LEAST SQUARES APPROXIMATION

SYNOPSIS. Having established the key theoretical results, the main linear algebra problems can now be solved. The typical scenario within data science applications is that a large dimensional vector representation of some object is available and greater insight is sought by seeking a description in terms of linear combination of a small number of vectors. The large dimensional object might not be exactly recovered and the focus is on obtaining the best possible approximation. In Euclidean spaces with distances measured by the 2-norm, best approximants are readily found by the generalization of the Pythagorean theorem to high-dimensional spaces.

1. Orthogonal projection

Consider a partition of a vector space U into orthogonal subspaces $U = V \oplus W$, $V = W^{\perp}$, $W = V^{\perp}$, typically $U = \mathbb{R}^m$, $V \subset \mathbb{R}^m$, $W \subset \mathbb{R}^m$, $\dim V = n$, $\dim W = m - n$. If $\boldsymbol{V} = [\, \boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_n \,] \in \mathbb{R}^{m\times n}$ is a basis for V and $\boldsymbol{W} = [\, \boldsymbol{w}_1 \ \ldots \ \boldsymbol{w}_{m-n} \,] \in \mathbb{R}^{m\times(m-n)}$ is a basis for W, then $\boldsymbol{U} = [\, \boldsymbol{v}_1 \ \ldots \ \boldsymbol{v}_n \ \boldsymbol{w}_1 \ \ldots \ \boldsymbol{w}_{m-n} \,]$ is a basis for U. Even though the matrices $\boldsymbol{V}, \boldsymbol{W}$ are not necessarily square, they are said to be orthonormal when all columns are of unit norm and orthogonal to one another. In this case computation of the matrix product $\boldsymbol{V}^T\boldsymbol{V}$ leads to the formation of the identity matrix within $\mathbb{R}^n$

$$\boldsymbol{V}^T\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots & \boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1^T\boldsymbol{v}_1 & \boldsymbol{v}_1^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_1^T\boldsymbol{v}_n \\ \boldsymbol{v}_2^T\boldsymbol{v}_1 & \boldsymbol{v}_2^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_2^T\boldsymbol{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{v}_n^T\boldsymbol{v}_1 & \boldsymbol{v}_n^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_n^T\boldsymbol{v}_n \end{bmatrix} = \boldsymbol{I}_n.$$

Similarly, $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_{m-n}$. Whereas for the square orthogonal matrix $\boldsymbol{U}$ multiplication both on the left and the right by its transpose leads to the formation of the identity matrix

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}_m,$$

the same operations applied to rectangular orthogonal matrices lead to different results

$$\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_n, \boldsymbol{V}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots & \boldsymbol{v}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{v}_i\boldsymbol{v}_i^T, \operatorname{rank}(\boldsymbol{v}_i\boldsymbol{v}_i^T) = 1$$

A simple example is provided by taking $\boldsymbol{V} = \boldsymbol{I}_{m,n}$, the first n columns of the identity matrix in which case

$$\boldsymbol{V}\boldsymbol{V}^T = \sum_{i=1}^{n} \boldsymbol{e}_i\boldsymbol{e}_i^T = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

Applying $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^T$ to some vector $\boldsymbol{b} \in \mathbb{R}^m$ leads to a vector $\boldsymbol{r} = \boldsymbol{P}\boldsymbol{b}$ whose first n components are those of $\boldsymbol{b}$, and the remaining $m - n$ are zero. The subtraction $\boldsymbol{b} - \boldsymbol{r}$ leads to a new vector $\boldsymbol{s} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b}$ that has the first components equal to zero, and the remaining $m - n$ the same as those of $\boldsymbol{b}$. Such operations are referred to as *projections*, and for $\boldsymbol{V} = \boldsymbol{I}_{m,n}$ correspond to projection onto the $\operatorname{span}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$.

Returning to the general case, the orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{V} \in \mathbb{R}^{m\times n}$, $\boldsymbol{W} \in \mathbb{R}^{m\times(m-n)}$ are associated with linear mappings $\boldsymbol{b} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{U}\boldsymbol{x}$, $\boldsymbol{r} = \boldsymbol{g}(\boldsymbol{b}) = \boldsymbol{P}\boldsymbol{b}$, $\boldsymbol{s} = \boldsymbol{h}(\boldsymbol{b}) = (\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{b}$. The mapping $\boldsymbol{f}$ gives the components in the $\boldsymbol{I}$ basis of a vector whose components in the $\boldsymbol{U}$ basis are $\boldsymbol{x}$. The mappings $\boldsymbol{g}, \boldsymbol{h}$ project a vector onto $\text{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, $\text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{m-n}\}$, respectively. When $\boldsymbol{V}, \boldsymbol{W}$ are orthogonal matrices the projections are also orthogonal $\boldsymbol{r} \perp \boldsymbol{s}$. Projection can also be carried out onto nonorthogonal spanning sets, but the process is fraught with possible error, especially when the angle between basis vectors is small, and will be avoided henceforth. Notice that projection of a vector already in the spanning set simply returns the same vector, which leads to a general definition.

DEFINITION. *The mapping is called a projection if $\boldsymbol{f} \circ \boldsymbol{f} = \boldsymbol{f}$, or if for any $\boldsymbol{u} \in U$, $\boldsymbol{f}(\boldsymbol{f}(\boldsymbol{u})) = \boldsymbol{f}(\boldsymbol{u})$. With $\boldsymbol{P}$ the matrix associated $\boldsymbol{f}$, a projection matrix satisfies $\boldsymbol{P}^2 = \boldsymbol{P}$.*

Orthogonal projections onto the column space $C(\boldsymbol{Q})$ of an orthonormal matrix are of great practical utility, and satisfy the above definition

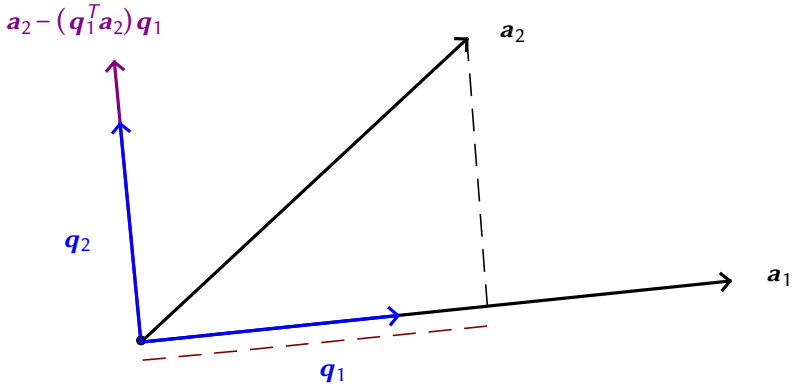
$$\boldsymbol{P}_Q = \boldsymbol{Q}\boldsymbol{Q}^T$$

$$\boldsymbol{P}_Q^2 = \boldsymbol{P}_Q\boldsymbol{P}_Q = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}\,(\boldsymbol{Q}^T\boldsymbol{Q})\,\boldsymbol{Q}^T = \boldsymbol{Q}\boldsymbol{I}\boldsymbol{Q}^T = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{P}_Q.$$

2. Gram-Schmidt algorithm

Orthonormal vector sets $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$ are of the greatest practical utility, leading to the question of whether some such a set can be obtained from an arbitrary set of vectors $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$. This is possible for independent vectors, through what is known as the Gram-Schmidt algorithm

1. Start with an arbitrary direction $\boldsymbol{a}_1$
2. Divide by its norm to obtain a unit-norm vector $\boldsymbol{q}_1 = \boldsymbol{a}_1 / \|\boldsymbol{a}_1\|$
3. Choose another direction $\boldsymbol{a}_2$
4. Subtract off its component along previous direction(s) $\boldsymbol{a}_2 - (\boldsymbol{q}_1^T\boldsymbol{a}_2)\,\boldsymbol{q}_1$
5. Divide by norm $\boldsymbol{q}_2 = (\boldsymbol{a}_2 - (\boldsymbol{q}_1^T\boldsymbol{a}_2)\,\boldsymbol{q}_1) / \|\boldsymbol{a}_2 - (\boldsymbol{q}_1^T\boldsymbol{a}_2)\,\boldsymbol{q}_1\|$
6. Repeat the above



$$\boldsymbol{P}_1\boldsymbol{a}_2 = (\boldsymbol{q}_1\,\boldsymbol{q}_1^T)\boldsymbol{a}_2 = \boldsymbol{q}_1\,(\boldsymbol{q}_1^T\boldsymbol{a}_2) = (\boldsymbol{q}_1^T\boldsymbol{a}_2)\,\boldsymbol{q}_1$$

The above geometrical description can be expressed in terms of matrix operations as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_n \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \dots & \boldsymbol{q}_n \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ 0 & r_{22} & r_{23} & \dots & r_{2n} \\ 0 & 0 & r_{33} & \dots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & r_{mn} \end{pmatrix} = \boldsymbol{QR},$$

equivalent to the system

$$\begin{cases} \boldsymbol{a}_1 = r_{11}\boldsymbol{q}_1 \\ \boldsymbol{a}_2 = r_{12}\boldsymbol{q}_1 + r_{22}\boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{a}_n = r_{1n}\boldsymbol{q}_1 + r_{2n}\boldsymbol{q}_2 + \dots + r_{nn}\boldsymbol{q}_n \end{cases}.$$

The system is easily solved by *forward substitution* resulting in what is known as the (modified) *Gram-Schmidt algorithm*, transcribed below both in pseudo-code and in Julia.

Algorithm (Gram-Schmidt)

Given n vectors $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$
Initialize $\boldsymbol{q}_1 = \boldsymbol{a}_1, .., \boldsymbol{q}_n = \boldsymbol{a}_n$, $\boldsymbol{R} = \boldsymbol{I}_n$
for $i = 1$ to n
  $r_{ii} = (\boldsymbol{q}_i^T \boldsymbol{q}_i)^{1/2}$
  if $r_{ii} < \epsilon$ break;
  $\boldsymbol{q}_i = \boldsymbol{q}_i / r_{ii}$
  for $j = i+1$ to n
    $r_{ij} = \boldsymbol{q}_i^T \boldsymbol{a}_j$; $\boldsymbol{q}_j = \boldsymbol{q}_j - r_{ij}\boldsymbol{q}_i$
  end
end
return $\boldsymbol{Q}, \boldsymbol{R}$

```
∴ function mgs(A)
    m,n=size(A); Q=A; R=eye(n);
    for i=1:n
      R[i,i] = sqrt(Q[:,i]'*Q[:,i]);
      if (R[i,i]<eps) break;
      Q[:,i] = Q[:,i]/R[i,i];
      for j=i+1:n
        R[i,j] = Q[:,i]'*A[:,j];
        Q[:,j] = Q[:,j] - R[i,j]*Q[:,i];
      end;
    end;
    return Q,R
  end
```

The input matrix $\boldsymbol{A}$ might have linearly dependent columns in which case $r_{ii} \approx 0$ for some i, and the if-instruction interrupts the algorithm. The Gram-Schmidt algorithm furnishes a *factorization*

$$\boldsymbol{QR} = \boldsymbol{A}$$

with $\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ an orthonormal matrix and $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ an upper triangular matrix, known as the QR-factorization. Since the column vectors within $\boldsymbol{Q}$ were obtained through linear combinations of the column vectors of $\boldsymbol{A}$, $C(\boldsymbol{A}) = C(\boldsymbol{Q})$. The QR-factorization is of great utility in solving problems within linear algebra.

3. Least squares problems in $\mathbb{R}^m$

3.1. Problem formulation and solution by orthogonal projection

A typical situation in applications is that a vector $\boldsymbol{y} \in \mathbb{R}^m$ represents a complex object with $m \gg 1$. A simpler representation of the object is sought through a linear combination $\boldsymbol{v} = \boldsymbol{Ac}$, with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $n < m$, usually $m \ll n$ (Fig. 1).

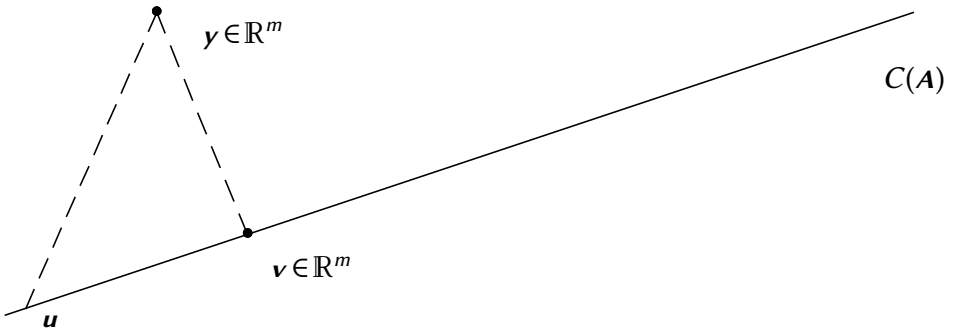


Figure 1. Least squares problem: find $\boldsymbol{v} \in C(\boldsymbol{A})$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ closest to some given $\boldsymbol{y}$ in the 2-norm

The magnitude of the difference between the exact object $\boldsymbol{y}$ and the approximation $\boldsymbol{Ac}$ is measured through a norm, and in least squares the 2-norm is adopted. The problem is to minimize the error $\varepsilon = \| \boldsymbol{e} \| = \| \boldsymbol{y} - \boldsymbol{Ac} \| = (\boldsymbol{e}^T \boldsymbol{e})^{1/2}$. This is stated mathematically as

$$\min_{\boldsymbol{c} \in \mathbb{R}^n} \| \boldsymbol{y} - \boldsymbol{Ac} \|,$$

and within $\mathbb{R}^m$ the minimal (2-norm) distance is obtained when $\boldsymbol{y} - \boldsymbol{v}$ is orthogonal to $C(\boldsymbol{A})$. Note that is another type of norm were to be adopted, the orthogonality condition would no longer necessarily hold. For the 2-norm however, the orthogonality condition leads to a straightforward solution of the problem through orthogonal projection:

1. Find an orthonormal basis for column space of $\boldsymbol{A}$ by $\boldsymbol{QR}$ factorization, $\boldsymbol{QR} = \boldsymbol{A}$.
2. State that $\boldsymbol{v}$ is the projection of $\boldsymbol{y}$, $\boldsymbol{v} = \boldsymbol{P}_{C(\boldsymbol{A})} \boldsymbol{y} = \boldsymbol{P}_{\boldsymbol{Q}} \boldsymbol{y} = \boldsymbol{Q} \boldsymbol{Q}^T \boldsymbol{y}$.
3. State that $\boldsymbol{v}$ is within the column space of $\boldsymbol{A}$, $\boldsymbol{v} = \boldsymbol{Ac} = \boldsymbol{QRc}$.
4. Set equal the two expressions of $\boldsymbol{v}$, $\boldsymbol{Q} \boldsymbol{Q}^T \boldsymbol{y} = \boldsymbol{QRc}$. This is an equality between two linear combinations of the columns of $\boldsymbol{Q}$. For $\boldsymbol{Q}$ orthonormal the scaling coefficients of the two linear combinations must be equal leading to $\boldsymbol{Rc} = \boldsymbol{Q}^T \boldsymbol{y}$.
5. Solve the triangular system to find $\boldsymbol{c}$.

3.2. Linear regression

The approach to compressing data $D = \{(x_i, y_i) \mid i = 1, \ldots, m\}$ suggested by calculus concepts is to form the sum of squared differences between $y(x_i)$ and y_i, for example for $y(x) = c_0 + c_1 x$ when carrying out linear regression,

$$S(c_0, c_1) = \sum_{i=1}^{m} (y(x_i) - y_i)^2 = \sum_{i=1}^{m} (c_0 + c_1 x_i - y_i)^2$$

and seek (c_0, c_1) that minimize $S(c_0, c_1)$. The function $S(c_0, c_1) \geq 0$ can be thought of as the height of a surface above the $c_0 c_1$ plane, and the gradient ∇S is defined as a vector in the direction of steepest slope. When at some point on the surface if the gradient is different from the zero vector $\nabla S \neq \mathbf{0}$, travel in the direction of the gradient would increase the height, and travel in the opposite direction would decrease the height. The minimal value of S would be attained

when no local travel could decrease the function value, which is known as stationarity condition, stated as $\nabla S = 0$. Applying this to determining the coefficients (c_0, c_1) of a linear regression leads to the equations

$$\frac{\partial S}{\partial c_0} = 0 \Longrightarrow 2\sum_{i=1}^{m} (c_0 + c_1 x_i - y_i) = 0 \Longleftrightarrow m c_0 + \left(\sum_{i=1}^{m} x_i\right) c_1 = \sum_{i=1}^{m} y_i,$$

$$\frac{\partial S}{\partial c_1} = 0 \Longrightarrow 2\sum_{i=1}^{m} (c_0 + c_1 x_i - y_i) x_i = 0 \Longleftrightarrow \left(\sum_{i=1}^{m} x_i\right) c_0 + \left(\sum_{i=1}^{m} x_i^2\right) c_1 = \sum_{i=1}^{m} x_i y_i.$$

The above calculations can become tedious, and do not illuminate the geometrical essence of the calculation, which can be brought out by reformulation in terms of a matrix-vector product that highlights the particular linear combination that is sought in a liner regression. Form a vector of errors with components $e_i = y(x_i) - y_i$, which for linear regression is $y(x) = c_0 + c_1 x$. Recognize that $y(x_i)$ is a linear combination of 1 and x_i with coefficients c_0, c_1, or in vector form

$$\boldsymbol{e} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} - \boldsymbol{y} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x} \end{bmatrix} \boldsymbol{c} - \boldsymbol{y} = \boldsymbol{A}\boldsymbol{c} - \boldsymbol{y}.$$

The norm of the error vector $\|\boldsymbol{e}\|$ is smallest when $\boldsymbol{A}\boldsymbol{a}$ is as close as possible to $\boldsymbol{y}$. Since $\boldsymbol{A}\boldsymbol{a}$ is within the column space of $C(\boldsymbol{A})$, $\boldsymbol{A}\boldsymbol{a} \in C(\boldsymbol{A})$, the required condition is for $\boldsymbol{e}$ to be orthogonal to the column space

$$\boldsymbol{e} \perp C(\boldsymbol{A}) \Longrightarrow \boldsymbol{A}^T\boldsymbol{e} = \begin{bmatrix} \mathbf{1}^T \\ \boldsymbol{x}^T \end{bmatrix} \boldsymbol{e} = \begin{bmatrix} \mathbf{1}^T\boldsymbol{e} \\ \boldsymbol{x}^T\boldsymbol{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{0}$$

$$\boldsymbol{A}^T\boldsymbol{e} = \mathbf{0} \Longleftrightarrow \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{c} - \boldsymbol{y}) = 0 \Longleftrightarrow (\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{c} = \boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{N}\boldsymbol{c} = \boldsymbol{b}.$$

The above is known as the normal system, with $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ is the normal matrix. The system $\boldsymbol{N}\boldsymbol{c} = \boldsymbol{b}$ can be interpreted as seeking the coordinates in the $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ basis of the vector $\boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{y}$. An example can be constructed by randomly perturbing a known function $y(x) = a_0 + a_1 x$ to simulate measurement noise and compare to the approximate $\tilde{c}$ obtained by solving the normal system.

```
∴ m=100; x=(0:m-1)./m; c0=2; c1=3; yex=c0.+c1*x; y=(yex.+rand(m,1).-0.5);
```

```
∴ A=ones(m,2); A[:,2]=x[:]; At=transpose(A); N=At*A; b=At*y;
```

```
∴ c = N\b
```

$$\begin{bmatrix} 1.9524719146017488 \\ 3.114552636517776 \end{bmatrix} \tag{1}$$

```
∴
```

3.3. Least squares polynomial approximations of data

Forming the normal system of equations can lead to numerical difficulties, especially when the columns of $\boldsymbol{A}$ are close to linear dependence. It is preferable to adopt the general procedure of solving a least squares problem by projection, in which case the above linear regression becomes:

$$\boldsymbol{Q}\boldsymbol{R} = \boldsymbol{A}, \boldsymbol{R}\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{y}.$$

```
∴ QR=qr(A); Q=QR.Q[:,1:2]; R=QR.R[1:2,1:2];
```

```
∴ c = R\(transpose(Q)*y)
```

$$\begin{bmatrix} 1.9524719146017504 \\ 3.114552636517771 \end{bmatrix} \tag{2}$$

```
∴
```

The above procedure can be easily extended to define quadratic or cubic regression, the problem of finding the best polynomials of degree 2 or 3 that fit the data. Quadratic regression is simply accomplished by adding a column to $\boldsymbol{A}$ containing the squares of the $\boldsymbol{x}$ vector

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x} & \boldsymbol{x}^2 \end{bmatrix}$$

with the column vector $\boldsymbol{a}_k = \boldsymbol{x}^{k-1} \in \mathbb{R}^m$ has components x_i^{k-1} for $i = 1, 2, \ldots, m$.

```
∴ m=100; x=(0:m-1)./m; c0=2; c1=3; c2=-5; yex=c0.+c1*x.+c2*x.^2;
```

```
∴ y=(yex.+rand(m,1).-0.5);
```

```
∴ A=ones(m,3);   A[:,2]=x[:];   A[:,3]=x[:].^2;   QR=qr(A);   Q=QR.Q[:,1:3];
  R=QR.R[1:3,1:3];
```

```
∴ c = R\(transpose(Q)*y)
```

$$\begin{bmatrix} 1.968925863819198 \\ 2.9292845098793223 \\ -4.836797382728068 \end{bmatrix} \tag{3}$$

```
∴
```