Gram-Schmidt Process and QR Decomposition

This page is not a typical set of notes from a class lecture, but rather a short crash course on the Gram-Schmidt process and QR decomposition, a very important topic within the field of Matrices and Linear Transformations.

Gram-Schmidt Process

The Gram-Schmidt process starts with a basis $\{\vec{a}_1, \ldots, \vec{a}_n\}$ of the vector space V. The goal is to end up with an orthonormal basis $\{\vec{q}_1, \ldots, \vec{q}_2\}$ for V. The important process to note for us to reach our final orthonormal basis is that given $\vec{x}$ and $\vec{v}$, two vectors of the same dimension:

$$\vec{p} = \text{proj}_{\vec{v}}\vec{x} = \left(\frac{\vec{x}^T\vec{v}}{\vec{v}^T\vec{v}}\right)\vec{v} \qquad \vec{o} = \vec{x} - \vec{p}$$

From here, it follows that $\vec{o}$ is orthogonal to $\vec{v}$. This can be generalized to a set of orthogonal vectors, $\{\vec{v}_1, \ldots, \vec{v}_n\}$. As you can see below:

$$\vec{p} = \sum_{k=1}^{n} \text{proj}_{\vec{v}_k}\vec{x} \qquad \vec{o} = \vec{x} - \vec{p}$$

In this way, it follows that $\vec{o}$ is orthogonal to every member of $\{\vec{v}_1, \ldots, \vec{v}_n\}$. The Gram-Schmidt process simply uses this equation above iteratively. For instance, now that we know $\vec{o}$ is orthogonal to every member of our set, we can define a new set of orthogonal vectors $\{\vec{v}_1, \ldots, \vec{v}_n, \vec{o}\}$. If we have a new vector, say $\vec{y}$, then we simply use the equation above to produce $\vec{o}_y$, and redefine our set as $\{\vec{v}_1, \ldots, \vec{v}_n, \vec{o}, \vec{o}_y\}$. We could keep going on in perpetuity!

The one thing we have to change is that this produces *orthogonal* vectors, not *orthonormal* ones! This can be solved by normalizing each vector when we get it (divide the vector by its magnitude). This will not change the direction of the vectors, so it won't bother with our orthogonalization.

So, let's explicitly state how these nice equations form the Gram-Schmidt process. Well, we start with $\{\vec{a}_1, \ldots, \vec{a}_n\}$. Then getting $\vec{v}_1$ is simple – we just get the unit vector (normalize) $\vec{a}_1$. The next step is to get $\vec{v}_2$. We simply normalize the vector $\vec{a}_2 - \text{proj}_{\vec{v}_1}\vec{a}_2$ to get $\vec{v}_2$. Then $\vec{v}_3$ is the normalization of $\vec{a}_3 - \text{proj}_{\vec{v}_1}\vec{a}_2 - \text{proj}_{\vec{v}_2}\vec{a}_2$. And this continues onwards! The "nice" recursive formula to do all of this is:

$$\vec{w}_i = \vec{a}_i - \sum_{k=0}^{i-1} \text{proj}_{\vec{v}_k}\vec{a}_i \qquad \vec{v}_i = \frac{\vec{w}_i}{|\vec{w}_i|}$$

Note that while this could all be written in one equation, this is much more succinct and clear. The equation relating $\vec{w}_i$ and $\vec{v}_i$ is simply a normalization of $\vec{w}_i$. Also note that for an orthonormal vector $\vec{v}$, it follows that $\vec{v}^T\vec{v} = 1$. Thus, we can simplify the formula above for our orthogonal vectors as $\text{proj}_{\vec{v}_k}\vec{a}_i = (\vec{v}_k^T\vec{a}_i)\vec{v}_k$, making the computation a bit easier!

QR Decomposition

QR Decomposition builds off of the Gram-Schmidt process, applying it to matrices! We start with an $m \times n$ matrix A with linearly independent columns. This is equivalent to saying A has full column rank, which implies $m \geq n$. Now, we start by splitting the columns of A into the set $\{a_1, \ldots, a_n\}$. As these columns are linearly independent, they form the basis for a subspace. Sound familiar? We can just take the Gram-Schmidt of this set to produce the new set $\{\vec{v}_1, \ldots, \vec{v}_n\}$. Now, let V be an $m \times n$ matrix such that $\vec{v}_i$ is the i-th column of V. Now, we set up the equation $A = VR$, where R is accordingly an $n \times n$ matrix. Solving out the matrix equation, we find that $R_{ij} = \vec{v}_i^T\vec{a}_j$. Furthermore, due to the vectors of our set being orthogonal, we find that for all $i > j$, $R_{ij} = 0$, so R is upper triangular!