TOPIC 1

Least squares: Cholesky, QR and Householder

3 Overdetermined systems, positive semidefinite matrices and Cholesky decomposition

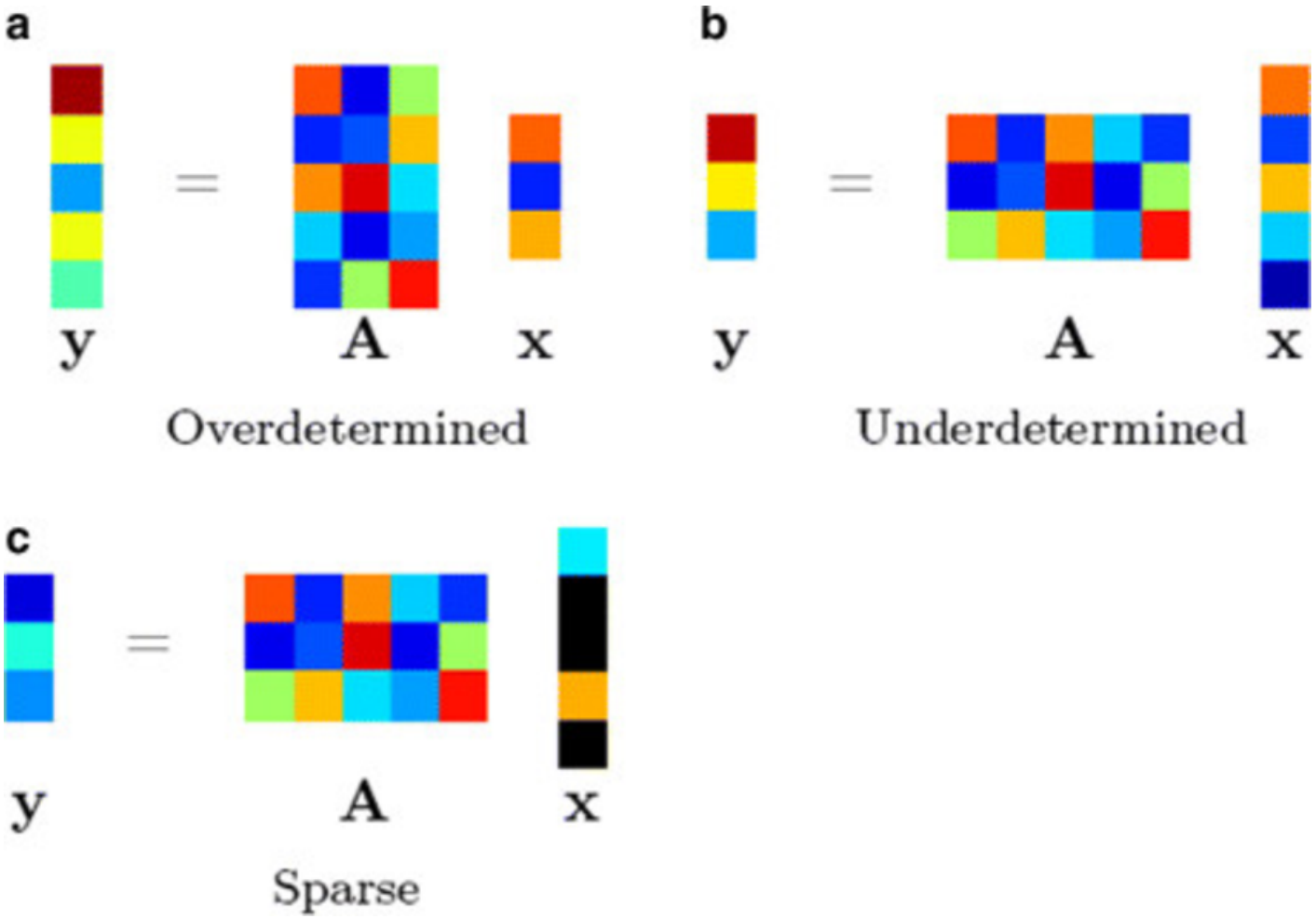
Course: Math 535 (http://www.math.wisc.edu/~roch/mmids/) - Mathematical Methods in Data Science (MMiDS)
Author: Sebastien Roch (http://www.math.wisc.edu/~roch/), Department of Mathematics, University of Wisconsin-Madison
Updated: Sep 21, 2020

3.1 Solving an overdetermined linear system

In this section, we discuss the least-squares problem and return to regression. Let $A \in \mathbb{R}^{n \times m}$ be an $n \times m$ matrix with linearly independent columns and let $\mathbf{b} \in \mathbb{R}^n$ be a vector. We are looking to solve the system

$$A\mathbf{x} \approx \mathbf{b}.$$

If $n = m$, that is, if A is a square matrix, we can use the matrix inverse (https://en.wikipedia.org/wiki/Invertible_matrix) to solve the system. But we are particularly interested in the overdetermined case, i.e. when $n > m$: there are more equations than variables. We cannot use the matrix inverse then.



(Source) (https://www.researchgate.net/figure/Stylized-visualization-of-three-examples-of-linear-equation-systems-A-and-y-represent_fig14_287108600)

Exercise: Let $A \in \mathbb{R}^{n \times m}$ be an $n \times m$ matrix with linearly independent columns. Show that $m \leq n$.◁

A natural way to make sense of the overdetermined problem is to cast it as the [least-squares problem (https://en.wikipedia.org/wiki/Least_squares)]

$$\min_{\mathbf{x}\in\mathbb{R}^m} \|A\mathbf{x} - \mathbf{b}\|.$$

In words, we look for the best-fitting solution under the Euclidean norm. Equivalently, writing

$$A = \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \dots & \mathbf{a}_m \\ | & & | \end{pmatrix} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,m} \\ a_{2,1} & \cdots & a_{2,m} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,m} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

we seek a linear combination of the columns of A that minimizes the objective

$$\left\| \sum_{j=1}^{m} x_j \mathbf{a}_j - \mathbf{b} \right\|^2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} x_j a_{i,j} - b_i \right)^2.$$

We already have a solution to this problem: the orthogonal projection.

Theorem (Normal Equations): Let $A \in \mathbb{R}^{n\times m}$ be an $n \times m$ matrix with $n \geq m$ and let $\mathbf{b} \in \mathbb{R}^n$ be a vector. A solution $\mathbf{x}^*$ to the least-squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^m} \|A\mathbf{x} - \mathbf{b}\|$$

satisfies the normal equations

$$A^T A\mathbf{x}^* = A^T\mathbf{b}.$$

If further the columns of A are linearly independent, then there exists a unique solution $\mathbf{x}^*$.

Proof idea: Apply our characterization of the orthogonal projection onto the column space of A.

Proof: Let $\mathcal{U} = \mathrm{col}(A) = \mathrm{span}(\mathbf{a}_1, \dots, \mathbf{a}_m)$. By the *Orthogonal Projection Theorem*, the orthogonal projection $\mathbf{b}^* = \mathcal{P}_{\mathcal{U}}\mathbf{b}$ of $\mathbf{b}$ onto $\mathcal{U}$ is the closest vector to $\mathbf{b}$ in $\mathcal{U}$. Because $\mathbf{b}^*$ is in $\mathcal{U} = \mathrm{col}(A)$, it must be of the form $\mathbf{b}^* = A\mathbf{x}^*$ and therefore it is a solution to the least-squares problem above. In particular, it must satisfy $\langle \mathbf{b} - A\mathbf{x}^*, \mathbf{u} \rangle = 0$ for all $\mathbf{u} \in \mathcal{U}$. Because the columns $\mathbf{a}_i$ are in $\mathcal{U}$, it implies that $\langle \mathbf{b} - A\mathbf{x}^*, \mathbf{a}_i \rangle = 0$ for all $i \in [m]$. Stacking up the equations

$$\langle \mathbf{b} - A\mathbf{x}^*, \mathbf{a}_i \rangle = \mathbf{a}_i^T(A\mathbf{x}^* - \mathbf{b}) = 0$$

gives in matrix form

$$A^T(A\mathbf{x}^* - \mathbf{b}) = \mathbf{0},$$

as claimed. We have seen in a previous example that, when A has full column rank, the matrix $A^T A$ is invertible. That proves the uniqueness claim. □

NUMERICAL CORNER To solve a linear system in Julia, use `\` . As an example, we consider the overdetermined system with

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

In [1]:
```
# Julia version: 1.5.1
using Plots, LinearAlgebra
```

In [2]:
```
w1, w2 = [1.,0.,1.], [0.,1.,1.]
b = [0.,0.,1.]
A = hcat(w1,w2)
```

Out[2]:
```
3×2 Array{Float64,2}:
 1.0  0.0
 0.0  1.0
 1.0  1.0
```

In [3]:
```
x = A'*A \ A'*b
```

Out[3]:
```
2-element Array{Float64,1}:
 0.33333333333333337
 0.3333333333333333
```

We can also use `\` directly on the overdetermined system to compute the least-square solution.

In [4]:
```
x = A \ b
```

Out[4]:
```
2-element Array{Float64,1}:
 0.3333333333333333
 0.33333333333333326
```

3.2 Least squares via Cholesky

You have seen in a first linear algebra course how to solve a square linear system, such as the normal equations. For this task a common approach is Gaussian elimination, or row reduction. Quoting Wikipedia (https://en.wikipedia.org/wiki/Gaussian_elimination):

> To perform row reduction on a matrix, one uses a sequence of elementary row operations to modify the matrix until the lower left-hand corner of the matrix is filled with zeros, as much as possible. [...] Once all of the leading coefficients (the leftmost nonzero entry in each row) are 1, and every column containing a leading coefficient has zeros elsewhere, the matrix is said to be in reduced row echelon form. [...] The process of row reduction [...] can be divided into two parts. The first part (sometimes called forward elimination) reduces a given system to row echelon form, from which one can tell whether there are no solutions, a unique solution, or infinitely many solutions. The second part (sometimes called back substitution) continues to use row operations until the solution is found; in other words, it puts the matrix into reduced row echelon form.

Here is an example.

$$\left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 5 & 35 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 2 & 8 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 0 & 0 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 0 & -2 & -3 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

(Source) (https://en.wikipedia.org/wiki/Gaussian_elimination)

We will not go over Gaussian elimination here. Instead we will derive a modifed approach to solve the normal equations that takes advantage of the special structure of this system to compute the solution twice as fast. In the process, we will also obtain an important notion of matrix factorization.

3.2.1 Triangular systems

We will need one component of Gaussian elimination, back-substitution.

Definition (Triangular matrix): A matrix $U \in \mathbb{R}^{n \times n}$ is upper-triangular if all entries below the diagonal are zero. A lower-triangular matrix is defined similarly. $\lhd$

So an upper-triangular matrix is of the following form:

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

(Source) (https://en.wikipedia.org/wiki/Triangular_matrix)

Triangular systems of equations are straightforward to solve. It works as follows. Let $U \in \mathbb{R}^{m \times m}$ be upper-triangular and let $\mathbf{b} \in \mathbb{R}^m$ be the left-hand vector, i.e., we want to solve the system

$$U\mathbf{x} = \mathbf{b}.$$

Starting from the last row of the system, $U_{m,m}x_m = b_m$ or $x_m = b_m/U_{m,m}$, assuming that $U_{m,m} \neq 0$. Moving to the second-to-last row, $U_{m-1,m-1}x_{m-1} + U_{m-1,m}x_m = b_{m-1}$ or $x_{m-1} = (b_{m-1} - U_{m-1,m}x_m)/U_{m-1,m-1}$, assuming that $U_{m-1,m-1} \neq 0$. And so on. This procedure is known as back substitution (https://en.wikipedia.org/wiki/Triangular_matrix#Forward_and_back_substitution).

Analogously, in the lower triangular case, we have forward substitution (https://en.wikipedia.org/wiki/Triangular_matrix#Forward_substitution). These procedures implicitly define an inverse for U and L *when the diagonal elements are all non-zero*. We will not write them down explicitly here.

NUMERICAL CORNER We implement back substitution in Julia. In our naive implementation, we assume that the diagonal entries are not zero, which will suffice for our purposes.

In [5]:
```
function mmids_backsubs(U,b)
    m = length(b)
    x = zeros(Float64,m)
    for i=m:-1:1
        x[i] = (b[i] - dot(U[i,i+1:m],x[i+1:m]))/U[i,i] # assumes non-ze
ro diagonal
    end
    return x
end
```

Out[5]:
```
mmids_backsubs (generic function with 1 method)
```

Forward substitution is implemented similarly.

In [6]:
```
function mmids_forwardsubs(L,b)
    m = length(b)
    x = zeros(Float64,m)
    for i=1:m
        x[i] = (b[i] - dot(L[i,1:i-1],x[1:i-1]))/L[i,i] # assumes non-zero diagonal
    end
    return x
end
```

Out[6]: mmids_forwardsubs (generic function with 1 method)

3.2.2 Positive semidefinite matrices

What special structure does the matrix $A^T A$ have? For one, it is square and symmetric.

Definition (Symmetric Matrix): A square matrix $B \in \mathbb{R}^{n\times n}$ is symmetric if $B^T = B$. ◁

By a previous exercise, $(A^T A)^T = A^T A$. That is, $A^T A$ is symmetric.

The matrix $A^T A$ also has a less obvious, but important property.

Definition (Positive Semidefinite Matrix): A symmetric matrix $B \in \mathbb{R}^{n\times n}$ is positive semidefinite if

$$\langle \mathbf{x}, B\mathbf{x}\rangle \geq 0, \quad \forall \mathbf{x} \neq \mathbf{0}.$$

We also write $B \succeq 0$ in that case. If the inequality above is strict, we say that B is positive definite in which case we write $B \succ 0$. ◁

Note that by definition a positive semidefinite matrix is necessarily symmetric. (For a discussion on this point, see here (https://math.stackexchange.com/questions/1954167/do-positive-semidefinite-matrices-have-to-be-symmetric).)

Exercise: Let $A = [a]$ be a 1×1 positive definite matrix. Show that $a > 0$. ◁

Example: Consider a square, symmetric matrix of the form $B = \mathrm{diag}(\beta_1, \ldots, \beta_d)$, i.e., all non-diagonal entries of B are 0 and its diagonal elements are $b_{ii} = \beta_i, \forall i$.

We claim first that

$$(*) \qquad \langle \mathbf{x}, B\mathbf{x}\rangle = \sum_{i=1}^{d} \beta_i x_i^2, \qquad \forall \mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d.$$

Indeed, by the diagonal structure of B, we have that $B\mathbf{x} = (\beta_1 x_1, \ldots, \beta_d x_d)^T$. Equation $(*)$ immediately follows.

Using $(*)$ we prove

$$(**) \qquad B \succeq 0 \iff \beta_i \geq 0, \forall i.$$

From $(*)$ it is immediate that, if $\beta_i \geq 0, \forall i$, it holds that $\langle \mathbf{x}, B\mathbf{x} \rangle = \sum_{i=1}^{d} \beta_i x_i^2 \geq 0$ for all $\mathbf{x}$ since each term in the sum is nonnegative. For the other direction, we argue by contradiction. Suppose that $B \succeq 0$ and that there is a $\beta_i < 0$. Then, for the unit basis vector $\mathbf{x} = \mathbf{e}_i$, we get from $(**)$ that

$$\langle \mathbf{x}, B\mathbf{x} \rangle = \sum_{i=1}^{d} \beta_i x_i^2 = \beta_i (1)^2 < 0,$$

a contradiction. $\lhd$

Exercise: Prove an analogue of the previous example for positive definite matrices. $\lhd$

Example: Perhaps counter-intuitively, in general, $B \succeq 0$ is *not* the same as B having only nonnegative elements. Here is an example. Consider the matrix

$$A = \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix}.$$

While all of its elements are nonnegative, it is not positive semidefinite. Indeed let $\mathbf{x} = (1, -1)^T$. Then

$$\langle \mathbf{x}, A\mathbf{x} \rangle = (1, -1)\,(1 - 10, 10 - 1)^T = (1, -1)\,(-9, 9)^T = -18 < 0.$$

$\lhd$

We return to our main example.

Lemma (Least Squares and Postive Semidefiniteness): Let $A \in \mathbb{R}^{n \times m}$ be an $n \times m$ matrix with $n \geq m$. The matrix $B = A^T A$ is positive semidefinite. If further the columns of A are linearly independent, then the matrix B is positive definite.

Proof: As we have observed in a previous example (see Section 1.4 of Topic 1), for any $\mathbf{x}$,

$$\mathbf{x}^T (A^T A)\mathbf{x} = (A\mathbf{x})^T (A\mathbf{x}) = \|A\mathbf{x}\|^2 \geq 0.$$

Hence $B \succeq 0$. If the inequality above is an equality, by the point-separating property of the Euclidean norm, that means that we must have $A\mathbf{x} = \mathbf{0}$. If A has full column rank, the *Equivalent Definition of Linear Independence* implies that $\mathbf{x} = \mathbf{0}$, which establishes the second claim. $\square$

Before introducing the Cholesky decomposition of positive definite matrices, we will need a few more properties of this important class of matrices. All three properties follow more or less immediately from the definitions of positive definiteness and linear independence.

Recall that a block matrix is a partitioning of the rows and columns of a matrix of the form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where $A \in \mathbb{R}^{n \times m}$, $A_{ij} \in \mathbb{R}^{n_i \times m_j}$ for $i, j = 1, 2$ with the conditions $n_1 + n_2 = n$ and $m_1 + m_2 = m$. More generally, one can consider larger numbers of blocks. Block matrices have a convenient algebra that mimics the usual matrix algebra. Specifically, if $B_{ij} \in \mathbb{R}^{m_i \times p_j}$ for $i, j = 1, 2$, then it holds that

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}.$$

Observe that the block sizes of A and B must match for this formula to make sense. You can convince yourself of this identity by trying it on a simple example. For a (somewhat tedious) proof, see e.g. here (https://sites.math.washington.edu/~morrow/498_13/blockmatrices.pdf).

Lemma (Invertibility of Positive Definite Matrices): Let $B \in \mathbb{R}^{n \times n}$ be positive definite. Then B is invertible.

Proof: For any $\mathbf{x} \neq \mathbf{0}$, it holds by positive definiteness that $\mathbf{x}^T B\mathbf{x} > 0$. In particular, it must be that $\mathbf{x}^T B\mathbf{x} \neq 0$ and therefore, by contradiction, $B\mathbf{x} \neq \mathbf{0}$ (since for any $\mathbf{z}$, it holds that $\mathbf{z}^T \mathbf{0} = 0$). The claim follows from the *Equivalent Definition of Linear Independence*. □

A principal submatrix is a square submatrix obtained by removing certain rows and columns. Moreover we require that the set of row indices that remain is the same as the set of column indices that remain.

Lemma (Principal Submatrices): Let $B \in \mathbb{R}^{n \times n}$ be positive definite and let $Z \in \mathbb{R}^{n \times p}$ have full column rank. Then $Z^T BZ$ is positive definite. In particular all principal submatrices of positive definite matrices are positive definite.

Proof: If $\mathbf{x} \neq \mathbf{0}$, then $\mathbf{x}^T (Z^T BZ)\mathbf{x} = \mathbf{y}^T B\mathbf{y}$, where we defined $\mathbf{y} = Z\mathbf{x}$. Because Z has full column rank and $\mathbf{x} \neq \mathbf{0}$, it follows that $\mathbf{y} \neq \mathbf{0}$ by the *Equivalent Definition of Linear Independence*. Hence, since $B \succ 0$, we have $\mathbf{y}^T B\mathbf{y} > 0$ which proves the first claim. For the second claim, take Z of the form $(\mathbf{e}_{i_1} \ \mathbf{e}_{i_2} \ \dots \ \mathbf{e}_{i_p})$, where the indices $i_1, \dots, i_p$ are distinct. The columns of Z are then linearly independent since they are distinct basis vectors. □

Exercise: Show that the diagonal elements of a positive definite matrix are necesarily positive. ◁

Lemma (Schur Complement): Let $B \in \mathbb{R}^{n \times n}$ be positive definite and write it in block form

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{pmatrix}$$

where $B_{11} \in \mathbb{R}^{n_1 \times n_1}$, $B_{12} \in \mathbb{R}^{n_1 \times n - n_1}$ and $B_{22} \in \mathbb{R}^{n-n_1 \times n-n_1}$. Then the Schur complement of the block B_{11}, i.e. the matrix $B_{22} - B_{12}^T B_{11}^{-1} B_{12}$, is positive definite.

Proof: By the *Principal Submatrices Lemma*, B_{11} is positive definite. By the *Invertibility of Positive Definite Matrices Lemma*, B_{11} is therefore invertible. Hence the Schur complement is well defined. The result then follows from the observation (proved in the following exercise): for any $\mathbf{x}$

$$\mathbf{x}^T (B_{22} - B_{12}^T B_{11}^{-1} B_{12}) \mathbf{x} = \mathbf{z}^T \begin{pmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{pmatrix} \mathbf{z} \quad \text{where} \quad \mathbf{z} = \begin{pmatrix} B_{11}^{-1} B_{12} \mathbf{x} \\ -\mathbf{x} \end{pmatrix}.$$

□

Exercise: Check the calculation in the previous proof. ◁

3.2.3 Cholesky decomposition

Our key linear-algebraic result of this section is the following. The matrix factorization in the next theorem is called a Cholesky decomposition. It has many applications (https://en.wikipedia.org/wiki/Cholesky_decomposition#Applications).

Theorem (Cholesky Decomposition): Any positive definite matrix $B \in \mathbb{R}^{n \times n}$ can be factorized uniquely as

$$B = LL^T$$

where $L \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with positive entries on the diagonal.

Proof idea: Assuming by induction that the upper-left corner of the matrix B has a Cholesky decomposition, one finds equations for the remaining row that can be solved uniquely by the properties established in the previous subsection.

Proof: If $n = 1$, a previous exercise shows that $b_{11} > 0$, and hence we can take $L = [\ell_{11}]$ where $\ell_{11} = \sqrt{b_{11}}$. Assuming the result holds for positive definite matrices in $\mathbb{R}^{n-1\times n-1}$, we first re-write $B = LL^T$ in block form

$$\begin{pmatrix} \boldsymbol{B}_{11} & \boldsymbol{\beta}_{12} \\ \boldsymbol{\beta}_{12}^T & \beta_{22} \end{pmatrix} = \begin{pmatrix} \Lambda_{11} & \mathbf{0} \\ \boldsymbol{\lambda}_{12}^T & \lambda_{22} \end{pmatrix} \begin{pmatrix} \Lambda_{11}^T & \boldsymbol{\lambda}_{12} \\ \mathbf{0}^T & \lambda_{22} \end{pmatrix}$$

where $\boldsymbol{B}_{11}, \Lambda_{11} \in \mathbb{R}^{n-1\times n-1}$, $\boldsymbol{\beta}_{12}, \boldsymbol{\lambda}_{12} \in \mathbb{R}^{n-1}$ and $\beta_{22}, \lambda_{22} \in \mathbb{R}$. By block matrix algebra, we get the system

$$\begin{aligned} \boldsymbol{B}_{11} &= \Lambda_{11}\Lambda_{11}^T \\ \boldsymbol{\beta}_{12} &= \Lambda_{11}\boldsymbol{\lambda}_{12} \\ \beta_{22} &= \boldsymbol{\lambda}_{12}^T\boldsymbol{\lambda}_{12} + \lambda_{22}^2. \end{aligned}$$

By the *Principal Submatrices Lemma*, the principal submatrix $\boldsymbol{B}_{11}$ is positive definite. Hence, by induction, there is a unique lower-triangular matrix Λ_{11} with positive diagonal elements satisfying the first equation. We can then obtain $\boldsymbol{\beta}_{12}$ from the second equation by forward substitution. And finally we get

$$\lambda_{22} = \sqrt{\beta_{22} - \boldsymbol{\lambda}_{12}^T\boldsymbol{\lambda}_{12}}.$$

We do have to check that the square root above exists. That is, we need to argue that the expression inside the square root is non-negative. In fact, for the claim to go through, we need it to be strictly positive. We notice that the expression inside the square root is in fact the Schur complement of the block $\boldsymbol{B}_{11}$:

$$\begin{aligned} \beta_{22} - \boldsymbol{\lambda}_{12}^T\boldsymbol{\lambda}_{12} &= \beta_{22} - (\Lambda_{11}^{-1}\boldsymbol{\beta}_{12})^T(\Lambda_{11}^{-1}\boldsymbol{\beta}_{12}) \\ &= \beta_{22} - \boldsymbol{\beta}_{12}^T(\Lambda_{11}^{-1})^T\Lambda_{11}^{-1}\boldsymbol{\beta}_{12} \\ &= \beta_{22} - \boldsymbol{\beta}_{12}^T(\Lambda_{11}\Lambda_{11}^T)^{-1}\boldsymbol{\beta}_{12} \\ &= \beta_{22} - \boldsymbol{\beta}_{12}^T(\boldsymbol{B}_{11})^{-1}\boldsymbol{\beta}_{12} \end{aligned}$$

where we used the equation $\boldsymbol{\beta}_{12} = \Lambda_{11}\boldsymbol{\lambda}_{12}$ on the first line, the identities $(QW)^{-1} = W^{-1}Q^{-1}$ and $(Q^T)^{-1} = (Q^{-1})^T$ (see the exercise below) on the third line and the equation $\boldsymbol{B}_{11} = \Lambda_{11}\Lambda_{11}^T$ on the fourth line. By the *Schur Complement Lemma*, the Schur complement is positive definite. Because it is a scalar in this case, it is strictly positive (by a previous exercise), which concludes the proof. □

Exercise: Let $Q, W \in \mathbb{R}^{n\times n}$ be invertible. Show that $(QW)^{-1} = W^{-1}Q^{-1}$ and $(Q^T)^{-1} = (Q^{-1})^T$. ◁

An important consequence of the proof above is an algorithm for computing the Cholesky decomposition. Indeed it follows from the equations in the proof that we can grow L starting from its top-left corner by successively computing its next row based on the previously constructed submatrix. Note that, because L is lower triangular, it suffices to compute its elements on and below the diagonal.

Write $B = [b_{ij}]_{i,j=1}^n$ and $L = [\ell_{ij}]_{i,j=1}^n$. Let $L_{(k)} = [\ell_{ij}]_{i,j=1}^k$ be the first k rows and columns of L, let $\lambda_{(k)}^T = (\ell_{k,1}, \ldots, \ell_{k,k-1})$ be the row vector corresponding to the first $k-1$ entries of row k of L, and let $\beta_{(k)}^T = (b_{k,1}, \ldots, b_{k,k-1})$ be the row vector corresponding to the first $k-1$ entries of row k of B. We first have

$$L_{(1)} = \ell_{11} = \sqrt{b_{11}}.$$

Assuming $L_{(j-1)}$ has been constructed, we then have

$$L_{(j-1)}\lambda_{(j)} = \beta_{(j)}$$

which can be solved by forward substitution, and

$$\ell_{jj} = \sqrt{b_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2}.$$

From this, we construct $L_{(j)}$ as follows:

$$L_{(j)} = \begin{pmatrix} L_{(j-1)} & \mathbf{0} \\ \lambda_{(j)}^T & \ell_{jj} \end{pmatrix}.$$

Here is an illustration of the flow of this construction.

(Source) (https://en.wikipedia.org/wiki/File:Chol.gif)

NUMERICAL CORNER We implement the algorithm above. In our naive implementation, we assume that B is positive definite, and therefore that all steps are well-defined.

In [7]:
```
function mmids_cholesky(B)
    n = size(B)[1] # number of rows
    L = zeros(Float64, n, n) # initiallization of L
    for j=1:n
        L[j,1:j-1] = mmids_forwardsubs(L[1:j-1,1:j-1],B[j,1:j-1])
        L[j,j] = sqrt(B[j,j] - norm(L[j,1:j-1])^2)
    end
    return L
end
```

Out[7]: mmids_cholesky (generic function with 1 method)

Here is a simple example.

In [8]:
```
B = [2. 1.; 1. 2.]
```

Out[8]:
```
2×2 Array{Float64,2}:
 2.0  1.0
 1.0  2.0
```

In [9]:
```
L = mmids_cholesky(B)
```

Out[9]:
```
2×2 Array{Float64,2}:
 1.41421   0.0
 0.707107  1.22474
```

We can check that it produces the right factorization.

In [10]:
```
L*L'
```

Out[10]:
```
2×2 Array{Float64,2}:
 2.0  1.0
 1.0  2.0
```

3.2.4 Using a Cholesky decomposition to solve the least squares problem

In this section, we restrict ourselves to the case where $A \in \mathbb{R}^{n \times m}$ has full column rank. By the *Least Squares and Positive Semidefiniteness Lemma*, we then have that $A^T A$ is positive definite. By the *Cholesky Decomposition Theorem*, we can factorize this matrix as $A^T A = L L^T$ where L is lower triangular with positive diagonal elements. The normal equations then reduce to

$$L L^T \mathbf{x} = A^T \mathbf{b}.$$

This system can be solved in two steps. We first obtain the solution to

$$L \mathbf{z} = A^T \mathbf{b}$$

by forward substitution. Then we obtain the solution to

$$L^T \mathbf{x} = \mathbf{z}$$

by back-substitution. Note that L^T is indeed an upper triangular matrix.

NUMERICAL CORNER We implement this algorithm above. In our naive implementation, we assume that A has full column rank, and therefore that all steps are well-defined.

In [11]:
```
function ls_by_chol(A, b)
    L = mmids_cholesky(A'*A)
    z = mmids_forwardsubs(L, A'*b)
    return mmids_backsubs(L', z)
end
```

Out[11]:
```
ls_by_chol (generic function with 1 method)
```

3.3 Regression

We return to the regression problem and apply the least squares approach.

3.3.1 Linear regression

We seek an affine function to fit input data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. The common approach involves finding coefficients β_j's that minimize the criterion

$$\sum_{i=1}^{n} \left(y_i - \left\{ \beta_0 + \sum_{j=1}^{d} \beta_j x_{i,j} \right\} \right)^2.$$

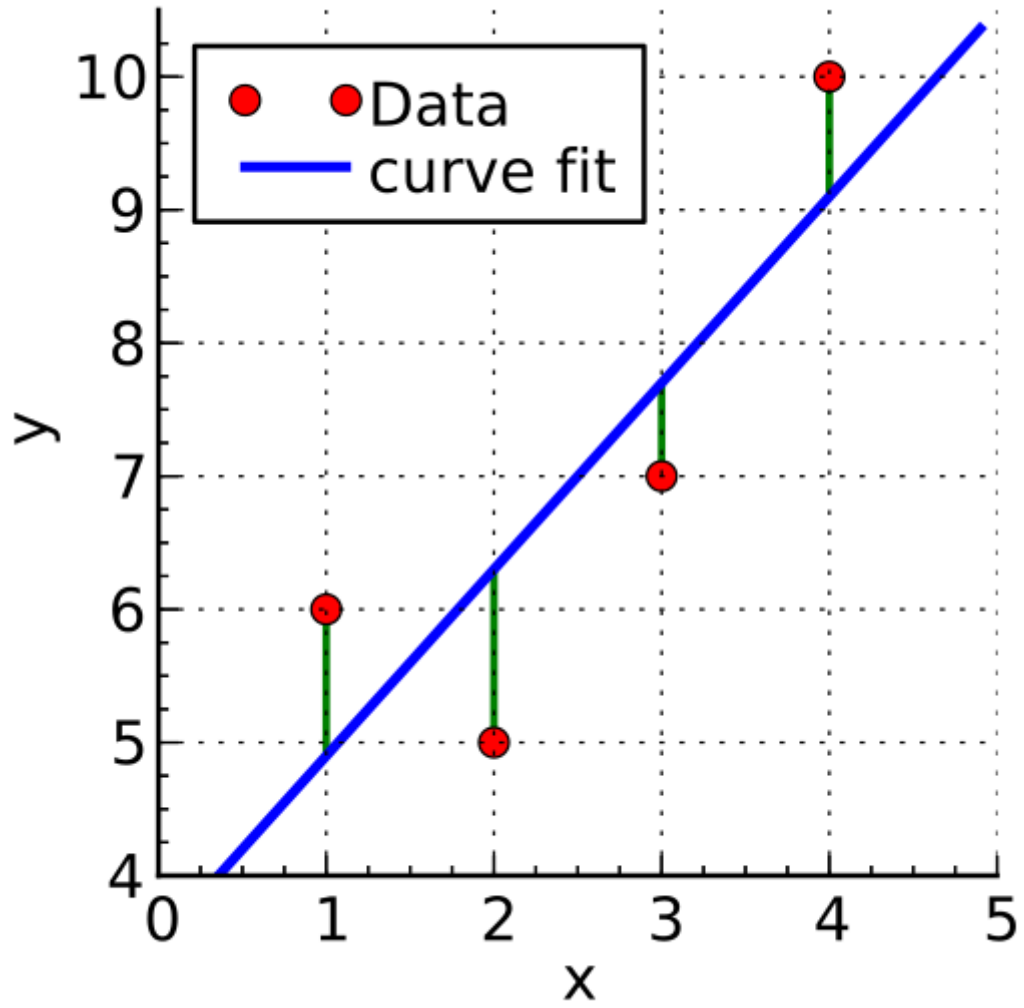
This is indeed a least-squares problem.

We re-write it in matrix form. Let

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^T \end{pmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix}.$$

Then the problem is

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - A\boldsymbol{\beta}\|^2.$$



(Source (https://commons.wikimedia.org/wiki/File:Linear_least_squares_example2.svg))

We assume that the columns of the X matrix are linearly independent, which is typically the case with real data. The normal equations are then

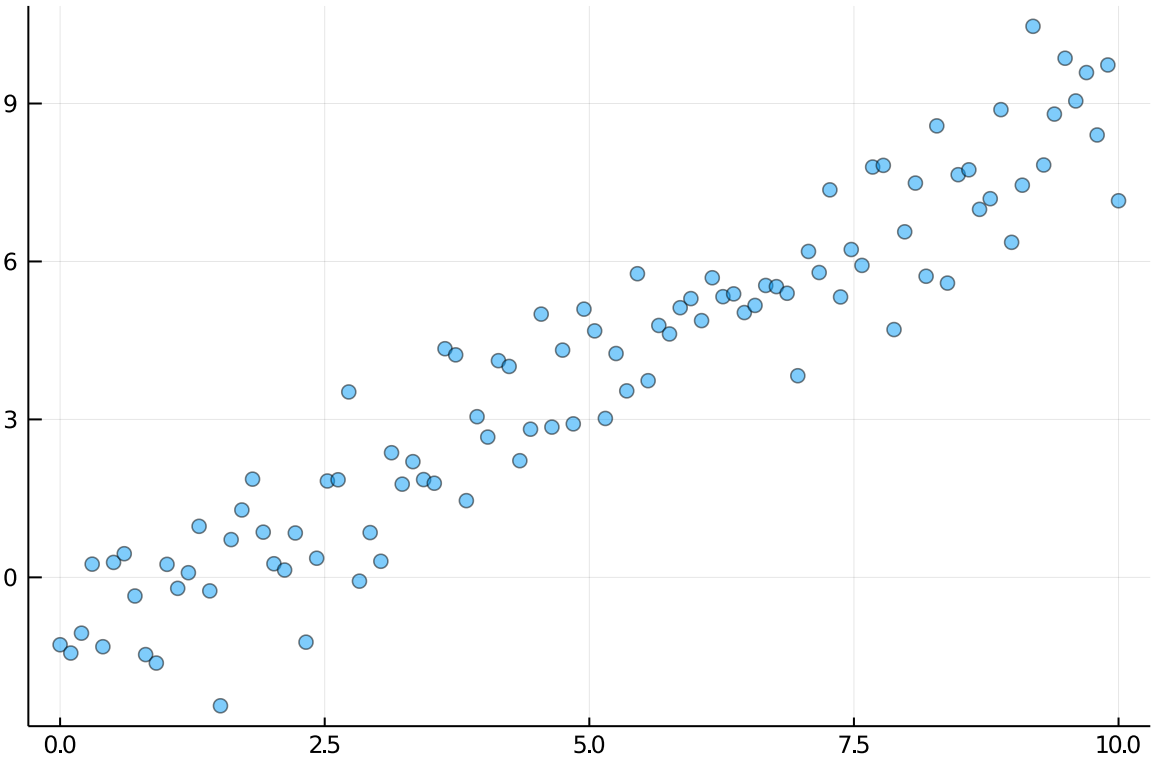
$$A^T A\boldsymbol{\beta} = A^T \mathbf{y}.$$

NUMERICAL CORNER We test our least-squares method on simulated data.

Suppose the truth is a linear function of one variable with Gaussian noise.

In [12]:
```
n, b0, b1 = 100, -1, 1
x = LinRange(0,10,n)
y = b0 .+ b1*x .+ randn(n)
scatter(x,y,legend=false,alpha=0.5)
```

Out[12]:



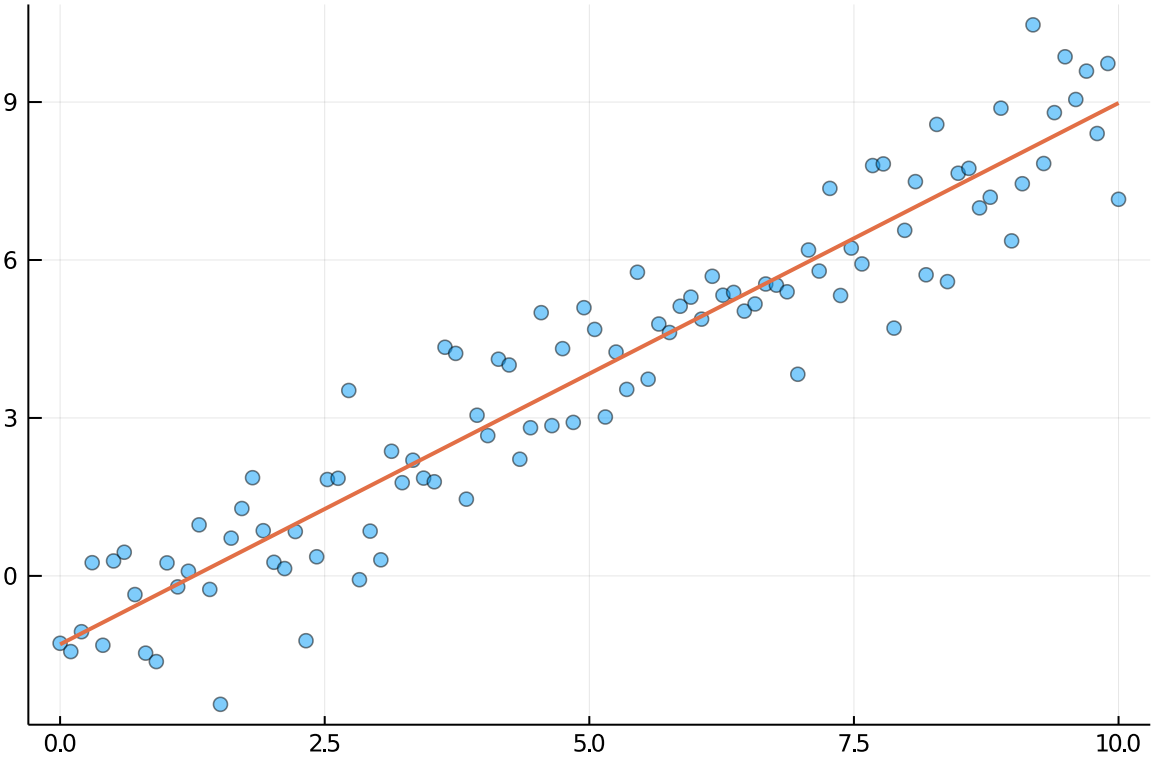
We form the matrix A and use our least-squares code to solve for $\hat{\boldsymbol{\beta}}$.

In [13]:
```
A = hcat(ones(n),x)
coeff = ls_by_chol(A,y)
```

Out[13]:
```
2-element Array{Float64,1}:
 -1.2998954089553174
  1.0281526814735116
```

In [14]:
```
scatter(x,y,legend=false,alpha=0.5)
plot!(x,coeff[1].+coeff[2]*x,lw=2) # recall that array indices start at
 1
```

Out[14]:



3.3.2 Beyond linearity

The linear assumption is not as restrictive as it may first appear. The same approach can be extended straightforwardly to fit polynomials or more complicated combination of functions. For instance, suppose $d = 1$. To fit a second degree polynomial to the data $\{(x_i, y_i)\}_{i=1}^n$, we add a column to the X matrix with the squares of the x_i's. That is, we let

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix}.$$

Then, we are indeed fitting a degree-two polynomial as follows

$$(X\boldsymbol{\beta})_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2.$$

The solution otherwise remains the same.

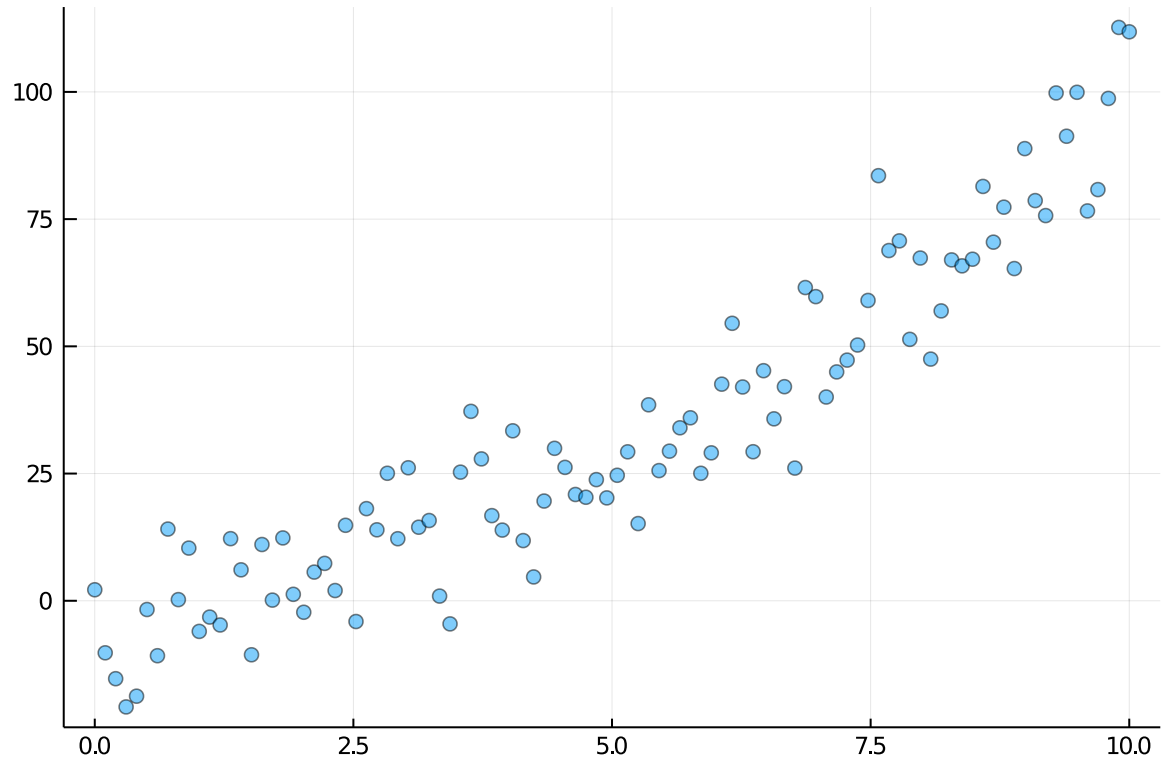
This idea of adding columns can also be used to model interactions between predictors. Suppose $d = 2$. Then we can consider the following X matrix, where the last column combines both predictors into their product,

$$X = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,1}x_{1,2} \\ 1 & x_{2,1} & x_{2,2} & x_{2,1}x_{2,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & x_{n,1}x_{n,2} \end{pmatrix}.$$

NUMERICAL CORNER Suppose the truth is in fact a degree-two polynomial of one variable with Gaussian noise.

In [15]:
```
n, b0, b1, b2 = 100, 0, 0, 1
x = LinRange(0,10,n)
y = b0 .+ b1*x .+ b2*x.^2 .+ 10*randn(n)
scatter(x,y,legend=false,alpha=0.5)
```

Out[15]:



We form the matrix A and use our least-squares code to solve for $\hat{\beta}$.

In [16]:
```
A = reduce(hcat, [ones(n),x,x.^2])
coeff = ls_by_chol(A,y)
```

Out[16]:
```
3-element Array{Float64,1}:
 -3.762829388032236
  2.0963212425816136
  0.8047318543788937
```

In [17]:
```
scatter(x,y,legend=false,alpha=0.5)
plot!(x,coeff[1].+coeff[2]*x.+coeff[3]*x.^2,lw=2)
```

Out[17]: