We have spent a few lectures on how to process images and detect features such as edges. We will use these image measurements to estimate the parameters of several models. Today and the next lecture, we will introduce several estimation techniques: least squares, the Hough transform, and RANSAC.

Least squares: fitting points to a line

Suppose you have set of noisy points $\{(x_i, y_i)\}$ that roughly fall along a line in 2D plane (x, y). You would like to find the equation of that line. There are two ways to set up this problem:

1. Linear regression

If you have taken a statistics course, then you have seen the following version of the line fitting problem. (The problem is an example of *linear regression*.) You have two data variables x and y and you want to fit a linear relationship between samples (x_i, y_i) such that

$$y_i = mx_i + c. \tag{1}$$

(I am using the symbol c rather than b, because we will use $\mathbf{b}$ below to mean something else, and I don't want you to get confused.)

Typically this model does not fit the data exactly because of noise or other variability and, in particular, the variability is in y only, e.g. the x variable might be chosen by the experimentor (and is not random) and so only the y is random. In this case, one assumes a model

$$y_i = mx_i + c + n_i$$

where the n_i are noise. To fit the line, you find the m and c that minimizes $\sum_i n_i^2$, namely you find the m and c that minimizes

$$\sum_i (y_i - mx_i - c)^2 \tag{2}$$

To do so, you take the partial derivatives with respect to m and c and set them to 0. This gives you two linear equations with two unknowns m and c and coefficients that depends on the data x_i and y_i. You can easily solve these equations to get m and c.

We will see several versions of this linear regression problem in the course. It is typically easier to write the problem using matrices, so let's do that now. As you recall from your linear algebra course(s), it is common to write a linear system of equations

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{3}$$

where $\mathbf{A}$ is an $m \times n$ matrix of constants and $\mathbf{u}$ is an $n \times 1$ column vector of variables, and $\mathbf{b}$ is an $m \times 1$ column vector of constants). In the line fitting example, if we stack the various Eq. 1 for different values of i, then we can write them together as follows, which is in the form of Eq. 3.

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix}_{m\times 2} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}_{m\times 1}$$

Let's now restate the (general) linear regression problem:

Given an $m \times n$ matrix A with $m > n$ and a non-zero m-vector b, find the values of vector u that minimizes $\| \mathbf{Au} - \mathbf{b} \|_2$ where $\| \cdot \|_2$ is the L_2 norm.

Minimizing the L_2 norm is equivalent to minimizing the sum of squares of the elements of the vector $\mathbf{Au} - \mathbf{b}$, i.e. the L_2 norm is just the square root of the sum of squares of the elements of $\mathbf{Au} - \mathbf{b}$.

Let's give a general solution to this problem. We expand:

$$\| \mathbf{Au} - \mathbf{b} \|^2 \;=\; (\mathbf{Au} - \mathbf{b})^T(\mathbf{Au} - \mathbf{b}) \;=\; \mathbf{u}^T\mathbf{A}^T\mathbf{Au} - 2\mathbf{b}^T\mathbf{Au} + \mathbf{b}^T\mathbf{b}$$

To solve for $\mathbf{u} \in \Re^n$, take partial derivatives with respect to the n $\mathbf{u}$ variables and set them to 0. If you write out the above matrix and vector products as summations of the various elements, then taking these partial derivatives will give:

$$2\mathbf{A}^T\mathbf{Au} - 2\mathbf{A}^T\mathbf{b} = 0$$

or

$$\mathbf{A}^T\mathbf{Au} = \mathbf{A}^T\mathbf{b} \tag{4}$$

Eq. 4 are called the *normal equations*. One can show that if the columns of the $m \times n$ matrix $\mathbf{A}$ have full rank n (they are linearly independent) then $\mathbf{A}^T\mathbf{A}$ is invertible, then the solution is

$$\mathbf{u} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}.$$

2. Total least squares

Let's look at a slightly different version of the line fitting problem, which is known as *total least squares*. Again we have a set of points (x_i, y_i) in a 2D plane and we want to fit a line to these points. However, rather than taking our noise or "error" to be in the y direction only, we take the error to be the distance from (x_i, y_i) to a line

$$x\cos\theta + y\sin\theta = r \tag{5}$$

where θ is the direction of the *normal* to the line and r is the perpendicular distance from the origin to the line in direction θ. Note Eq. (5)is a more general equation for a line than Eq. 1 since Eq. (5) allows for lines of the form $x = constant$. BTW, if you've forgotten why this is the equation of a line, then rewrite it as

$$(x, y) \cdot (\cos\theta, \sin\theta) = r \tag{6}$$

and note that r is the component of (x, y) in the direction $(\cos\theta, \sin\theta)$ that is normal to the line (by definition). Hence, r is the shortest distance to the line, or the perpendicular distance.

The perpendicular distance from any point (x_i, y_i) in the plane to such a line is:

$$|x_i\cos\theta + y_i\sin\theta - r|.$$

We want to find the θ and r that minimize the sum of squared distances.

How do we solve for θ and r ? We cannot just use the same method as above, i.e. try to minimize

$$\sum_i (x_i\cos\theta + y_i\sin\theta - r)^2$$

directly, because you have the non-linearity of the cosine and sine. (As you vary θ, for any fixed r, the sum of squares expression is not a simple quadratic as before.) Instead, we find the values of a, b and r that minimize

$$\sum_i (x_i a + y_i b - r)^2 \tag{7}$$

subject to the (non-linear) constraint that $a^2 + b^2 = 1$, that is, (a, b) is a unit vector. This means we have a *constrained minimization* problem.

How to solve this problem? Taking the derivative of (7) with respect to r and set it to 0 gives:

$$a \sum_i x_i + b \sum_i y_i = r \sum_i 1.$$

Letting $\bar{x} = \frac{\sum_i x_i}{\sum_i 1}$ and $\bar{y} = \frac{\sum_i y_i}{\sum_i 1}$ be the sample means, we get

$$a\bar{x} + b\bar{y} = r$$

which says that the sample mean $(\bar{x}, \bar{y})$ falls on the solution line. The question then is which orientation does the solution line have?

Substituting r into (7) gives

$$\sum_i ((x_i - \bar{x})a + (y_i - \bar{y})b)^2.$$

Recall we are trying to minimize this expression, subject to $a^2 + b^2 = 1$. We can write this expression by defining an $m \times 2$ matrix

$$\mathbf{A} = \begin{bmatrix} x_1 - \bar{x}, y_1 - \bar{y} \\ \vdots \\ x_m - \bar{x}, y_m - \bar{y} \end{bmatrix}$$

where the matrix $\mathbf{A}$ is obviously different from the one above. The expression to be minimized then becomes

$$\begin{bmatrix} a & b \end{bmatrix} \mathbf{A}^T \mathbf{A} \begin{bmatrix} a \\ b \end{bmatrix}$$

where $a^2 + b^2 = 1$. The expression is minimized by the eigenvector of $\mathbf{A}^T\mathbf{A}$ having the smaller eigenvalue. (Note both eigenvalues are non-negative since $\mathbf{v}^T\mathbf{A}^T\mathbf{A}\mathbf{v} \geq 0$ for any $\mathbf{v}$.) Thus, our solution amounts to considering all the lines that pass through $(\bar{x}, \bar{y})$, and seeing which perpendicular direction θ minimizes the sum of square distances to the points (x_i, y_i)

Example problem: a "vanishing point"

Let's consider a different computer vision problem where we can apply least problem squares. We'll consider only a toy version of the problem for now. We'll return to the problem later in the course when we discuss 3D vision.

Many man-made 3D scenes contain large sets of parallel lines. These are typically aligned with the gravity axis or with a square floor pattern. For example, think of a hallway of an office building. The base and top of the wall where the wall meets the floor and ceiling, respectively, define parallel

lines. The vertical door frames also define parallel lines. A similar example is a street with buildings on the side. The street direction defines an orientation in which there are many parallel lines, and the lines of the building facades define both vertical lines as well as line parallel to the street.

Later in the course, we will discuss mathematical models of 3D geometry and how 3D scenes project into images. For now, I'll just appeal to your intuition. Whenever a set of parallel lines in the world project to an image, the lines in the image meet at a point which is called the *vanishing point*. One exception is that the lines in the world are parallel to the image plane. In that case, the lines do not meet in the image. (If we want to get fancy – and indeed we will later in the course – we can say that the lines meet in the image but at a 2d point at infinity.)

Consider an image that does contain a set of edges that belong to lines that pass through a vanishing point which is at pixel position (x_v, y_v). The vanishing point position does not actually have to be within the frame of the image. When I say that the image contains a set of edges, I mean that we have run an edge detector, and so we have a set of triples (x, y, θ) where (x, y) is the position of the edge and θ is the direction of the gradient of the image intensity, i.e. perpendicular to the edge.

For any hypothetical vanishing point (x_v, y_v), we need a way of saying whether an edge might be consistent with that vanishing point. For any edge that lies on a line that passes through the vanishing point (x_v, y_v), the following equation will hold:

$$(x - x_v, y - y_v) \cdot (\cos\theta, \sin\theta) = 0.$$

However, in practice the above equation will not hold exactly. There are two reasons. First, the edge estimate will have some noise, namely the underlying (continuous) edge won't pass exactly through pixel grid position (x, y) and also the estimation of θ might be off. Second, an edge might not correspond to this vanishing point – in other words, the edge might be an outlier.

We can write down a least squares version of the problem as follows: find the (x_v, y_v) that minimizes the following:

$$\sum_{edges(x,y,\theta))} | \ (x - x_v, y - y_v) \cdot (\cos\theta, \sin\theta) \ |^2$$

This formulation should work ok if there is very little noise and so the edge positions and orientations are accurate, and *if most the edges correspond to the vanishing point*. These conditions are unlikely to be met in real images, unfortunately, and in particular the assumption that all edges correspond to the vanishing point will fail in all but the most contrived cases. Next lecture we will discuss a few ways of dealing with this problem, namely the Hough transform and RANSAC. **You will implement these methods for this problem in Assignment 1.**

For now, we set up the least squares problem by writing the above summation as $\|\mathbf{A}\mathbf{u} - \mathbf{b}\|_2^2$, where

$$\mathbf{A} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ : & : \\ \cos\theta_m & \sin\theta_m \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} x_v \\ y_v \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} x_1\cos\theta_1 + y_1\sin\theta_1 \\ : \\ x_m\cos\theta_m + y_m\sin\theta_m \end{bmatrix}$$

We can then solve for $\mathbf{u}$ using the general method described above.