Robust Methods

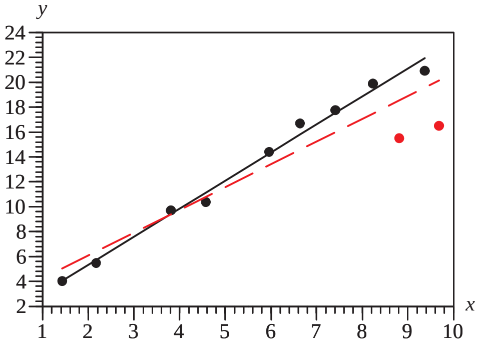
CS6240 Multimedia Analysis

Leow Wee Kheng

Department of Computer Science
School of Computing
National University of Singapore

Introduction

Consider the following linear fitting problem:



- black dots: inliers, black line: best fit of inliers
- red dots: outliers, red line: best fit of inliers and outliers
- Outliers can seriously shift the best fitting line.

Robust Methods

- Robust algorithms can identify and ignore outliers.
- Robust error functions are not affected by outliers.
- Robust statistics [HRRS86, Hub81, MMRK91]: Statistical estimation methods that are not affected by outliers.

Measure of Robustness: Breakdown Point

- The proportion of outliers required to increase the error by an arbitrarily large amount.
- Robust Statistics Theory: Maximum breakdown point is 50%.
- If fraction of outliers $> 50\%$, then for any error function, outliers can be aligned to produce a better fit than inliers.

Illustration of Robust Methods:

- Use least square fit as an example problem.
- Many other problems, e.g., registration, feature extraction, and model fitting, can be formulated in the same way [FB81, MMRK91, RL93, Ste99].

Least Square Fit

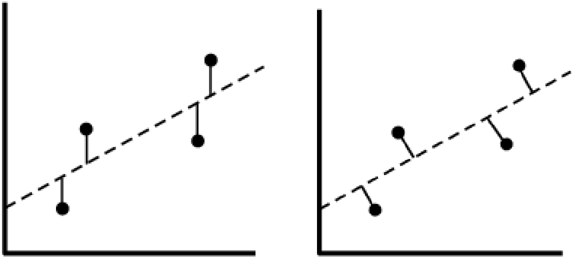
- Given a set of n points (x_i, y_i), $i = 1, \ldots, n$.
- Fit function f through the points so that $f(x_i; a_1, \ldots, a_m) = y_i$.
- This equation is the model that describes the points.
- f can represent a line or a curve.
- This fitting problem can be defined as the problem of finding the parameters $a_1, \ldots, a_m$ that minimize the sum square error E

$$E = \sum_{i=1}^{n} [f(x_i; a_1, \ldots, a_m) - y_i]^2 . \tag{1}$$

- The error $r_i = f(x_i; a_1, \ldots, a_m) - y_i$ is also called the residual.
- Multi-dimensional (multi-variate) case:

$$E = \sum_{i=1}^{n} \|\mathbf{f}(\mathbf{x}_i; \mathbf{a}) - \mathbf{y}_i\|^2 . \tag{2}$$

- The error r_i is the vertical error.



(a) vertical error (b) perpendicular error

- Perpendicular distance is more appropriate in some applications. But, it is more difficult to compute for nonlinear case.
- See Appendix A for details of computing perpendicular distance.

- An easier way to define f is to use implicit function:

$$f(x_i, y_i; a_1, \dots, a_m) = 0 \tag{3}$$

and minimize the error E

$$E = \sum_{i=1}^{n} f^2(x_i, y_i; a_1, \dots, a_m)\,. \tag{4}$$

- Example implicit functions:

shape	implicit function
line	$a_1 x + a_2 y + a_3 = 0$
quadratic	$a_1 x^2 + a_2 xy + a_3 y^2 + a_4 x + a_5 y + a_6 = 0$

- Actually, need 1 fewer parameter, e.g., $a_1 x + a_2 y + 1 = 0$.

- Multi-dimensional case:
  Convenient to express in terms of basis function b_j of point $\mathbf{x}$:

$$f(\mathbf{x};\mathbf{a}) = \sum_{j=1}^{m} a_j b_j(\mathbf{x})\,. \tag{5}$$

- Example basis functions [RL93]:

shape	basis functions
line	$x,\ y,\ 1$
conic	$x^2,\ y^2,\ xy,\ x,\ y,\ 1$
plane	$x,\ y,\ z,\ 1$
quadric	$x^2,\ y^2,\ z^2,\ xy,\ yz,\ zx,\ x,\ y,\ z,\ 1$

- Other basis functions: Gaussians, splines, etc.

Robust Error Function

Let r_i denote the residual (error) of point $\mathbf{p}_i$, $i = 1, \ldots, n$.

Typical error (cost) functions [RL93]:

- Least squares: $E(r_1, \ldots, r_n) = \displaystyle\sum_{i=1}^{n} r_i^2$.
- M-estimation: $E(r_1, \ldots, r_n) = \displaystyle\sum_{i=1}^{n} \rho(r_i^2)$, where ρ is a symmetric positive-definite function with a unique minimum at zero.
- Least median of squares: $E(r_1, \ldots, r_n) = \mathrm{median}(r_1^2, \ldots, r_n^2)$.

Properties of error functions:

- $E \geq 0$ for all possible parameters $\mathbf{a}$.
- $E = 0$ when all $r_i = 0$.

Robustness:

- Least squares error function is not robust.
  Breakdown point is $1/n$; just one point!
- M-estimation is robust:
  ρ is chosen so that points with large errors (possible outliers) have small influence so they can be ignored.
- Least median of squares is robust.
  Breakdown point is 50%.

Notes:

- Robust error functions can have many local minima [RL93].
- Need efficient algorithm to find global minimum.

Outlier Detection

- Inlier: Point that is taken into account by the error function E.
- Outlier: Its residual can be increased to any arbitrary value without affecting the error E (provided that E is robust).
- The best partition of points into inliers and outliers is the one at which the error E is the global minimum.

Outlier Detection [RL93]

- For a given $\mathbf{a}$, compute r_i for all points and $E(r_1, \ldots, r_n)$.
- For a particular point $\mathbf{p}_i$, set its residual r_i to ∞, and recompute $E(r_1, \ldots, r_n)$.
- If E remains the same, then $\mathbf{p}_i$ is an outlier.
  Otherwise, it is an inlier.

Robust Algorithms

Basic Ideas:

- Randomly select a subset of points and compute error.
- If subset contains only inliers, error should be small.
- If the model to be fitted is known, e.g., line, quadratic curve, then the minimum subset size can be determined.

Random Sample Consensus (RANSAC)

- RANSAC: A robust method for robust fitting [FB81].
- Given a set P of n points.
- Suppose at least $m < n$ points are needed to compute error E.

RANSAC Algorithm [FB81]

Repeat

1. Randomly select a subset S of m points to determine the best fitting model M (parameter vector $\mathbf{a}$).
2. Use model M to determine a subset S' of P within some error tolerance τ of M. S' is called the consensus set of S.
3. If $|S'| >$ threshold Γ, keep S'.

Use the largest consensus set to compute the final model.

RANSAC has three parameters that need to be specified:

- The error tolerance τ.
  Dependent on the expected error of the model when fitting inliers.
- The number of subsets to try in the repeat loop.
  Dependent on the expected number of trials required to select a subset of good data points.
- The threshold Γ.
  Dependent on the model. Should be large enough to ensure that the consensus set contains enough inliers for a good fit.

Minimal Subset Random Sampling

- A generalization of RANSAC [RL93].
- Use the idea of minimal subset of points required to fit the model.

Given a set of n points and an error function E.

For k randomly selected sets of m points:

1. Find the parameter vector $\mathbf{a}$ of the model through all the m points.
2. Compute the residuals r_i of all n points with respect to $f(\mathbf{x}_i; \mathbf{a})$.
3. Rank the goodness of this model by evaluating the error function $E(r_1, \ldots, r_n)$. E indicates the goodness of fit.
4. Save the model with the smallest error along with the associated $\mathbf{a}$.

Complexity of algorithm is at best $O(kn)$.

Selection of k

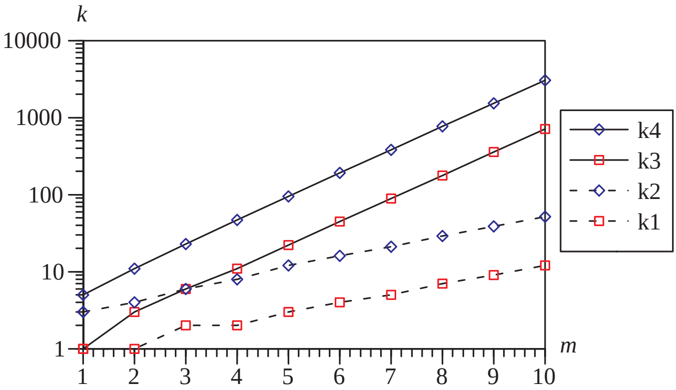
- Maximum value of k is ${}^{n}C_{m}$; very large.
- In practice, a smaller value is possible.
- Let ε be the probability that a randomly selected point $\mathbf{p}_i$ is an inlier.
- The probability that all the m points in the minimal subset are inliers is ε^m.
- Then, the probability s that at least one of the k minimal subsets contains all inliers is

$$s = 1 - (1 - \varepsilon^m)^k. \tag{6}$$

- The value k as a function of ε and m is

$$k = \frac{\ln(1 - s)}{\ln(1 - \varepsilon^m)}. \tag{7}$$

- k is small for large ε, small s, small m.

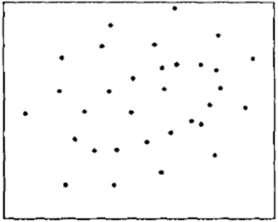


- Solid lines (k3, k4): $\varepsilon = 0.5$
- Dashed lines (k1, k2): $\varepsilon = 0.75$
- Blue diamonds (k2, k4): $s = 0.95$
- Red squares (k1, k3): $s = 0.5$

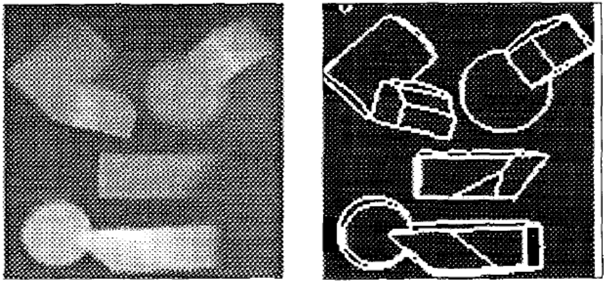
Notes:

- According to Robust Statistics, maximum breakdown point is 50%. This is for the general case.
- In many applications, properties of inliers can be used to distinguish between inliers and outliers.
- Then, even if fraction of outliers $> 50\%$, robust algorithms can still pick out the outliers.
- In some applications, can use properties of inliers to select subset of m points systematically instead of randomly.
- Can also extend the algorithm to fit multiple models to different subsets of the data points [RL93].

Example: Fit an ellipse to a set of points [RL93].



Example: Fit multiple shapes to a set of edges [RL93].



Appendix A.1
Perpendicular Distance to a Plane

A plane π in 3-D space can be parameterized by the equation

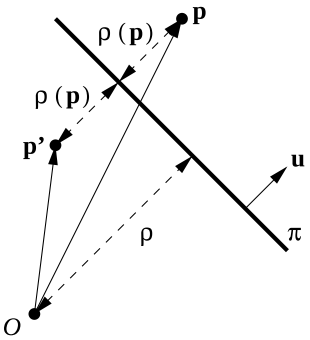
$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0 = 0 \tag{8}$$

where w_0 and $\mathbf{w} = (w_1, w_2, w_3)$ are the parameters of the plane, and $\mathbf{x}$ is any 3D point on the plane. Consider any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ lying on the plane. From Eq. 8, we obtain

$$\mathbf{w} \cdot (\mathbf{x}_2 - \mathbf{x}_1) = 0 \tag{9}$$

which means that $\mathbf{w}$ is normal to the plane. Thus, the plane's unit normal vector $\mathbf{u}$ is given by $\mathbf{w}/\|\mathbf{w}\|$.

The perpendicular (signed) distance ρ of the plane from the origin is given by $\mathbf{u} \cdot \mathbf{x}$ for any point $\mathbf{x}$ on the plane. That is, $\rho = -w_0/\|\mathbf{w}\|$.



Now, consider a point $\mathbf{p}$ not on the plane. From the figure above, it is obvious that the perpendicular (signed) distance of $\mathbf{p}$ to the plane π, denoted as $\rho(\mathbf{p})$, is

$$\rho(\mathbf{p}) = \mathbf{p} \cdot \mathbf{u} - \rho = \frac{\mathbf{w} \cdot \mathbf{p} + w_0}{\|\mathbf{w}\|} = \frac{f(\mathbf{p})}{\|\mathbf{w}\|}. \tag{10}$$

Appendix A.2
Perpendicular Distance to a Curve Surface

There is no closed form formula for computing the perpendicular distance of a point to a general curve surface.

A first-order approximation of the perpendicular (signed) distance $\rho(\mathbf{p})$ of a point $\mathbf{p}$ to a surface f is given by [RL93]

$$\rho(\mathbf{p}) = \frac{f(\mathbf{p})}{\|\nabla f(\mathbf{p})\|}. \tag{11}$$

In the case of a plane, Eq. 11 becomes Eq. 10.

Reference I

- M. A. Fischler and R. C. Bolles.
  Random sample consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography.
  Communications of ACM, 24(6):381–395, 1981.

- F. R. Hampel, P. J. Rousseeuw, E. Ronchetti, and W. A. Stahel.
  Robust Statistics: The Approach Based on Influence Functions.
  Wiley, 1986.

- P. Huber.
  Robust Statistics.
  Wiley, 1981.

- P. Meer, D. Mintz, A. Rosenfeld, and D. Kim.
  Robust regression methods in computer vision: A review.
  Int. Journal of Computer Vision, 6(1):59–70, 1991.

Reference II

- G. Roth and M. D. Levine.
  Extracting geometric primitives.
  CVGIP: Image Understanding, 58(1):1–22, 1993.

- C. V. Stewart.
  Robust parameter estimation in computer vision.
  SIAM Review, 41(3):513–537, 1999.