Least squares

We have seen several least squares problems thus far, and we will see more in the upcoming lectures. For this reason it is good to have a more general picture of these problems and how to solve them.

Version 1: Given an $m \times n$ matrix A, where $m > n$, find a unit length vector x that minimizes $\| \mathbf{Ax} \|$.

Here, and in the rest of this lecture, the norm $\| \cdot \|$ is the L_2 norm. Also note that minimizing the L_2 norm is equivalent to minimizing the sum squares of the elements of the vector $\mathbf{Ax}$, i.e. the L_2 norm is just the square root of the sum of squares of the elements of $\mathbf{Ax}$. Also, the reason we restrict the minimization to be for $\mathbf{x}$ of unit length is that, without this restriction, the minimum is achieved when $\mathbf{x} = \mathbf{0}$ which is uninteresting.

We can solve this constrained least squares problem using Lagrange multipliers, by finding the $\mathbf{x}$ that minimizes:

$$\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} + \lambda(\mathbf{x}^T\mathbf{x} - 1). \qquad (*)$$

That is, taking the derivative of (*) with respect to λ and setting it to 0 enforces that $\mathbf{x}$ has unit length. Taking derivatives with respect to the $\mathbf{x}$ components enforces

$$\mathbf{A}^T\mathbf{Ax} + \lambda\mathbf{x} = 0$$

which says that $\mathbf{x}$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$.

So which eigenvector gives the least value of $\mathbf{x}^T\mathbf{A}^T\mathbf{Ax}$? Since $\mathbf{x}^T\mathbf{A}^T\mathbf{Ax} \geq 0$, it follows that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are also greater than or equal to 0. Thus, to minimize $\| \mathbf{Ax} \|$, we take the (unit) eigenvector of $\mathbf{A}^T\mathbf{A}$ with the *smallest eigenvalue*, i.e. we are sure this eigenvalue is non-negative.

Version 2: Given an $m \times n$ matrix A with $m > n$, and given an m-vector b, minimize $\| \mathbf{Ax} - \mathbf{b} \|$.

If $\mathbf{b}$ is $\mathbf{0}$ then we have the same problem above, so let's assume $\mathbf{b} \neq \mathbf{0}$. We don't need Lagrange multipliers in this case, since the trivial solution $\mathbf{x} = 0$ is no longer a solution so we don't need to avoid it. Instead, we expand:

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

and then take partial derivatives with respect to the $\mathbf{x}$ variables and set them to 0. This gives

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

or

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b} \qquad (1)$$

which are called the *normal equations*. Assume the columns of the $m \times n$ matrix $\mathbf{A}$ have full rank n, i.e. they are linearly independent so $\mathbf{A}^T\mathbf{A}$ is invertible. (Not so obvious.) Then, the solution is

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

What is the geometric interpretation of Eq. (1)? Since $\mathbf{b}$ is an m-dimensional vector and $\mathbf{A}$ is an $m \times n$ matrix, we can *uniquely* write $\mathbf{b}$ as a sum of a vector in the column space of $\mathbf{A}$ and a vector in the space orthogonal to the column space of $\mathbf{A}$. To minimize $\| \mathbf{Ax} - \mathbf{b} \|$, by definition we find the $\mathbf{x}$ such that the distance from $\mathbf{Ax}$ to $\mathbf{b}$ is as small as possible. This is done by choosing $\mathbf{x}$ such that $\mathbf{Ax}$ is the component of $\mathbf{b}$ that lies in the column space of $\mathbf{A}$, that is, $\mathbf{Ax}$ is the orthogonal projection of $\mathbf{b}$ to the column space of $\mathbf{A}$. Note that if $\mathbf{b}$ already belonged in the column space of $\mathbf{A}$ then the "error" $\| \mathbf{Ax} - \mathbf{b} \|$ would be 0 and there would be an exact solution.

Pseudoinverse of A

In the above problem, we define the *pseudoinverse* of $\mathbf{A}$ to be the $n \times m$ matrix,

$$\mathbf{A}^+ \equiv (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T.$$

Recall we are assuming $m > n$, so that $\mathbf{A}$ maps from a lower dimensional space to a higher dimensional space. [1]

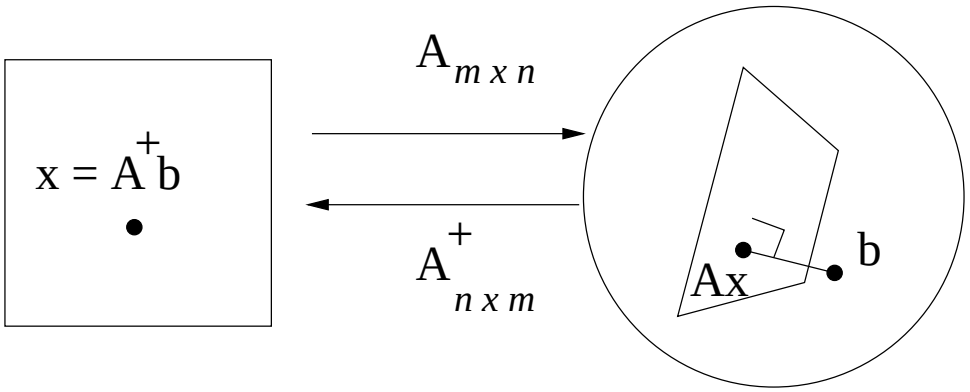
Note that

$$\mathbf{A}^+\mathbf{A} = I.$$

Moreover,

$$\mathbf{A}\mathbf{A}^+ = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$$

and $\mathbf{A}\mathbf{A}^+$ projects any vector $\mathbf{b} \in \Re^m$ onto the column space of $\mathbf{A}$, that is, it removes from $\mathbf{b}$ the component that is perpendicular to the column space of $\mathbf{A}$. This is illustrated in the figure below. Note that the plane in the figure on the right contains the origin, since it includes $\mathbf{A}\ \mathbf{x} = \mathbf{0}$, where $\mathbf{x} = \mathbf{0}$.



The pseudoinverse maps in the reverse direction of $\mathbf{A}$, namely it maps $\mathbf{b}$ in an m-D space to an $\mathbf{x}$ in an n-D space. Rather than inverting the mapping $\mathbf{A}$, however, only "inverts" the component of $\mathbf{b}$ that belongs to the column space of $\mathbf{A}$, i.e. $\mathbf{A}^+\mathbf{A} = \mathbf{I}$. It ignores the component of $\mathbf{b}$ that is orthogonal to the column space of $\mathbf{A}$.

[1] Note that if $m = n$ then $\mathbf{A}^+ = \mathbf{A}^{-1}$.

Non-linear least squares (Gauss-Newton method)

Let's look at a commmon application of version 2 above. Suppose we have m differentiable functions $f_i(\mathbf{x})$ that take $\Re^n$ to $\Re^m$. The problem we consider now is, given an initial value $\mathbf{x}_0 \in \Re^n$, find a nearby $\mathbf{x}$ that minimizes $\parallel \vec{f}(\mathbf{x}) \parallel$. Note that this problem is not well defined, in the sense that "nearby" is not well defined. Nonetheless, it is worth considering problems for which one can seek to improve the solution from some initial $\mathbf{x}_0$.

Consider a linear approximation of the m-vector of functions $\vec{f}(\mathbf{x})$ in the neighborhood of $\mathbf{x}_0$, and try to minimize

$$\parallel \vec{f}(\mathbf{x}_0) + \frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0) \parallel,$$

where the Jacobian $\frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}}$ is evaluated at $\mathbf{x}_0$. It is an $m \times n$ matrix. Notice that if you square this function then you get a quadratic that increases as any compoonent of $\mathbf{x}$ goes to infinity. So there is a unique minimum of this quadratic and that's what we want to find.

This linearized minimization problem is of the form we saw above in version 2 where $\mathbf{A} = \frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}}$ and $\mathbf{b} = \vec{f}(\mathbf{x}_0) - \frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{x}_0$. However, since we are using a linear approximation to the $\vec{f}(\mathbf{x})$ functions, we do not expect to minimize $\parallel \vec{f}(\mathbf{x}) \parallel$ exactly. Instead, we iterate

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \Delta\mathbf{x}.$$

At each step, we evaluate the Jacobian at the new point $\mathbf{x}^{(k)}$.

Examples

- To fit a line $y = mx + b$ to a set of points (x_i, y_i) we found the m and b that minimized $\sum_i (y_i - mx_i - b)^2$. This is a linear least squares problem (version 2).
- To fit a line $x\cos\theta + y\sin\theta = r$ to a set of points (x_i, y_i), we found the θ and r that minimized $\sum_i (ax_i + by_i - r)^2$, subject to the constraint $a^2 + b^2 = 1$. As we saw in lecture 15, this reduces to solving for the (a, b) that minimized $\sum_i (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2$, which is the version 1 least squares problem. Once we have θ, we can solve for r since (recall lecture 15), namely $r = \bar{x}\cos\theta + \bar{y}\sin\theta$.
- Vanishing point detection (lecture 15). It is of the form version 2.
- Recall the image registration problem where we wanted the (h_x, h_y) that minimizes:

$$\sum_{(x,y) \in Ngd(x_0,y_0)} \{I(x + h_x, y + h_y) - J(x, y)\}^2$$

  This is a non-linear least squares problem, since the (h_x, h_y) are parameters of $I(x+h_x, y+h_y)$. To solve the problem, we took a first order Taylor series expansion of $I(x + h_x, y + h_y)$ and thereby turned it into a linear least squares problem. Our initial estimate of (h_x, h_y) was $(0, 0)$. We then iterated to try to find a better solution.

  Note that we required that $\mathbf{A}$ is of rank 2, so that $\mathbf{A}^T\mathbf{A}$ is invertible. This condition says that the second moment matrix $\mathbf{M}$ needs to be invertible.

SVD (Singular Value Decomposition)

In the version 1 least squares problem, we need to find the eigenvector of $\mathbf{A}^T\mathbf{A}$ that had smallest eigenvalue. In the following, we use the eigenvectors and eigenvalues of $\mathbf{A}^T\mathbf{A}$ to decompose $\mathbf{A}$ into a product of simple matrices. This *singular value decomposition* is a very heavily used tool in data analysis in many fields. Strangely, it is not taught in most introductory linear algebra courses. For this reason, I give you the derivation.

Let $\mathbf{A}$ be an $m \times n$ matrix with $m \geq n$. The first step is to note that the $n \times n$ matrix $\mathbf{A}^T\mathbf{A}$ is symmetric and positive semi-definite. (It is obvious that $\mathbf{A}^T\mathbf{A}$ is symmetric, and it is easy to see that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are non-negative since $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} \geq 0$ for any real $\mathbf{x}$.) It follows from basic linear algebra $\mathbf{A}^T\mathbf{A}$ has an orthonormal set of eigenvectors.

[Why? The claim is that any symmetric real valued matrix $\mathbf{M}$ has an orthogonal (and hence orthonormal) set of eigenvectors. Here is why. First, if we have two eigenvectors $\mathbf{v_1}$ and $\mathbf{v_2}$ with distinct eigenvalues λ_1 and λ_2, then $\mathbf{v_1^T M v_2} = \mathbf{v_1^T M^T v_2}$ because $\mathbf{M}$ is symmetric, and $\lambda_1 \mathbf{v_1^T v_2} = \lambda_2 \mathbf{v_1^T v_2}$ and since we are assuming $\lambda_1 \neq \lambda_2$, it follows that $\mathbf{v_1^T v_2} = 0$. Second, if we have two eigenvectors $\mathbf{v_1}$ and $\mathbf{v_2}$ with the same eigenvalue, then we can perform Gram-Schmidt orthogonalization and to make the two eigenvectors orthogonal. Note I have not shown that if $\mathbf{M}$ is $n \times n$ then there must be n distinct eigenvectors. I SHOULD FILL THAT IN ONE DAY.]

Let $\mathbf{V}$ be an $n \times n$ matrix whose columns are the orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$. Since the eigenvalues are non-negative, we can write[2] them as σ_i^2, that is, σ_i is the square root of the *ith* eigenvalue. We can define $\sigma_i > 0$, i.e. it is the positive square root. Define $\mathbf{\Sigma}$ to be an $n \times n$ diagonal matrix with values $\Sigma_{ii} = \sigma_i$ on the diagonal. The elements, σ_i are called the *singular values* of $\mathbf{A}$. I emphasize: they are the square roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$. Also note that $\mathbf{\Sigma}^2$ is an $n \times n$ diagonal matrix, and

$$\mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2.$$

Next define $\tilde{\mathbf{U}}_{m\times n}$ as follows:

$$\tilde{\mathbf{U}}_{m\times n} \equiv \mathbf{A}_{m\times n}\mathbf{V}_{n\times n}. \tag{2}$$

Then

$$\tilde{\mathbf{U}}^T\tilde{\mathbf{U}} = \mathbf{V^T A^T A V} = \mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^2 = \mathbf{\Sigma}^2.$$

Thus, the n columns of $\tilde{\mathbf{U}}$ are orthogonal and of length σ_i.

We now normalize the columns of $\tilde{\mathbf{U}}$ by defining an $m \times n$ matrix $\mathbf{U}$ whose columns are ortho*normal* (length 1), so that

$$\mathbf{U}\mathbf{\Sigma} = \tilde{\mathbf{U}}.$$

Substituting into Eq. 2 and right multiplying by $\mathbf{V}^T$ gives us

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times n}\mathbf{\Sigma}_{n\times n}\mathbf{V}^T_{n\times n}.$$

Thus, *any* $m \times n$ matrix $\mathbf{A}$ can be decomposed into

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V^T}$$

where $\mathbf{U}$ is $m \times n$, $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix, and $\mathbf{V}$ is $n \times n$. A similar construction can be given when $m < n$.

[2]Forgive me for using the symbol σ yet again, but σ is *always* used in the SVD.

One often defines the *singular value decomposition* of $\mathbf{A}$ slightly differently than this, namely one defines the $\mathbf{U}$ to be $m \times m$, by just adding $m - n$ orthonormal columns. One also needs to add $m - n$ rows of 0's to Σ to make it $m \times n$, giving

$$\mathbf{A}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{\Sigma}_{m \times n} \mathbf{V}^T_{n \times n}$$

For our purposes there is no important difference between these two decompositions.

Finally, note Matlab has a function `svd` which computes the singular value decomposition. One can use `svd(A)` to compute the eigenvectors and eigenvalues of $\mathbf{A}^T\mathbf{A}$.