Least Squares Fitting of an Ellipse

Daniel Coble

July 28, 2023

1 Abstract

I describe how an ellipse can be fit to pointcloud data, and discuss some consequences.

2 Fitting an Ellipse

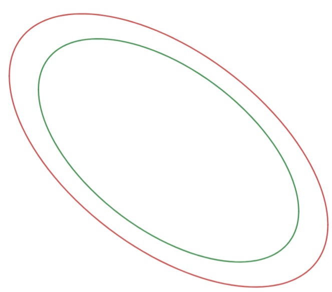
Say we have from experimentation a set of points $\{(x_i, y_i)\}$. Consider a general conic

$$0 = ax^2 + bxy + cy^2 + dx + ey + f \tag{1}$$

The *algebraic distance* from of a point to this conic is

$$F(x, y) = ax^2 + bxy + c^2 + dx + ey + f \tag{2}$$

So, all points on the conic have algebraic distance 0 (as expected). Considering an ellipse, points within the ellipse have negative distance and points outside the ellipse have positive distance. Algebraic distance is also not geometric/Euclidean distance to the closest point on the ellipse; see the figure below which shows an ellipse (green) and all points with algebraic distance 1 from the ellipse (red). Points near major axis are further away (geometrically) than those near the minor axis.



Let $\mathbf{a} = [a\ b\ c\ d\ e\ f]^T$. We would like to create a least squares fit which finds the ellipse with the least distance to the point cloud. But applying that right away won't work, since with $\mathbf{a} = 0$, all points on the plane are have distance 0. To avoid the trivial solution it is necessary to put a constraint on $\mathbf{a}$. Taking advantage of the fact that scaling $\mathbf{a}$ will not affect the conic (1), we can enforce a constraints are $f = 1$ and $a + c = 1$. In section 3 I discuss choice of constraint. There's also the quadratic constraint $\|\mathbf{a}\|^2 = 1$. I'll carry out computations below for the $f = 1$ and $a + c = 1$ cases.

2.1 $f = 1$

Since f is constrained let $\mathbf{a} = [a\ b\ c\ d\ e]^T$ We want to minimize

$$C(\mathbf{a}) = \sum_{i=0}^{N} \left(ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + 1\right)^2 \tag{3}$$

This can be rearranged into matrix form. Let D be the matrix with rows $\left[x_i^2\ x_iy_i\ y_i^2\ x_i\ y_i\right]$ for each i (called the design matrix). C can be expressed as

$$C(\mathbf{a}) = (D\mathbf{a} + \mathbf{1})^T (D\mathbf{a} + \mathbf{1}) \tag{4}$$

Where $\mathbf{1}$ is a n-vector of 1's. At the minimum the gradients are zero

$$\frac{\partial C}{\partial \mathrm{a}} = 2\mathrm{a}^T D^T D + 2\mathbf{1}^T D = 0 \tag{5}$$

And rearranging, becomes

$$\mathbf{a} = -\left(D^T D\right)^{-1} D^T \mathbf{1} \tag{6}$$

2.2 $a + c = 1$

Let $\mathbf{a} = [a\ b\ d\ e\ f]^T$. The cost function is

$$\begin{aligned} C(\mathbf{a}) &= \sum_{i=0}^{N} \left(ax_i^2 + bx_iy_i + (1-a)y_i^2 + dx_i + ey_i + f\right)^2 && (7)\\ &= \sum_{i=0}^{N} \left(a\left(x_i^2 - y_i^2\right) + bx_iy_i + dx_i + ey_i + f + y_i^2\right)^2 && (8) \end{aligned}$$

Define the vector $\mathbf{y^2}$ to be the n-vector of all y_i^2, and the matrix D to have rows $\left[x_i^2 - y_i^2\ x_iy_i\ x_i\ y_i\ 1\right]$. The matrix form is

$$C(\mathbf{a}) = \left(D\mathbf{a} + \mathbf{y^2}\right)^T \left(D\mathbf{a} + \mathbf{y^2}\right) \tag{9}$$

And by the same steps as in 2.1, the solution is

$$\mathbf{a} = -\left(D^T D\right)^{-1} D^T \mathbf{y^2} \quad ; \quad c = 1 - a \tag{10}$$

3 Discussion

The method shown above is simple but has the two drawbacks. (1) It solves for a general conic rather than an ellipse; and; (2) we must constrain the solution to avoid a trivial case. Fitzgibbon et al. [1] resolved both these issues with their method (which was implemented in this blog post). In short, it uses the determinant constraint of an ellipse $b^2 - 4ac < 0$ in place of the constraint, which neatly fixes both issues. That, like the constraint on the norm $\|\mathbf{a}\|^2 = 1$ is a quadratic constraint, and can be solved with Lagrange multipliers.

An investigation into the effect of the constraints was done in [2] (which argues that $f = 1$ is better than $a + c = 1$ for most instances), but considering the existence and choice between constraints, there's no way to argue that any choice of method produces the *best fit* ellipse. What we intuitively desire is the use of a *geometric distance* to an ellipse (the Euclidean distance to the closest point on the ellipse), and that can be done with gradient descent methods, but isn't a linear least squares problem.

References

[1] Fitzgibbon, A., Pilu, M., and Fisher, R. B., "Direct least square fitting of ellipses," *IEEE Transactions on pattern analysis and machine intelligence* **21**(5), 476–480 (1999).

[2] Rosin, P. L., "A note on the least squares fitting of ellipses," *Pattern Recognition Letters* **14**(10), 799–808 (1993).