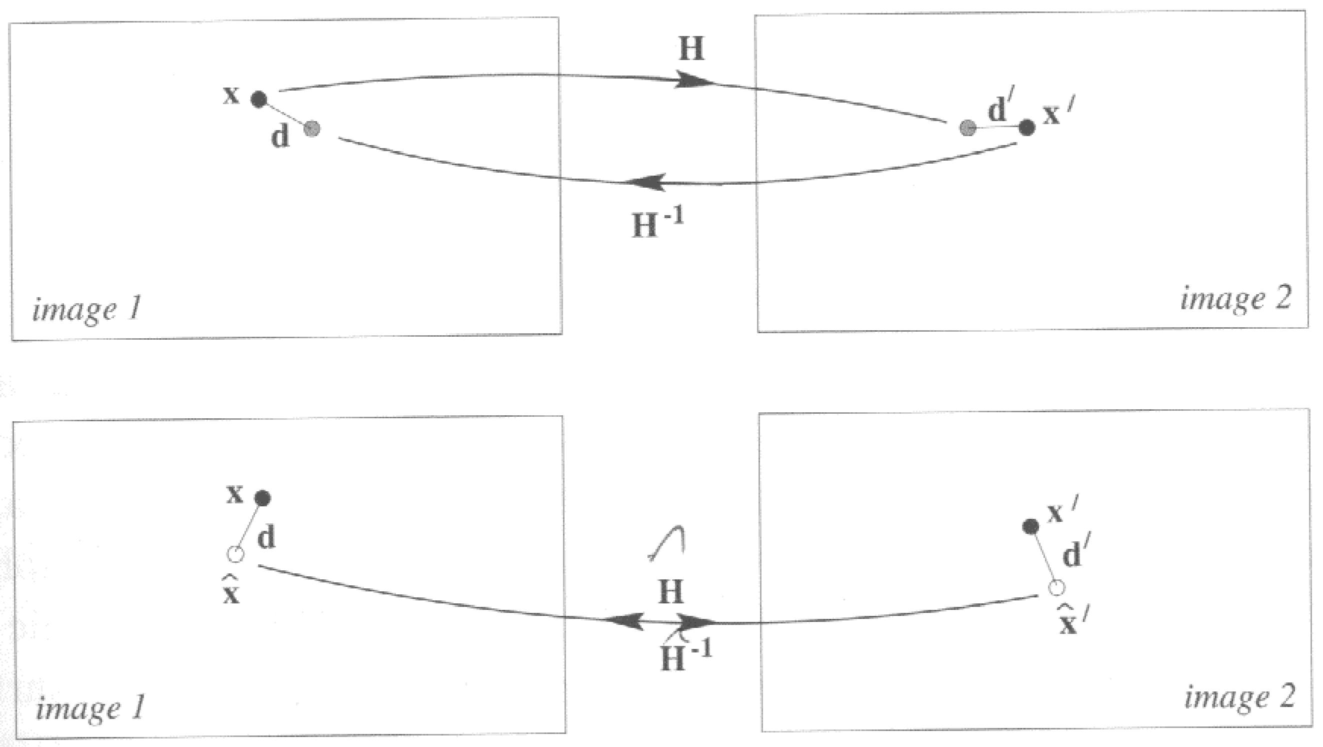


Fig. 3.2. *A comparison between symmetric transfer error (upper) and reprojection error (lower) when estimating a homography. The points* $\mathbf{x}$ *and* $\mathbf{x}'$ *are the measured (noisy) points. Under the estimated homography the points* $\mathbf{x}'$ *and* $\mathtt{H}\mathbf{x}$ *do* not *correspond perfectly (and neither do the points* $\mathbf{x}$ *and* $\mathtt{H}^{-1}\mathbf{x}'$*). However, the estimated points,* $\hat{\mathbf{x}}$ *and* $\hat{\mathbf{x}}'$*, do correspond perfectly by the homography* $\hat{\mathbf{x}}' = \mathtt{H}\hat{\mathbf{x}}$*. Using the notation* $d(\mathbf{x}, \mathbf{y})$ *for the Euclidean image distance between* $\mathbf{x}$ *and* $\mathbf{y}$*, the symmetric transfer error is* $d(\mathbf{x}, \mathtt{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathtt{H}\mathbf{x})^2$*; the reprojection error is* $d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2$.

Optimization Techniques

- Different problem here
  - Given a set of locations $\mathbf{x}_i$ where one has measured a fitness function *χ²/ f↼x↽ find a vector of parameters ↼x↽ that minimizes it*
- *For the case where the function was linear we already have methods such as SVD to solve the linear system▷*
- *Here we are concerned with systems where the equation is not so simple▷*
  - *In particular f may be a nonlinear function of parameters x*
- *Differential calculus provides us with ways of estimating extrema▷*
  - *The minimum ↼max↽ of f occurs at ∇f / oᶜ or*
  - *∇f is in the direction of increasing fᶜ or*
  - *Given an interval ∇f has opposite signs at the boundary there must be a point inside where ∇f must be zero*
- *However calculus is local*
  - *So these methods can only guarantee a local extremum*

Bisection methods

- Given a function *f* at three points *a*,*b*,*c with [a<b<c]*, and a way to evaluate *f* at a new point
  - Given 2 initial guesses *f(a)* and *f(b)*, if *f(a)>f(b)* move in the direction *a* to *b* and choose a new parameter *c*.
  - *Find a triplet [a,b,c] so that* and *f(c)>f(b) and f(a)>f(b)*
  - Choose a new point between *a* and *b* or *b* and *c*
  - Repeat until the points *a*, *b* and *c* are sufficiently close

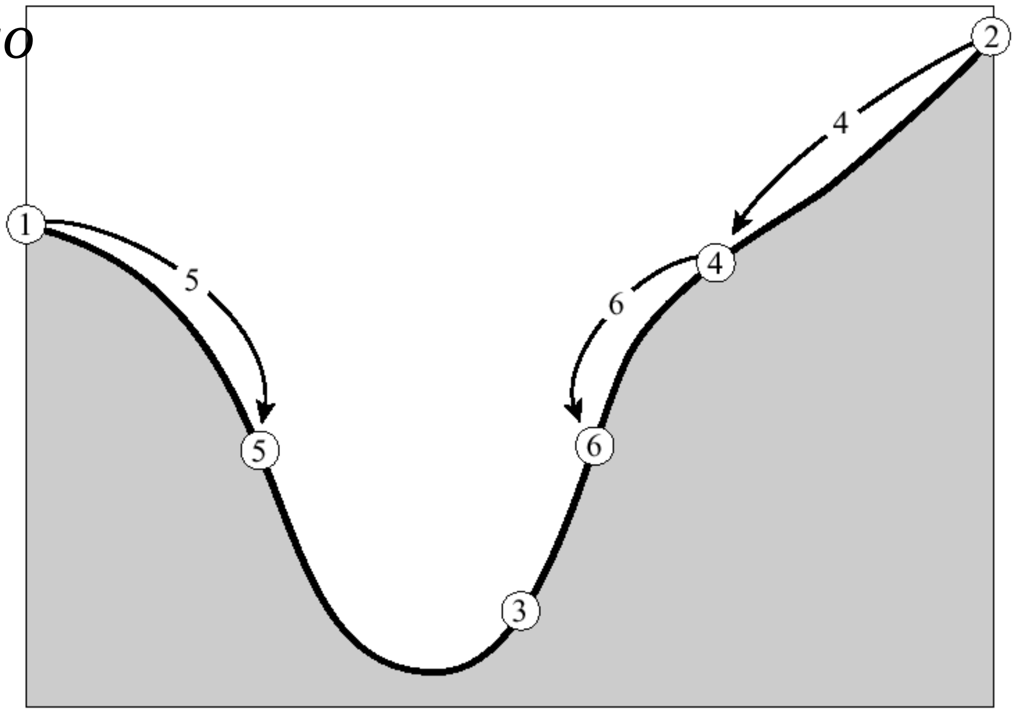


Figure 10.1.1. Successive bracketing of a minimum. The minimum is originally bracketed by points 1,3,2. The function is evaluated at 4, which replaces 2; then at 5, which replaces 1; then at 6, which replaces 4. The rule at each stage is to keep a center point that is lower than the two outside points. After the steps shown, the minimum is bracketed by points 5,3,6.

Paraboloic bracketing

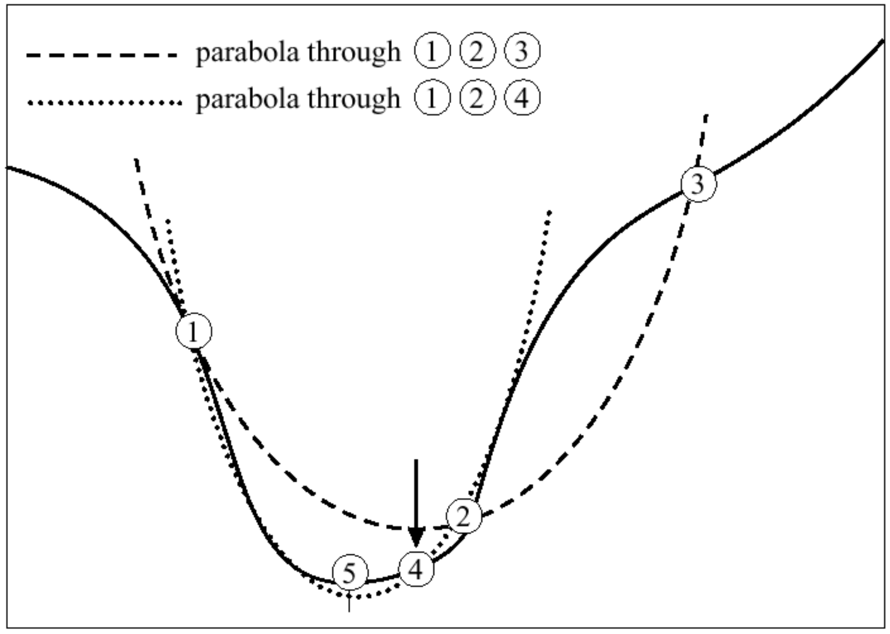


Figure 10.2.1. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1,2,3 on the given function (solid line). The function is evaluated at the parabola's minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1,4,2. The minimum of this parabola is at 5, which is close to the minimum of the function.