1 Nullspace

Here we discuss methods to solve an underdetermined set of linear equations $Ax = b$ where the number n of variables is larger than the number m of equations. In other words A is an $m \times n$ matrix with $m < n$.

The set of solutions is then the hyperplane $x_0 + ker(A)$ where $ker(A)$ denotes the nullspace of A and x_0 is any particular solution of $Ax = b$.

We will represent the nullspace of A as the columnspace $colspace(F)$ of a matrix F such that

$$AF = 0.$$

If F satisfies this equation, then the column space of F is a subspace of the null space $ker(A)$. Recall also that the column space of F is the range $Im(F)$ of the linear map defined by the matrix F.

We will generally assume that A *has full rank*, i.e. $rank(A) = m$. The nullspace $ker(A)$ then has dimension $n - m$ and the columns of F span the entire nullspace of A if and only if $rank(F) = n - m$.

If we have found such a matrix F and x_0 with $Ax_0 = b$, then the hyperplane of all solutions to $Ax = b$ can be represented as

$$x_0 + ker(A) = x_0 + Im(F).$$

We want to apply this to the minimization problem

$$\text{minimize } f : \mathbb{R}^n \to \mathbb{R} \text{ under the constraint } Ax = b. \tag{1}$$

Since the solutions x to the constraint are exactly the vectors x of the form $x = x_0 + Fu$, $u \in \mathbb{R}^{n-m}$, we can turn this into the unconstrained problem

$$\text{minimize } g(u) = f(x_0 + Fu) \text{ on all of } \mathbb{R}^{n-m}. \tag{2}$$

Thereby we reduce the dimension of the problem from n to $n-m$. In large scale problems one would not do this since this operation can destroy the sparseness of the Hessian $H(f)$ which is critical for computation in very large dimensional problems.

In dense small problem (up to say $n = 3000$ variables) the above elimination of the constraint $Ax = b$ is a reasonable approach. We will discuss two algorithms to finding F and x_0. In both cases the columns of F will be orthonormal and hence an orthornomal basis for the null space $ker(A)$.

1.1 Null space with QR-decomposition of A

This approach relies on the relation $ker(A) = Im(A')^\perp$, where the prime denotes the matrix transpose and $V^\perp$ is the orthogonal complement of a subspace V. Recall that

$$\begin{aligned} n &= dim(ker(A)) + dim(ker(A)^\perp) = dim(ker(A)) + dim(Im(A')) \\ &= (n - m) + dim(Im(A')). \end{aligned}$$

Thus $dim(Im(A')) = m$. We will find the range $Im(A')$ and its orthogonal complement using the QR-decomposition of the transpose A' (which has dimension $n \times m$, i.e. maps from $\mathbb{R}^m$ to $\mathbb{R}^n$):

$$A' = QR, \tag{3}$$

where Q is orthogonal and R upper triangular with nonzero diagonal, where R is $p \times m$ (i.e. maps from $\mathbb{R}^m \to \mathbb{R}^p$ and Q is $m \times p$, i.e maps from $\mathbb{R}^p \to \mathbb{R}^m$.

In such a factorization Q and R must be $n \times p$ and $p \times m$ respectively, since A' is $n \times m$. Moreover $rank(Q) \geq rank(A') = rank(A) = m$ and so we must have $p \geq m$. Because the columns of Q are linearly independent and of dimension n we also must have $p \leq n$.

Indeed such factorizations exist for all $m \leq p \leq n$ but in practice (i.e. available in libraries) there are only two flavours:

(A) The reduced (minimal) form gives us R as an $m \times m$ matrix and $Q = [q_1, \ldots, q_m]$ as an $n \times m$ matrix with m-columns $q_j \in \mathbb{R}^n$. It follows that R maps onto $\mathbb{R}^m = dom(Q)$ and hence

$$Im(A') = Im(Q) = span(q_1, \ldots, q_m).$$

From this we see that $ker(A) = Im(A')^{\perp} = span(q_1, \ldots, q_m)^{\perp}$ but if we use the reduced form we have to compute this orthogonal complement ourselves. In other words we have to extend $\{ q_1, \ldots, q_m \}$ to an orthonormal basis $\{ q_1, \ldots, q_m, q_{m+1}, \ldots q_n \}$ of $\mathbb{R}^n$ ourselves.
In `R` the minimal form is obtained as follows

```
qrA <- qr(t(A)); Q <- qr.Q(qrA); R <- qr.R(qrA)
```

Here we first compute a QR-decomposition object `qrA` and from this object extract the matrices Q and R using the helper functions `qr.Q` and `qr.R`.

The intermediate forms of the decomposition (3) with $m \leq p \leq n$ extend this matrix Q by adding (arbitrarily) orthonormal columns on the right

$$Q \to Q_+ = [q_1, \ldots, q_m, q_{m+1}, \ldots q_p]$$

and adjusting the matrix R by adding zero rows at the bottom (so that R becomes $p \times m$. This of course does not change the matrix product QR as we can see from block multiplication

$$Q_+ = [Q, Q_1], \ R_+ = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad \Longrightarrow \quad Q_+R_+ = QR + Q1 * 0 = QR.$$

(B) The full (maximal) form of the decomposition has $p = n$, that is, the columns of Q have been extended to a full orthonormal basis of $\mathbb{R}^n$ and so clearly

$$ker(A) = Im(A')^{\perp} = span(q_1, \ldots, q_m)^{\perp} = span(q_{m+1}, \ldots, q_n). \tag{4}$$

In other words the matrix F can be chosen to be the matrix with columns $q_{m+1}, \dots, q_n$:

$$F = [q_{m+1}, \dots, q_n] \in Mat_{n\times(n-m)}(\mathbb{R}). \tag{5}$$

To get a special solution x_0 of $Ax = b$ note that $A = R'Q'$ and solve the factored form $R'Q'x = b$. Set $y = Q'x$. Then *forward solve* the lower triangular system $R'y = b$ and get x from $Q'x = y$ as $x = Qy$.

To get the complete factorization $A' = Q_+R_+$ in R we do

```
qrA <- qr(t(A)); Qp <- qr.Q(qrA,complete=TRUE); R <- qr.R(qrA)
```

and then extract the matrix Q via `Q <- Qp[,1:m]`. R is already in incomplete form (no zero rows at bottom) since we have not specified `complete=TRUE` in the function `qr.R` (the default is `complete=FALSE`).

With this the factorization of A' becomes $A' = QR$ (and not $A' = QpR$) and this is what we need in the computation of the special solution x_0 above. The QR factorization is computed by repeatedly applying (orthogonal) Householder updates and is thus a very stable algorithm.

1.2 Nullspace via SVD decomposition of A

We can also compute the nullspace $ker(A)$ and special solution x_0 of $Ax = b$ using the more involved *SVD-decomposition* of A:

$$A = U\Sigma V'$$

where $U \in Mat_{m\times m}(\mathbb{R})$ and $V \in Mat_{n\times m}(\mathbb{R})$ are orthonormal matrices and Σ is an $m \times n$ diagonal matrix with entries

$$\sigma_1 \geq \sigma_2 \cdots \geq \sigma_m \geq 0$$

(the singular values of A). Since $rank(A) = m$ we must have $\sigma_i > 0$ and so the matrix Σ is invertible. Write $V = [v_1, \dots, v_m]$, where the $v_j \in \mathbb{R}^n$ are the columns of V. Thus, for all $x \in \mathbb{R}^n$,

$$\begin{aligned}
Ax = 0 \quad &\iff \quad U\Sigma V'x = 0 \\
&\iff \quad \Sigma V'x = 0 \\
&\iff \quad 0 = V'x = (v_1 \cdot x, v_2 \cdot x, \dots, v_m \cdot x)' = 0 \\
&\iff \quad x \perp \{\, v_1, \dots, v_m \,\} \\
&\iff \quad x \in span(v_{m+1}, \dots, v_n)
\end{aligned}$$

In other words

$$ker(A) = span(v_{m+1}, \dots, v_n). \tag{6}$$

where $v_{m+1}, \dots, v_n \in \mathbb{R}^n$ are vectors which extend $v_1, \dots, v_m$ to an ON-basis of $\mathbb{R}^n$. To get the full $n \times n$ matrix

$$V_+ = [V, v_{m+1}, \dots, v_n]$$

we must compute the SVD in R via

```
svdA <- svd(A,nv=n); Vp <- svdA$v; V <- V[,1:m]
```

Now a particular solution of $U\Sigma V'x = Ax = b$ can be found by solving $\Sigma V'x = U'b := c$ which is equivalent to

$$V'x = (c_1/\sigma_1, \ldots, c_m/\sigma_m)',$$

A particular solution of this is given by

$$x_0 = Vw, \quad \text{where} \quad w = (c_1/\sigma_1, \ldots, c_m/\sigma_m)' \in \mathbb{R}^m. \tag{7}$$

since the columns of V are orthonormal.