CHAPTER 3

Introduction to Statistical Estimation

1. The Simplest Kind of Statistics

Least Squares is the simplest and most intuitive kind of statistics and often the most useful. The most straightforward application is as follows. We have a set of quantities that we would like to be zero or as close to zero as possible and they all depend on a set of unknowns. We take the squares of all these quantities, sum them up and then minimize this sum with respect to the unknowns. There are many alternatives to least squares that sometimes have interesting properties (most notably robustness to outliers) but least squares is not only the simplest but is also the basis for most of the alternatives.

2. Point in the Middle

Consider the following very simple problem. We want to find a point P and all we have is a set of several approximations of P which we call P_i, $i = 1..N$. If of course all the P_is are identical the choice is easy. Otherwise we would like P to be as close to all of them as possible. We form the sum of the squared differences

$$Q(P) = \sum_{i=1}^{N}(P_i - P)^2.$$

The standard way to minimize Q is to take its derivatives with respect to the unknowns and equate them to zero. Solving these equations will give us P, the vector of the unknowns. In this very simple problem solving the equations is easy, but taking the derivatives is slightly more complex. We examine two ways to take these derivatives. One is scalar (element by element) derivatives and the other is vector derivatives.

2.1. Scalar Derivatives

Our unknowns are the elements of the vector P

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_K \end{bmatrix}$$

and our data are the vectors P_i

$$P_i = \begin{bmatrix} p_{i1} \\ p_{i2} \\ \cdots \\ p_{iK} \end{bmatrix}$$

So

$$Q(P) = \sum_{i=1}^{N} \sum_{j=1}^{K} (p_{ij} - p_j)^2$$

and

$$\frac{\partial Q(P)}{\partial p_k} = \sum_{i=1}^{N} \sum_{j=1}^{K} \frac{\partial}{\partial p_k} (p_{ij} - p_j)^2 =$$

$$-2 \sum_{i=1}^{N} \sum_{j=1}^{K} \frac{\partial p_j}{\partial p_k} (p_{ij} - p_j) = -2 \sum_{i=1}^{N} \sum_{j=1}^{K} \delta_{jk} (p_{ij} - p_j)$$

where δ_{ij} is the Kronecker delta, e.g. $\delta_{ij} = 0$ iff $i \neq j$ and $\delta_{ii} = 1$ which of course makes perfect sense: the derivative of an unknown with respect to itself is equal to one and the derivative of an unknown with respect to a different unknown is zero. The delta affords us some simplifications, so

$$\frac{\partial Q(P)}{\partial p_k} = -2 \sum_{i=1}^{N} (p_{ik} - p_k)$$

which if we equate to zero we get

$$p_k = \frac{\sum_{i=1}^{N} p_{ik}}{N}$$

e.g. every element of the unknown vector is the average of the corresponding elements of the data.

2.2. Vector Derivatives

A more compact and mainly more elegant way of doing the same thing is taking vector derivatives. Most of the rules of scalar derivatives apply, some with a small quirk. Let's start.

The notation

$$\frac{\partial Q(P)}{\partial P}$$

indicates a vector whose elements are the scalar derivatives of $Q(P)$ with respect to the corresponding element of P (remember Q is a scalar). Sometimes the "grad" notation is used to indicate the same thing

$$\frac{\partial Q(P)}{\partial P} = \nabla_P Q(P)$$

where the subscript P is the vector with respect to which the derivatives are taken. If it is obvious what this vector is (in many physics problems it is always the position vector) it is omitted. So

$$\frac{\partial Q(P)}{\partial P} = \frac{\partial}{\partial P} \sum_{i=1}^{N} (P_i - P)^2 = \sum_{i=1}^{N} \frac{\partial}{\partial P} \left((P_i - P)^T (P_i - P) \right) =$$
$$2 \sum_{i=1}^{N} \left(\frac{\partial}{\partial P} (P_i - P)^T \right) (P_i - P) = -2 \sum_{i=1}^{N} \left(\frac{\partial}{\partial P} P^T \right) (P_i - P)$$

where the derivative of P_i is zero, because it is constant. The derivative of a row vector with with respect to a column vector is a matrix. Every row of this matrix is the derivative of the row vector with the corresponding element of the column vector. In our case the derivative of P with respect to itself is the identity matrix **1**. So

$$\frac{\partial Q(P)}{\partial P} = 2 \sum_{i=1}^{N} \mathbf{1} (P_i - P) = 2 \sum_{i=1}^{N} (P_i - P)$$

which if we equate to zero we get

$$P = \frac{\sum_{i=1}^{N} P_i}{N} \tag{2.1}$$

which is essentially the same as before.

3. Line Fitting

In the problem above we had a collection of points P_i and we found a point P that is closest to all of them, in the least squares sense. We can try something slightly more complex now, like finding a line l that is closest to all points P_i. Let line l be represented by two vectors

$$l = (p, q)$$

where a point P_i belongs to l iff there is a λ such that $p + \lambda q = P_i$. Vector p is a point on the line and vector q is the direction of the line. There are other ways to represent a line but this one suits, for the time, our purpose better. It works equally well for lines in the plane or in a space of arbitrary number of dimensions. But most important, allows us to explore a few interesting concepts.

Since we want to minimize the distance of the points P_i from the line l we first need to express this distance as a nice and easy to use expression. There are two equivalent ways to define the point to line distance. The one is to define a normal line that goes through the point and intersects the line l at a right angle and then measure the distance of the point from the line l along the normal line. The second way is to find the distance of the point P_i from a point P'_i that lies on the line l and then slide the point P'_i along the line l till this distance is minimized. This minimal distance is the one we want. Since we have the machinery to minimize things we opt for the second approach. If you have a hammer everything looks like a nail.

So the (squared) distance D_i^2 of the point P_i from the line l is

$$D_i^2 = \min_{\lambda}(P_i - p - \lambda q)^2$$

but if we want to do anything useful with it we have to get rid of the min symbol by finding the minimizing with respect to λ. As before we take derivatives

$$\frac{\partial(P_i - p - \lambda q)^2}{\partial\lambda} = -2q^T(P_i - p - \lambda q)$$

and by equating it to zero we get

$$\lambda = \frac{q^T(P_i - p)}{q^T q}$$

which gives us the expression for distance

$$D_i^2 = \left((P_i - p) - \frac{qq^T}{q^T q}(P_i - p)\right)^2$$

which, striving for elegance we rewrite as

$$D_i^2 = \left((\mathbf{1} - \frac{qq^T}{q^T q})(P_i - p)\right)^2. \tag{3.1}$$

Now we have an expression for the distance of a point P_i from the line l and it is already squared. To proceed with our least squares we sum up all these squared distances and find the line parameters p and q that minimize this sum. We start by defining the sum

$$Q(p, q) = \sum_{i=1}^{N} D_i^2$$

and we take the derivatives first with respect to p

$$\frac{\partial Q(p, q)}{\partial p} = \sum_{i=1}^{N} \frac{\partial D_i^2}{\partial p} = -2\sum_{i=1}^{N} \frac{\partial p^T}{\partial p}(\mathbf{1} - \frac{qq^T}{q^T q})(\mathbf{1} - \frac{qq^T}{q^T q})(P_i - p) =$$

$$-2\sum_{i=1}^{N}(\mathbf{1} - \frac{qq^T}{q^T q})(\mathbf{1} - \frac{qq^T}{q^T q})(P_i - p) =$$

$$-2(\mathbf{1} - \frac{qq^T}{q^T q})(\mathbf{1} - \frac{qq^T}{q^T q})\sum_{i=1}^{N}(P_i - p)$$

which we can equate to zero. We can verify that

$$p = \frac{\sum_{i=1}^{N} P_i}{N} \tag{3.2}$$

satisfies the resulting equation, although it is not a unique solution. Since p is just a point on the line it is as good as any other point on the line. But we prefer the one given by Eq. (3.2) because it is identical to the one given by Eq. (2.1) and does not contain q.

On to q now. It appears that Eq. (3.1) is not elegant enough and we should improve it. Consider the following well known identities for a vector v

$$v^2 = v \cdot v = v^T v = tr(vv^T)$$

where $tr(\cdots)$ is the trace operator (sum of diagonal elements). We are mainly interested in the last version to apply it to Eq. (3.1) which becomes

$$Q(p,q) = \sum_{i=1}^{N} tr\left(\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)(P_i - p)(P_i - p)^T\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\right)$$

and by noticing that the trace operator is linear and that the first and last parenthesized quantities do not depend on the index i we can rewrite it as

$$Q(p,q) = tr\left(\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\left(\sum_{i=1}^{N}(P_i - p)(P_i - p)^T\right)\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\right) =$$

$$N\ tr\left(\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)C\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\right) = N\ tr\left(\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)C\right)$$

where

$$C = \frac{\sum_{i=1}^{N}(P_i - p)(P_i - p)^T}{N}$$

and we applied the following property of the trace for two matrices A and B of appropriate dimensions

$$tr(AB) = tr(BA)$$

Then noticing that the product

$$\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right) =$$

$$\mathbf{1} + \frac{qq^Tqq^T}{q^Tqq^Tq} - \frac{qq^T}{q^Tq} - \frac{qq^T}{q^Tq} =$$

$$\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)$$

$$Q(p,q) = N\ tr\left(C\left(\mathbf{1} - \frac{qq^T}{q^Tq}\right)\right)$$

and by invoking the above property of the trace we get

$$Q(p,q) = N\ tr(C) - N\ tr\left(\frac{q^TCq}{q^2}\right) \tag{3.3}$$

which is undoubtedly the most elegant of equations.

Matrix C is a constant so it does not affect the minimization which can be achieved by maximizing the scalar

$$\frac{q^TCq}{q^2}$$

which is just a Rayleigh Quotient and is maximized when q is the eigenvector that corresponds to the largest eigenvalue.

There are a few remarks that we can make on this result. First, we find the line direction q without taking derivatives, just by using a canned theorem (we, in other words, outsourced the derivatives to Dr. Rayleigh). Second the points P_i can be of any dimension: two dimensional points on the image plane, three dimensional points in the real world or ten dimensional characters in a Douglas Adams novel. Third we can extend the result to structures of higher dimensions than lines, like fitting a plane in a four dimensional space, as we might need if we fit an affine flow to a set of image displacement data. And finally, the same technique can be applied to find the principal direction of any elongated object, even if we are not particularly interested in line fitting.

3.1. Alternative Way to Fit a Line

In a field as ancient as analytic geometry there are for sure many ways to represent a line and about as many ways to fit a line. If we are interested in lines in the 2-D plane only, the equation

$$ax + by + c = 0 \tag{3.4}$$

can represent a line. Parameters a, b and c are the parameters of the line and a point $P = [x, y]^T$ belongs to the line if and only if it satisfies Eq. (3.4). It is easy to see that if we are given two mpints $P_1 = [x_1, y_1]^T$ and $P_2 = [x_2, y_2]^T$ we can form two equations

$$ax_1 + by_1 + c = 0$$
$$ax_2 + by_2 + c = 0$$

and solve for a, b and c. Wait a minute! We have two equations and three unknowns. Before we jump into conclusions like “Eucleides was wrong”, it is worth noticing that the two equations are homogeneous and we can recover the triplet of unknowns up to a scale only. So we either set c to unity and solve for the other two, or, to avoid, in case the real value of c is zero, falling into a singularity, which all respected authors of the blogosphere assure us it is a black hole, we set the sum of the equares of the three unknowns to unity and solve three equations with three unknowns.

A far more interesting problem is to find the solution when we have many more points $P_i = [x_i, y_i]^T$, for $i = 1..N$ than the minimum two. The easiest and laziest thing to do is just sum up the squares of Eq. (3.4) applied to all points P_i

$$Q'(a, b, c) = \sum_{i=1}^{N} (ax_i + by_i + c)^2.$$

It is not at all obvious what the physical meaning of the squared quantities is, but this does not stop us. If we define

$$A = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

and

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

we can rewrite

$$\begin{aligned} Q'(a, b, c) &= \sum_{i=1}^{N} \left(A^T \mathbf{x}_i \right)^2 = \sum_{i=1}^{N} A^T \mathbf{x}_i \mathbf{x}_i^T A \\ & A^T \left(\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \right) A = A^T MA \end{aligned} \qquad (3.5)$$

where

$$M = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T .$$

We notice immediatelly two things. The first is that Eq. (3.5) is homogeneous and so the trivial solution ($A = 0$) minimizes Q. The second is that A, the vector of the parameters of the line is scale independent, that is, A and αA, for $\alpha \neq 0$ represent the same line. This means that we can avoid the trivial solution by using the Rayleigh quotient again. So we minimize

$$Q(a, b, c) = \frac{A^T MA}{A^T A} \qquad (3.6)$$

which we know it is minimized when A is parallel to the eigenvector with the smallest eigenvalue.

3.2. Alternative to the Alternative

The above derivation leaves a void in the the mind of the reader. The squared quantities, the sum of which we minimize, have no immediate physical meaning. The question is then whether the solution we get is the same as the one we get from formulations that minimize the sum of the squared distances of the points from the line. To see this we solve the problem once more by minimizing the squared distances, this time using Eq. (3.4) to represent the line.

In this representation the distance of a point $P_0 = [x_0, y_0]^T$ from the line is

$$D = (x - x_0)^2 + (y - y_0)^2 \tag{3.7}$$

where x and y are the coordinates of a point on the line, that is

$$ax + by + c = 0. \tag{3.8}$$

Furthermore this point is chosen such that the distance D is minimal. To find the distance D we can solve Eq. (3.8) for one of the two unknowns, say x, substitute it in Eq. (3.7) and then minimize the resulting expression to find

$$D = \frac{(ax_0 + by_0 + c)^2}{a^2 + b^2} \tag{3.9}$$

hence the quantity we will minimize is

$$Q(a, b, c) = \frac{\sum_{i=1}^{N}(ax_i + by_i + c)^2}{a^2 + b^2} = \frac{A^T MA}{a^2 + b^2} = \frac{A^T MA}{A^T \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} A}\,. \tag{3.10}$$

It is clear that the solution of Eq. (3.10) will differ from that of Eq. (3.6) in general. This is not an uncommon situation, where two approaches that have great appeal to intuition give two different solutions. And although the latter approach that minimizes the sum of squared distances appears to be more rigorous, it is not necessarily the “optimal” solution. Any claim to optimality depends on the underlying statistical model and unless we know this model we cannot assert that a solution is optimal. Under some rather common and very convenient assumptions (independent, identically distributed error with isotropic variance), the optimal solution is indeed provided by the minimization of the sum of squared differences, but one can easily imagine situations that this is not the case.

4. Overdetermined Linear Systems

Quite often we have to solve a system of linear equations where we have many more equations than we need but each one is of low quality. If we discard the extra equations and solve the linear system, then we might get low quality results. The solution is nothing less than least squares. How could it be. We are in the chapter about least squares.

As always we take all these equations, put everything in the left hand side, if it is not there already, square them and add them together. Minimizing this sum is a simple issue of differentiating with respect to the unknowns.

Let, then, A be a $M \times N$ matrix where $M > N$, b be a M-dimensional known vector and x an N-dimensional vector of unknowns. The quest is to find the best possible solution to

$$Ax = b$$

which we do by differentiating the squared norm of $Ax - b$, which is nothing more than the sum of the squares of the elements of $Ax - b$

$$\frac{\partial}{\partial x}(Ax - b)^2 = \frac{\partial}{\partial x}\left((Ax - b)^T(Ax - b)\right) =$$

$$2\left(\frac{\partial}{\partial x}(Ax - b)\right)^T(Ax - b) = 2A^T(Ax - b) = 2A^T A - A^T b$$

which we equate to zero and get

$$A^T Ax = A^T b$$

what statisticians call “normal equations”[‡].

Matrix $A^T A$ has a host of nice properties. It is symmetric, it is non-negative definite (and if invertible positive definite) and requires less storage than the original matrix A and more often than not we do not even need to compute matrix A as an intermediate result at all. If, at the set up stage of the problem the rows A_i of A and the corresponding elements b_i of b (e.g. the individual equations and the corresponding knowns) are produced successively then $A^T A$ and $A^T b$ can be computed by

$$A^T A = \sum_i A_i A_i^T$$

and

$$A^T b = \sum_i A_i b_i$$

both of which can easily be done incrementally.

The definiteness of $A^T A$ makes it easy to invert and any matrix inversion method performs better on this than other non-definite equations. And as if this was not enough, there are methods best suited for such normal equations, most notably *Conjugate Gradient*, *Cholesky Factorization*, *Singular Value Decomposition*, *Successive Overrelaxation* etc. The variety is stunning if not truly disheartening to anyone that has never heard any of these methods. But hold this pill. Rather few of these are needed to survive in Computer Vision.

[‡] since there is no mention in the literature of any abnormal equations, one can speculate that normal here means orthogonal.

5. Constrained Optimization

Quite often the solution we are seeking comes with strings attached. We do not just want to minimize some function, but we need to do so subject to certain constraints. Let's see a realistic example.

Assume we are tracking the projections of N points in a sequence of images and we want to compute the velocity of the points. We consult the visual motion literature and we decide to use a certain function $Q(u, v)$ that is minimized by the most probable u and v. But we also know that the scene is a rigid scene, which gives us a powerful constraint, since a rigid motion is a special kind of motion that gives rise to specific patterns of velocities for the projections of the points. So what we have is a minimization tempered by a constraint. These kinds of minimizations appear in many disciplines of Science and Engineering as well Political Science, Economics etc.

The simplest way to do the minimization is to use the constraint to solve for one or more of the unknowns, and eliminate it from the minimization. This is the method of choice when such elimination is possible. Unfortunately, it is not always.

5.1. Lagrange Multipliers

One of the most popular ways to do constraint minimization is Lagrange Multipliers. It works like magic, ones feels difficulty believing it when one sees it but it has been used on an extreme range of things, from flying in the air to sorting your socks. It is the Mary Poppins of methods.

Assume you want to minimize $Q(p)$ where p is a vector of dimension K subject to a constraint $c(q) = 0$ where $c(q)$ is a vector valued function of dimension M, $M < K$. It can be shown that this constrained minimization is equivalent to performing unconstrained minimization to the following expression

$$L(p, \lambda) = Q(p) + \lambda^T c(p) \tag{5.1}$$

where λ is a vector of dimension M. We now have $K + M$ unknowns, the vectors p and λ and an equal number of equations. We solve for the unknowns, throw away the vector λ and keep p. That's all? Yes, that's all.

Yet, as opposed to Mary Poppins, the method is not just perfect in every way. If we eliminated M unkowns by using the constraint $c(q) = 0$ we would have $K - M$ left. Now we have $K + M$. Moreover, as we shall see later, some of the nicer properties that we have been addicted to, are lost with Lagrange multipliers.

Well, magic is not really magic, it is mathematics or science. At least that's what some scientists say. Then how do the Lagrange Multipliers work? The exact proof is way beyond the scope of the text and the patience of the sane among its readers, but a little intuition can be helpful. We do this with a simple two dimensional example where we minimize a function of two variables x and y that is subject to an 1-D constraint on x and y sketched in Fig. 5.1. The constraint $c(x, y) = 0$ is represented by the almost straight line running from top left to about bottom right. The function S we want to minimize is depicted by a few of its level crossings at $S(x, y) = 8, 6, 4, 2$. These level crossings are usually closed curves. If the constraint was absent S would achieve its minimum somewhere in the inner oval, but since we are obliged to choose a solution that satisfies the constraint, we have to move up and down the constraint curve $c(x, y,) = 0$ until we find a minimum. The minimum on the constraint curve is where this curve touches a level crossing. In this example this level crossing is depicted by a dotted line. So the minimum is achieved right at the point of contact.

Now that we know that the minimum is achieved at such a point of contact all we have to do is ask a mathematician to translate this to equations and any competent mathematician will tell us that at such a point the two curves have gradients that are a scalar multiples of each other. If we name this scalar factor $-\lambda$, we will have little difficulty arriving at Eq. (5.1).

5.2. Distance of a Point from a Line

We saw a problem above that looked like a constrained minimization. We had to find the minimum of Eq. (3.7) subject to the condition of Eq. (3.8), the condition, in other words that the point belongs to the line. This is a simple problem and needs little help from a dead French math guru, but nevertheless we want to see if Mr. Lagrange was right. We form our expression to minimize

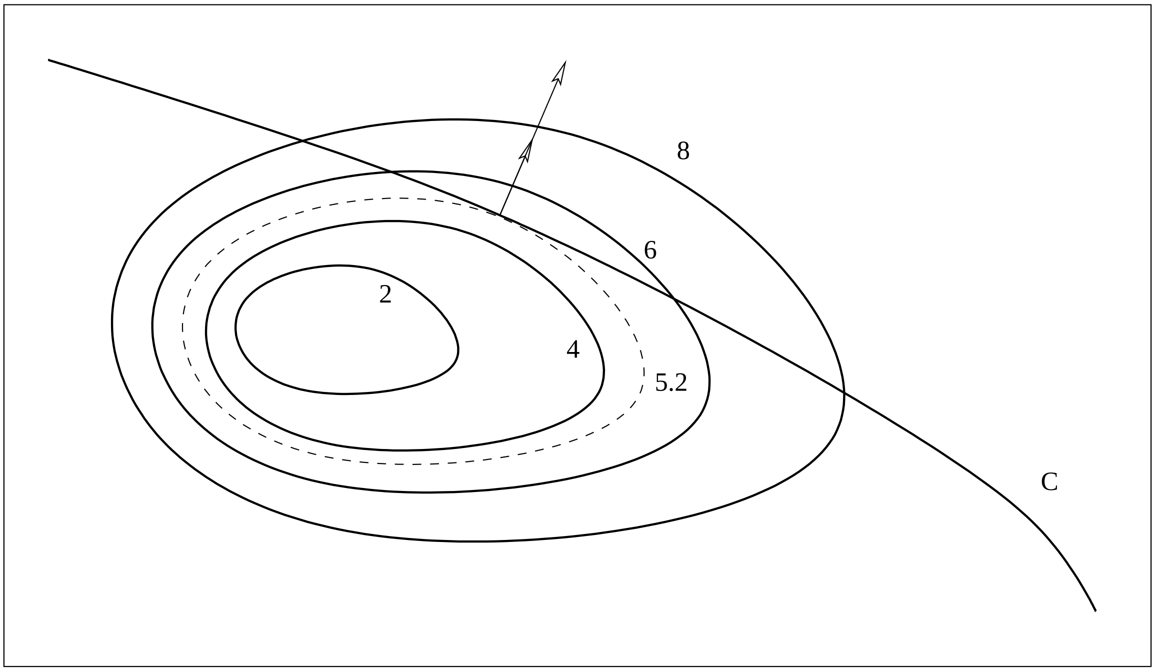

Figure 5.1: The minimum without the constraint should be somewhere in the middle of the innermost contour, but with the constraint the minimum is where the constraint line and the dotted line touch.

$$L(x, y, \lambda) = (x - x_0)^2 + (y - y_0)^2 + 2\lambda(ax + by + c)$$

and take derivatives with respect to the unknowns and λ. Notice a small deviation from the rules set out by Lagrange, namely we scaled the condition by 2λ instead of λ, to save a bit on typesetting. This has no other effect. So taking the derivatives with respect to x and y we get

$$\frac{\partial L}{\partial x} = 2(x - x_0 + \lambda a) = 0$$
$$\frac{\partial L}{\partial y} = 2(y - y_0 + \lambda b) = 0$$

from which we get

$$x = x_0 - \frac{1}{2}\lambda a$$
$$y = y_0 - \frac{1}{2}\lambda b.$$

If we take the derivative of L with respect to λ we get the constraint, and if we substitute x and y from above in it we can find that

$$\lambda = \frac{ax_0 + by_0 + c}{a^2 + b^2}$$

and from this we can easily derive that the minimum distance is

$$D = \frac{(ax_0 + by_0 + c)^2}{a^2 + b^2}$$

precisely as before. Thus far, Lagrange was right.

5.3. Application to Rayleigh Quotient

We can try the Lagrangian multipliers on something with known answer before we jump head first to something with unknown answer. What better than the Rayleigh Quotient we met a section ago.

$$R = \frac{q^T Cq}{q^2}$$

can be simplified a bit if we set $s = \frac{q}{|q|}$

$$R = s^T Cs \tag{5.2}$$

and we can now find s without nasty denominators and the funny differentiation rules for division. But from the above definition we have the constraint that s is a unit vector, or

$$1 - s^2 = 0. \tag{5.3}$$

Since we have a single constraint, λ is a scalar. The quantity to minimize is

$$L(s, \lambda) = s^T Cs + \lambda(1 - s^2). \tag{5.4}$$

Differentiation with respect to λ will give us the original constraint from Eq. (5.3) and differentiation with respect to the original unknowns q will give

$$\frac{\partial}{\partial s} L(s, \lambda) = 2Cs - 2\lambda s$$

which, if we equate to zero we get

$$Cs = \lambda s$$

or that s is an eigenvector of matrix C. Almost done. We know that the solution is an eigenvector but there are as many of them as dimensions in matrix C. So we replace s with eigenvector e_i in Eq. (5.2)

$$R = e_i^T Ce_i = \lambda_i e_i^T e_i = \lambda_i$$

and we see that it is equal to the corresponding eigenvalue. So if we want to minimize R then s is the eigenvector corresponding to the smallest eigenvalue. If we want to maximize R, to the largest eigenvalue.

5.4. Overdetermined System with Additional Constraints

Let's look at an overdetermined system of equations with additional constraints. We are given a system linear equations where the number of equations exceeds the number of unknowns, as is often the case in the presence of noise and uncertainty, and we want to solve it subject to a single (scalar) linear constraint. It should not be hard, and in a sense it is not. But we would like to avoid having to solve for some of the unknowns using the linear constraint and substituting them in the expression to be minimized.

Let, then as before, A be a $M \times N$ matrix where $M > N$, b be a M-dimensional known vector, x an N-dimensional vector of unknowns and c an N-dimensional known vector. We want to minimize

$$(Ax - b)^2$$

subject to the constraint

$$c^T x = 0$$

which we know is minimized for the same value of x as

$$L(x, \lambda) = (Ax - b)^2 + \lambda c^T x$$

where λ is an scalar unknown. If we differentiate $L(x, \lambda)$ we get

$$\frac{\partial}{\partial x} L(x, \lambda) = 2A^T(Ax - b) + \lambda c = 0$$

and

$$\frac{\partial}{\partial \lambda} L(x, \lambda) = c^T x = 0$$

which are linear equations and should be easy to solve. As we know any finite system of linear equations can be written in matrix form so we combine the two equations together to get

$$\begin{bmatrix} & . & \\ 2A^T A & . & c \\ & . & \\ . \;\; . \;\; . & . & . \;\; . \\ c^T & . & 0 \end{bmatrix} \begin{bmatrix} \\ x \\ \\ . \; . \\ \lambda \end{bmatrix} = \begin{bmatrix} \\ 2A^T b \\ \\ . \;\; . \;\; . \\ 0 \end{bmatrix} \qquad (5.5)$$

but unfortunately the matrix in the left hand side is not positive definite. This does not mean that the system is unsolvable, just means that it is much harder. The moral of the story: use Lagrange multipliers for analytic rather than numerical work, or do something about your addiction to positive definiteness.

Exercises

1. Find the extrema of the Rayleigh Quotient without using Lagrangian multipliers.

2. Let A be a symmetric matrix and $\dot{A}$ its time derivative. Find the time derivative of one of its eigenvalue and eigenvector pairs, say λ_0, e_0. No, no, no. That’ s too hard. Show that the derivatives are:

$$\dot{\lambda}_0 = e_0^T \dot{A} e_0$$

$$\dot{e}_0 = \sum_{i=1}^{N-1} \frac{e_i e_i^T}{\lambda_0 - \lambda_i} \dot{A} e_0$$

3. Let u_i be 2-D flow vectors measured at locations x_i. In a four dimensional space form the vector

$$v_i = \begin{bmatrix} u_i \\ \cdots \\ x_i \end{bmatrix}$$

and fit a plane through these points. Using this plane express flow as an affine function of x.

4. Let $u[i]$, $i = 1..N$, be a vector of unknowns and

$$u_g[i] = u(*)g[i] = \sum_{j=j_{\min}}^{j_{\max}} u[i-j]g[j]$$

be the convolution of u with convolution kernel or template g. Find the u that minimizes

$$\sum_i \sum_k \left(\alpha[i,k]u[i] + \beta[i,k]u_g[i] + c[i,k] \right)^2$$

5. Show how to solve Eq. (5.5) using blockwise matrix inversion (aka inversion by parts). Do you have to invert any non-positive matrix in the process?