6 Gram-Schmidt procedure, QR-factorization, Orthogonal projections, least square

Class notes with a few problems and their solutions are found

For Gram-Schmidt and QR by Eric Carlen

`http://www.math.gatech.edu/~carlen/1502/html/pdf/gram.pdf`
`http://www.math.gatech.edu/~carlen/1502/html/pdf/qr.pdf`

For Gram-Schmidt and QR by Michael Lacey

`http://www.math.gatech.edu/~lerdos/math2601/QR.pdf`

For orthogonal complements and projections by Eric Carlen

`http://www.math.gatech.edu/~carlen/1502/html/pdf/proj.pdf`
`http://www.math.gatech.edu/~carlen/1502/html/pdf/prj.pdf`

For orthogonal complements and projections by Michael Lacey
`http://www.math.gatech.edu/~lerdos/math2601/projection.pdf`

For Least squares by Eric Carlen

`http://www.math.gatech.edu/~carlen/1502/html/pdf/least.pdf`

In this review class, we will mostly follow Eric Carlen's presentation. For examples and more details, see the webpages above. Here we just state the most important results (some of them are not stated as explicitly in Carlen's notes as here).

6.1 Gram-Schmidt orthogonalization procedure

This is an algorithm to find an orthonormal basis (ONB, for brevity) $\{\mathbf{u}_1, \mathbf{u}_2, \dots \mathbf{u}_\ell\}$ in the span of a given set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_k\}$. The algorithm is successive; first it finds an ONB in $\mathcal{V}_1 = \text{Span}\{\mathbf{v}_1\}$, then in $\mathcal{V}_2 = \text{Span}\{\mathbf{v}_1, \mathbf{v}_2\}$, then in $\mathcal{V}_3 = \text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ etc.

It could happen that some vector $\mathbf{v}_j$ is already in $\mathcal{V}_{j-1}$, i.e. in the span of $\{\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_{j-1}\}$ for some j. In this case, we do not have to add any new vector to the ONB at that stage. For this reason, there could be a "jump" in the indices of the ONB. Suppose that $j_1 < j_2 < j_3 < \dots < j_\ell$ are those indices, where the sequence of $\mathcal{V}_j$-subspaces truly increases, i.e.

$$\begin{aligned} \{\mathbf{0}\} &= \mathcal{V}_1 = \mathcal{V}_2 = \dots = \mathcal{V}_{j_1-1} \subset \\ &\subset \mathcal{V}_{j_1} = \mathcal{V}_{j_1+1} = \dots = \mathcal{V}_{j_2-1} \subset \\ &\subset \mathcal{V}_{j_2} = \mathcal{V}_{j_2+1} = \dots = \mathcal{V}_{j_3-1} \subset \\ &\subset \dots \\ &\subset \mathcal{V}_{j_\ell} = \mathcal{V}_{j_\ell+1} = \dots = \mathcal{V}_k \end{aligned}$$

(with the convention that $\mathcal{V}_0 = \{\mathbf{0}\}$, which is needed if $j_1 = 1$). Of course in most cases $j_1 = 1$, $j_2 = 2$, $j_3 = 3$ etc. But if there is a "jump", say $j_1 = 1$, $j_2 = 2$ but $j_3 = 4$, then it means that $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ span the same space as $\{\mathbf{v}_1, \mathbf{v}_2\}$, i.e. $\mathbf{v}_3$ is not linearly independent from the previous vectors. In this case $\mathcal{V}_3$ is also spanned by only two orthonormal vectors, $\mathbf{u}_1, \mathbf{u}_2$.

The algorithm gives orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2, \dots$ such that

$$\text{span}\{\mathbf{u}_1\} = \mathcal{V}_{j_1}$$

$$\text{span}\{\mathbf{u}_1, \mathbf{u}_2\} = \mathcal{V}_{j_2}$$

etc.

$$\text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots \mathbf{u}_\ell\} = \mathcal{V}_{j_\ell}$$

The algorithm will give the $j_1, j_2, \dots j_\ell$ indices as well. Notice that the number of $\mathbf{v}$ vectors may not be the same as the number of $\mathbf{u}$ vectors, in fact the latter, ℓ, is the dimension of the span of $\{\mathbf{v}_1, \dots \mathbf{v}_k\}$.

In the description below the main text refers to the standard case $j_m = m$ (when the $\{\mathbf{v}_1, \dots \mathbf{v}_k\}$ set is linearly independent) and in parenthesis we remark the general case.

Step (1.) Normalize the first nonzero vector, i.e. define

$$\mathbf{u}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$$

Clearly $\text{Span}\{\mathbf{u}_1\} = \text{Span}\{\mathbf{v}_1\}$.

[*General case:* Above we tacitly supposed that $\mathbf{v}_1 \neq \mathbf{0}$, i.e. $j_1 = 1$. If $\mathbf{v}_1 = \mathbf{0}$ then j_1 will be the index of the first nonzero vector in the sequence $\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_k$, and $\mathbf{u}_1$ will be $\frac{\mathbf{v}_{j_1}}{\|\mathbf{v}_{j_1}\|}$,]

Step (2.) Compute the projection of $\mathbf{v}_2$ onto $\mathbf{u}_1$

$$\mathbf{w}_2 := \mathbf{v}_2 - (\mathbf{v}_2^t \cdot \mathbf{u}_1)\mathbf{u}_1$$

If $\mathbf{w}_2 \neq \mathbf{0}$, then define

$$\mathbf{u}_2 = \frac{\mathbf{w}_2}{\|\mathbf{w}_2\|}$$

as our second orthogonal basis vector. Clearly $\mathcal{V}_2 = \text{Span}\{\mathbf{v}_1, \mathbf{v}_2\} = \text{Span}\{\mathbf{u}_1, \mathbf{u}_2\}$. If $\mathbf{w}_2 = \mathbf{0}$, then notice that $\mathbf{v}_2$ is paralell with $\mathbf{v}_1$, i.e. no need for one more basis vector to generate the span of $\{\mathbf{v}_1, \mathbf{v}_2\}$. In this case we do not add a new basis vector.

[*General case:* Again, the paragraph above referred to the typical case. But if $\mathbf{u}_1 = \frac{\mathbf{v}_{j_1}}{\|\mathbf{v}_{j_1}\|}$ in the previous step, then

$$\mathbf{w}_{j_1+1} := \mathbf{v}_{j_1+1} - (\mathbf{v}_{j_1+1}^t \cdot \mathbf{u}_1)\mathbf{u}_1$$

i.e you always pick the next untouched vector from the sequence $\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_k$. Moreover, it could happen that $\mathbf{w}_{j_1+1} = \mathbf{0}$, in which case you have to go on to the next vector

$$\mathbf{w}_{j_1+2} := \mathbf{v}_{j_1+2} - (\mathbf{v}_{j_1+2}^t \cdot \mathbf{u}_1)\mathbf{u}_1$$

and continue until you find the first a nonzero vector, say $\mathbf{w}_{j_1+m}$. Then $j_2 = j_1 + m$ and $\mathbf{u}_2$ will be $\mathbf{w}_{j_1+m} = \mathbf{w}_{j_2}$ normalized.]

Step (i.) Suppose that so far we have found orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_i$ such that

$$\text{Span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_i\} = \text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_i\} = \mathcal{V}_i$$

Consider the vector

$$\mathbf{w}_{i+1} = \mathbf{v}_{i+1} - (\mathbf{v}_{i+1}^t \cdot \mathbf{u}_1)\mathbf{u}_1 - (\mathbf{v}_{i+1}^t \cdot \mathbf{u}_2)\mathbf{u}_2 - \ldots - (\mathbf{v}_{i+1}^t \cdot \mathbf{u}_i)\mathbf{u}_i$$

i.e. the projection of $\mathbf{v}_{i+1}$ onto the subspace spanned by $\{\mathbf{u}_1, \ldots \mathbf{u}_i\}$ (at this stage you do not have to "know" that it is a projection, since we have not defined it yet, but it is good to have an idea what's going on). If this vector is nonzero, $\mathbf{w}_{i+1} \neq \mathbf{0}$, then let

$$\mathbf{u}_{i+1} = \frac{\mathbf{w}_{i+1}}{\|\mathbf{w}_{i+1}\|}$$

If it is zero, then we do not create a new $\mathbf{u}$ vector and we go on to the next untouched $\mathbf{v}$ vector.

[*General case:* The algorithm in the general case goes as follows. Suppose so far we have found orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_i$ such that

$$\text{Span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_i\} = \text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_{j_i}\} = \mathcal{V}_i$$

Consider the vectors

$$\mathbf{w}_m = \mathbf{v}_m - (\mathbf{v}_m^t \cdot \mathbf{u}_1)\mathbf{u}_1 - (\mathbf{v}_m^t \cdot \mathbf{u}_2)\mathbf{u}_2 - \ldots - (\mathbf{v}_m^t \cdot \mathbf{u}_i)\mathbf{u}_i$$

for $m = j_i + 1, j_i + 2, \ldots$. Let j_{i+1} be the index of the first nonzero among these vectors. This means that $\mathbf{v}_{j_{i+1}}$ is the first vector *not* in the span of $\{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_i\}$. Then we define

$$\mathbf{u}_{i+1} := \frac{\mathbf{v}_{j_{i+1}}}{\|\mathbf{v}_{j_{i+1}}\|}$$

and this will be the new basis vector.]

For more details and examples, see

`http://www.math.gatech.edu/~carlen/1502/html/pdf/gram.pdf`

6.2 Orthogonal projections

Suppose that $\{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_k\}$ is an orthonormal basis in $\mathcal{V}$ which is a subspace of $\mathbf{R}^n$. Consider the $n \times k$ matrix Q formed from these vectors as its columns and define

$$P = QQ^t$$

Clearly

$$P\mathbf{w} = \sum_{i=1}^{k} (\mathbf{u}_i^t \cdot \mathbf{w})\mathbf{u}_i$$

and notice that $Q^tQ = I_k$, the $k \times k$ identity matrix. The following is the key theorem

Theorem 6.1 *The matrix $P = QQ^t$ defined above is independent of the orthogonal basis chosen, it depends only on the subspace $\mathcal{V}$. It has the properties*

$$P^2 = P, \qquad P^t = P$$

and for any $\mathbf{w} \in \mathbf{R}^n$

$$\|\mathbf{w} - P\mathbf{w}\| < \|\mathbf{w} - \mathbf{v}\|$$

for any $\mathbf{v} \in \mathcal{V}$ other than $P\mathbf{w}$. In other words, $P\mathbf{w}$ is the vector that is closest to $\mathbf{w}$ lying in $\mathcal{V}$.

Given a subspace $\mathcal{V}$ of $\mathbf{R}^n$ we can consider all vectors orthogonal to that, i.e. we define the set

$$\mathcal{V}^\perp := \{\mathbf{w} \ : \ \mathbf{w}^t \cdot \mathbf{v} = 0 \text{ for all } \mathbf{v} \in \mathcal{V}\}$$

It is easy to see the following properties

Theorem 6.2 *(i.) The set $\mathcal{V}^\perp$ is a linear subspace;*

(ii.) $\mathcal{V} \cap \mathcal{V}^\perp = \{\mathbf{0}\}$;

(iii.) $\mathcal{V}$ and $\mathcal{V}^\perp$ span $\mathbf{R}^n$, moreover, every vector $\mathbf{w} \in \mathbf{R}^n$ can be **uniquely** *written as*

$$\mathbf{w} = \mathbf{v} + \mathbf{v}^\perp \tag{6.1}$$

with $\mathbf{v} \in \mathcal{V}$, $\mathbf{v}^\perp \in \mathcal{V}^\perp$.

(iv.) If P is the orthogonal projection onto $\mathcal{V}$, and $P^\perp$ is the orthogonal projection onto $\mathcal{V}$, then $P + P^\perp = I$ (identity). The range and the kernel of P are given as

$$R(P) = \mathcal{V} \qquad \textit{and} \qquad \textit{Ker}(P) = \mathcal{V}^\perp$$

Moreover $\mathbf{v} = P\mathbf{w}$ and $\mathbf{v}^\perp = P^\perp\mathbf{w} = (I - P)\mathbf{w} = \mathbf{w} - P\mathbf{w}$ in the decomposition (6.1).

(v.) We have the relation

$$(\mathcal{V}^\perp)^\perp = \mathcal{V}.$$

(vi.) An orthonormal basis of $\mathcal{V}^\perp$ can be obtained from an orthonormal basis $\{\mathbf{u}_1, \ldots \mathbf{u}_k\}$ of $\mathcal{V}$ by either applying the Gram-Schmidt procedure to the columns of $I - P = I - QQ^t$, or to the spanning set

$$\{\mathbf{u}_1, \mathbf{u}_2 \ldots \mathbf{u}_k, \mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_n\}$$

and keeping only the last $n - k$ elements of the resulting basis of $\mathbf{R}^n$.

The subspaces $\mathcal{V}$ and $\mathcal{V}^\perp$ are called **orthogonal complements** of each other (see relation (v.)). One of the most important example is expressed in Lemma 4.13, i.e. for any matrix A

$$N(A)^\perp = R(A^t) \qquad \text{or} \qquad N(A^t)^\perp = R(A)\ . \tag{6.2}$$

Finally, recall that any subspace $\mathcal{V}$ can be described in two different ways: *in parametric form* or *with constraints*. Either you give a basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ in $\mathcal{V}$ (and orthonormal basis

are usually even better), and use that any element of $\mathbf{v}$ can be uniquely written as

$$\mathbf{v} = \sum_{j=1}^{k} a_j \mathbf{u}_j$$

and the numbers (coordinates) a_j are the parameters specifying $\mathbf{v}$.

Or, you specify $\mathbf{v}$ belonging to $\mathcal{V}$ by the property that $\mathbf{v} \in \mathcal{V}$ if and only if

$$P^{\perp}\mathbf{v} = 0$$

i.e.

$$\mathbf{v}^t \cdot \mathbf{u}_i = 0 \qquad i = j+1, j+2, \ldots n,$$

where $\{\mathbf{u}_{j+1}, \mathbf{u}_{j+2}, \ldots \mathbf{u}_n\}$ is an ONB in $\mathcal{V}^{\perp}$. These are the constraint equations.

The parametric form is useful if you want to **list** all elements of the subspace (e.g. have to present lots of elements of the subspace for some test). The constraint form is useful for **membership** problems, if you have to decide whether a given element is in the subspace or not.

Recall Problems 4.5 and 4.6 about the description of $R(A)$. Problem 4.6 asked to give a basis in $R(A)$, i.e. it asked for a parametric representation (though at that time orthogonality was not required). Problem 4.5 asked for constraints for $R(A)$, i.e. relations among the coordinates of a vector $\mathbf{b}$ to ensure that $\mathbf{b} \in R(A)$. This is the same, by (6.2), as finding basis vectors in $N(A^t)$ and requiring that $\mathbf{b}$ be orthogonal to all of them. In the solution to Problem 4.5 we have found that $-b_2 - b_3 + b_4 = 0$. In other words, the vector

$$\mathbf{u} = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

spans $N(A^t)$ the nullspace of the transpose matrix (CHECK!). In this case $N(A^t)$ is one dimensional, but you could have ended up with more than one fully zero rows after the

elimination with a general right hand side $\mathbf{b}$. In this case you have as many constraints as fully zero rows and this is also the dimension of $N(A^t)$. Also these constraints, written as $\mathbf{u}_i^t \cdot \mathbf{b} = 0$, give immediately an orthogonal basis in $N(A^t)$ (which you can normalize if you wish).

For more details and examples, see Eric Carlen's notes:
`http://www.math.gatech.edu/~carlen/1502/html/pdf/proj.pdf`
and with Maple solutions to the problems:
`http://www.math.gatech.edu/~carlen/1502/html/pdf/prj.pdf`

6.3 QR decomposition

Once you understood the Gram-Schmidt procedure, then the QR decomposition is easy. The key point is that the QR-decomposition runs a Gram-Schmidt algorithm for the column vectors of a matrix A starting from the leftmost vector. It ignores those columns which are linearly dependent of the previous ones (hence, in the pivoting language, it picks only the pivot columns). The columns of the matrix Q is therefore the Gram-Schmidt output of the pivot column vectors. These vectors can be expressed by the orthogonalized columns in an upper triangular form:

$$\begin{aligned} \mathbf{a}_1 &= r_{11}\mathbf{q}_1 \\ \mathbf{a}_2 &= r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2 \\ \mathbf{a}_3 &= r_{13}\mathbf{q}_1 + r_{23}\mathbf{q}_2 + r_{33}\mathbf{q}_3 \\ &\vdots \end{aligned}$$

where $\mathbf{a}_1, \mathbf{a}_2, \ldots$ are the pivot columns of A. (Note: here $\mathbf{a}_i$, the columns of A are the vectors to be orthonormalized, i.e. these play the role of the vectors $\mathbf{v}_i$ in Section 6.1. The vectors $\mathbf{q}_i$ are the resulting orthonormal vectors, these play the role of the $\mathbf{u}_i$ vectors in Section 6.1.)

If all columns are pivot columns, then one immediately has $A = QR$ with

$$A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_k \end{bmatrix}$$

$$Q = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_k \end{bmatrix}$$

and

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots \\ 0 & r_{22} & r_{23} & \dots \\ 0 & 0 & r_{33} & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

If A has nonpivot columns, then R contains columns which express these nonpivot columns in terms of those columns in Q which were obtained from preceding pivot columns of A. In general, the decomposition looks like

$$A = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{pmatrix} = \begin{pmatrix} * & * \\ * & * \\ * & * \end{pmatrix} \begin{pmatrix} \underline{*} & * & * & * & * \\ 0 & 0 & \underline{*} & * & * \end{pmatrix} = QR$$

(where $*$ denotes elements not necessarily zero). The first nonzero (pivot) elements are underlined in each row. These determine the location of the linearly independent columns in A. In this example the column space of A is two dimensional. The two columns of Q is an orthonormal basis in $R(A)$. These two columns are obtained by applying Gram-Schmidt to the first and third column of A. The coefficients in this Gram-Schmidt procedure are in the first and third columns of R. Finally, the coefficients in the second, fourth and fifth column of R express the remaining (not linearly independent) column vectors of A as linear combinations of the columns of Q. One can easily express these coefficients (i.e. the matrix elements of R) as

$$r_{ij} = \mathbf{q}_i^t \cdot \mathbf{a}_j$$

These properties are summarized

Theorem 6.3 *Let A be an $n \times k$ matrix and let $r = rank(A)$. Then there exist a matrix Q of dimensions $n \times r$ consisting of orthonormal columns and an upper triangular matrix R of dimension $r \times k$ such that the following properties hold*

- $A = QR$
- $R(A) = R(Q)$. *In particular, the columns of* Q *form an orthonormal basis in* $R(A)$, *hence* $Q^tQ = I$. *The matrix* $P = QQ^t$ *is the orthogonal projection from* $\mathbf{R}^n$ *onto* $R(A)$.
- *Ker* $A =$ *Ker* R, *in particular rank*$(R) = r$.

Finally, it is always possible to ensure that the first nonzero elements in each row of R *be nonnegative. With this extra requirement, such a decomposition is unique.*

6.4 Least squares

The method of least squares aims at finding a vector $\mathbf{x}$ for any given $n \times k$ matrix A and n-vector $\mathbf{b}$ such that $\|A\mathbf{x} - \mathbf{b}\|^2$ is the smallest possible. Why is this interesting? Of course if $\mathbf{b} \in R(A)$, then just choose $\mathbf{x}$ to be (one of) the solution to $A\mathbf{x} = \mathbf{b}$ and this reaches the smallest possible value (namely zero). What if $\mathbf{b} \notin R(A)$? This especially often happens if you have many equations and a few unknowns ($n \gg k$). For example this is typical problem with curve fitting.

In this case you cannot solve $A\mathbf{x} = \mathbf{b}$ exactly, but you can aim for a solution $\mathbf{x}$ such that $A\mathbf{x}$ be as close as possible to $\mathbf{b}$. This is given by the QR factorization as well:

Theorem 6.4 *(Least squares for overdetermined systems) Let* $A = QR$ *be the QR-factorization of the matrix* A. *Then the equation*

$$R\mathbf{x} = Q^t\mathbf{b}$$

has a solution for any $\mathbf{b} \in \mathbf{R}^n$. *Every such solution minimizes the expression* $\|A\mathbf{x} - \mathbf{b}\|^2$, *i.e.*

$$\|A\mathbf{x} - \mathbf{b}\|^2 \leq \|A\mathbf{y} - \mathbf{b}\|^2$$

for any vector $\mathbf{y} \in \mathbf{R}^k$.

For the proof, in nutshell, recall that QQ^t is the orthogonal projection onto $R(A)$ in $\mathbf{R}^n$. Hence the point closest to $\mathbf{b}$ in $R(A)$ is $QQ^t\mathbf{b}$, i.e. we have to solve $A\mathbf{x} = QQ^t\mathbf{b}$ instead of $A\mathbf{x} = \mathbf{b}$. Since $A = QR$, we have $QR\mathbf{x} = QQ^t\mathbf{b}$. Multiplying it by Q^t from the left and using $Q^tQ = I$ we get exactly $R\mathbf{x} = Q^t\mathbf{b}$.

Finally we show that $R\mathbf{x} = \mathbf{v}$ has a solution for any $\mathbf{v} \in \mathbf{R}^r$ (in particular for $\mathbf{v} = Q^t\mathbf{v}$). But this is clear from $rank(R) = r$ hence the column space of R is $\mathbf{R}^r$.

There is a "least square" method for underdetermined systems as well, but it is less frequently used. It selects the smallest possible solution to $A\mathbf{x} = \mathbf{b}$, assuming that there is a solution at all.

Theorem 6.5 *(Minimal solution for the underdetermined case) For any* $\mathbf{b} \in \mathbf{R}^n$*, if* $A\mathbf{x} = \mathbf{b}$*, then* $A(P_r\mathbf{x}) = \mathbf{b}$ *as well, where* P_r *is the orthogonal projection onto the row space of* A*. In other words,* $\mathbf{x}^* = P_r\mathbf{x}$ *is also a solution and in fact*

$$\|\mathbf{x}^*\|^2 < \|\mathbf{x}\|$$

for any other solution $\mathbf{x}$ *of* $A\mathbf{x} = \mathbf{b}$*.*

For the proof, just recall that

$$A\mathbf{x} = AP_r\mathbf{x} + A(I - P_r)\mathbf{x}$$

but $I - P_r$ is the projection onto the orthogonal complement of the row space of A, hence $A(I - P_r) = 0$. This gives $A\mathbf{x} = A\mathbf{x}^*$ if $\mathbf{x}^* = P_r\mathbf{x}$. The minimality of $\|\mathbf{x}^*\|$ follows from the basic property of the orthogonal projections.

Recall that for getting P_r, you have to find the QR-factorization of A^t: $A^t = QR$ and $P_r = QQ^t$.

The least square method has numerous applications, some of the are found on the web-page mentioned above

`http://www.math.gatech.edu/~carlen/1502/html/pdf/least.pdf`

Here is another application which goes back to Gauss, the inventor of the method. (This part is taken from "Applied numerical linear algebra" by J. W. Demmel)

6.4.1 Application of the Least Squares in geodesy

How do you measure the distance on a real landscape? How do you measure the height of the mountains? In other words, how to you figure out the coordinates of a given geographic point with respect to the standard coordinate system of the Earth (latitude, longitude, height with respect to the sea level)? The modern General Positioning System (GPS) uses satellites, but let us go back to times where everything had to be done on the Earth...

The way to do it was to put reference points (so called landmarks) all over on the terrain at well visible points (when you hike, you can still see them on the top of big mountains). The US geodetic database consisted of about 700,000 landmarks (1974) more or less uniformly spaced points at visible distances (a few miles) from each other. The goal is to find the coordinates of these points very accurately. The number of unknowns is about 2,100,000. In fact Gauss in nineteenth century has been asked to solve a similar problem in Germany (of course with much less numbers). And he invented the method for this purpose...

The quantity which can be measured very accurately is the angle. At any landmark P they measured the angle between the lines PQ_i and PQ_j for a few nearby (visible) landmarks Q_i, Q_j. In this way one obtained a few numbers for each landmark, for definiteness, let's say, ten angles for each P. Hence altogether they obtained $7,000,000$ numbers. Of course these numbers are not independent of each other; from elementary geometry we know lots of relations between them (most notably, the sum of the angles of a triangle...). In any case, using the cosine theorem, one has $7,000,000$ equations between the unknown coordinates and the measured angles. These equations are actually not exactly linear, but one can linearize

them in a consistent way. Hence the problem is to solve a huge linear system $A\mathbf{x} = \mathbf{b}$ of $7,000,000$ equations with 2,100,000 unknowns. If everything were measured perfectly, there would be an exact solution to this overdetermined system. But the measured data are not exact, and since we have much more equations and unknowns, the errors most likely drive $\mathbf{b}$ out of the column space of A. But one can search for the least square solution.

In 1978 such a system was solved for updating the US geodetic database with about 2,5 million equations and 400,000 unknowns, which was the biggest Least Square problem ever solved at the time (some elementary geometric relations allowed to reduce the number of equations and unknowns a bit). The actual computation heavily used further special structure of the matrix A, namely that it is a very *sparse matrix* (most elements are zero). This is because all relations expressed by the measured angles are actually relations between neighboring landmarks. Each equation involves only a few out of the 400,000 variables. This is very typical in many applications, and the numerical algorithms for sparse matrices is a well developed separate branch of mathematics.