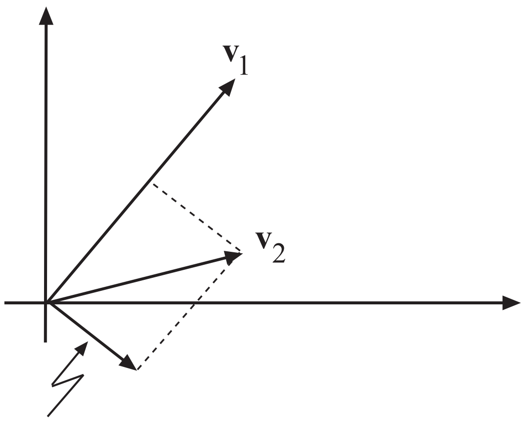
Holger Broman, August 19, 1999

Orthogonalization – Gram-Schmidt, and Projections

The basic problem we start with is the following: given a set of vectors $\{\mathbf{v}_m\}_{m=1}^M$ in $\mathbb{R}^N$, how do we create a set of *orthogonal* vectors that span the same subspace as does $\{\mathbf{v}_m\}_{m=1}^M$?

The first observation we make is that we can assume the vectors $\mathbf{v}_m$ to be linearly independent. If they are not, we just iteratively remove one of the linearly dependent vectors and subtract one from M; the subspace spanned remains the same.

Now, the basic philosophy is extremely simple. Take one vector, say $\mathbf{v}_1$. Then take the part of one vector, say $\mathbf{v}_2$, that is perpendicular to $\mathbf{v}_1$. Then, take the part of $\mathbf{v}_3$ perpendicular to both $\mathbf{v}_1$ and $\mathbf{v}_2$. Continue until finished.



Part of $\mathbf{v}_2$ *perpendicular to* $\mathbf{v}_1$

Gram-Schmidt – which makes the new set of vectors orthonormal, i.e. orthogonal with unity length.

Take $\quad \mathbf{u}_1 = \mathbf{v}_1/||\mathbf{v}_1||$

Take $\quad \mathbf{z}_2 = \mathbf{v}_2 - \left(\mathbf{u}_1^T\mathbf{v}_2\right)\mathbf{u}_1$
$\quad\quad\quad \mathbf{u}_2 = \mathbf{z}_2/||\mathbf{z}_2||$

Take $\quad \mathbf{z}_3 = \mathbf{v}_3 - \left(\mathbf{u}_1^T\mathbf{v}_3\right)\mathbf{u}_1 - \left(\mathbf{u}_2^T\mathbf{v}_3\right)\mathbf{u}_2$
$\quad\quad\quad \mathbf{u}_3 = \mathbf{z}_3/||\mathbf{z}_3||$

Etcetera.

This makes an acceptable but not preferred algorithm for numerical reasons.

Projections – Remember the definition of $P^{\perp}$, a projection matrix that projects on the orthogonal complement of the subspace projected upon by P. Here we go:

Take $\mathbf{u}_1 = \mathbf{v}_1$

Take $\mathbf{u}_2 = P_1^{\perp}\mathbf{v}_2$,
where P_1 projects onto $\mathbf{v}_1$

Take $\mathbf{u}_3 = P_2^{\perp}\mathbf{v}_3$,
where P_2 projects onto the subspace spanned by $\{\mathbf{v}_1, \mathbf{v}_2\}$

Etcetera.

Normalization can be performed afterwards if desired.

This makes for a lousy algorithm, but is a nice packaging for understanding what goes on.

Note 1: If you pick the vectors $\mathbf{v}_m$ in the same order and normalize the "projection-route" set, Gram-Schmidt and Projections will produce the same set of vectors $\{\mathbf{u}_m\}$, apart from a possible numerical difference in accuracy.

Note 2: A closer study of Gram-Schmidt reveals that it actually follows the projection route by subtracting from $\mathbf{v}_m$ the part of $\mathbf{v}_m$ that projects onto $\mathbf{u}_1, \ldots, \mathbf{u}_{m-1}$.

Note 3: You encounter orthogonalizations most often when there is a need to create orthonormal bases, or in the so-called QR factorization problem.

Note 4: The numerically preferred ways to perform orthogonalization will be described in the note on QR factorizations.

If you want a preview of how to implement orthogonalizations, you can run the m-files ortho and householder in Matlab. The theory is not completed yet so maybe you would like to postpone this until you have studied the leaflet on Factorization (QR-factorization).