

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.

Version 1: Given an $m \times n$ matrix \mathbf{A} , find a unit length vector \mathbf{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 sum of squares of the elements of the vector \mathbf{Ax} , gives the same result as minimizing the square root of the sum of squares of the elements of \mathbf{Ax} .

Obviously the minimum is achieved when $\mathbf{x} = \mathbf{0}$ but this is uninteresting. So, we repose the problem by constraining the solution so it only considers vectors \mathbf{x} of some fixed length, e.g. 1. We can solve this constrained least squares problem using Lagrange multipliers, by finding the \mathbf{x} that minimizes:

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

Taking derivatives with respect to the \mathbf{x} components gives

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

which says that \mathbf{x} is an eigenvector of $\mathbf{A}^T \mathbf{A}$. Setting the derivative with respect to λ to 0 enforces that \mathbf{x} has unit length.

Which eigenvector gives the least value of $\mathbf{x}^T \mathbf{A}^T \mathbf{Ax}$? Clearly $\mathbf{A}^T \mathbf{A}$ is symmetric and so all eigenvalues are non-negative, and thus we want the eigenvector with the *smallest eigenvalue*.

Version 2: Given an $m \times n$ matrix \mathbf{A} and an m -vector \mathbf{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 $\mathbf{x} = \mathbf{0}$ is no longer a solution so we don't need to avoid it).

First we expand:

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

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

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

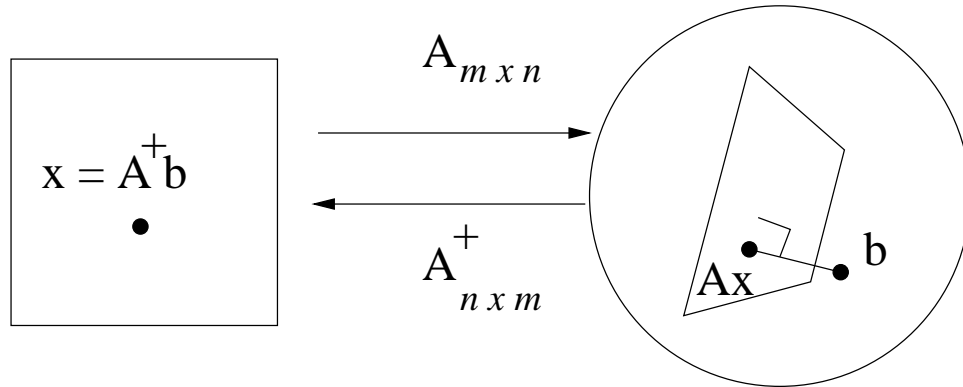
or

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}. \tag{1}$$

which are called the *normal equations*. We can solve for \mathbf{x} using simple matrix methods, as long as the columns of \mathbf{A} have rank n , i.e. they are linearly independent since, in that case, $\mathbf{A}^T \mathbf{A}$ is invertible. (You can verify that for yourself.)

What is the geometric interpretation of this solution? 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 $\|\mathbf{Ax} - \mathbf{b}\|$, we want to 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} . Equivalently, we want the vector $\mathbf{Ax} - \mathbf{b}$ to be entirely within the space that is orthogonal to the column space of \mathbf{A} , and so we require $\mathbf{A}^T(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}$ which is just Eq. (1)! Note that if \mathbf{b} already belonged in the column space of \mathbf{A} – that is, it could be represented as a linear combination of the columns of \mathbf{A} – then the least squares “error” would be 0 and there would be an exact solution.



Pseudoinverse of \mathbf{A}

For each \mathbf{b} , there is an \mathbf{x} that solves our minimization problem. If $\mathbf{A}^T \mathbf{A}$ is invertible - this happens if the columns of \mathbf{A} are linearly independent – our solution can be written

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

Define the *pseudoinverse* of \mathbf{A} to be the $n \times m$ matrix,

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

Note that if \mathbf{A} itself is invertible – in particular, it must be a square matrix – then $\mathbf{A}^+ = \mathbf{A}^{-1}$. Also,

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

projects any vector $\mathbf{b} \in \mathfrak{R}^m$ 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 was illustrated in the above figure, namely $\mathbf{A} \mathbf{A}^+$ projects \mathbf{b} onto the column space of \mathbf{A} .

The pseudoinverse maps in the reverse direction of \mathbf{A} , namely it maps \mathbf{b} in an m -D space to an n -D space and, rather than inverting, only “inverts” the component of \mathbf{b} that belongs to the column space of \mathbf{A} , i.e. $\mathbf{A}^+ \mathbf{A} = \mathbf{I}$ but $\mathbf{A} \mathbf{A}^+$ only equals \mathbf{I} when \mathbf{A} itself is invertible.

Non-linear least squares (Gauss-Newton method)

Suppose we have m differentiable functions $f_i(\mathbf{x})$ that take \mathfrak{R}^n to \mathfrak{R}^m . The problem we consider now is, given an initial value \mathbf{x}_0 , find a nearby \mathbf{x} that minimizes $\| \vec{f}(\mathbf{x}) \|$. 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 , so that we are trying to minimize

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

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.

This new linearized minimization problem is now of the form we saw above in version 2 where $\mathbf{A} = \frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}}$ and $\mathbf{b} = \vec{f}(\mathbf{x}_0) - \frac{\partial \vec{f}(\mathbf{x})}{\partial \mathbf{x}} \mathbf{x}_0$.

Since we are making a (first order) approximation, we do not expect to minimize $\| \vec{f}(\mathbf{x}) \|$ 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:

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

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 – that’s easy to see since $\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} \geq 0$ for any real \mathbf{x} . So $\mathbf{A}^T \mathbf{A}$ has an orthonormal set of eigenvectors and the eigenvalues are all real and non-negative. 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 write¹ them as σ_i^2 .

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} . Note that 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

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

Then

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

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

We now define an $m \times n$ matrix \mathbf{U} whose columns are *orthonormal* (length 1), so that

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

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

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

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 $m < n$.

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 $\mathbf{\Sigma}$ to make it $m \times n$, giving

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

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 $\mathbf{A}^T \mathbf{A}$.

¹Forgive me for using the symbol σ yet again, but σ is *always* used in the SVD.