# Linear Algebra – Week 4-5

Matrices as objects that map one vector into another

Matrix multiplication using Einstein Summation method.

A person standing in front of a blackboard

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Some resemblance between unit vector and the dot product.

A blackboard sign next to it

Description automatically generated

A blackboard sign next to it

Description automatically generated

And that's why we talk about a matrix multiplication with a vector as being the projection of that vector onto the vectors composing the matrix, the columns of the matrix.

Change of basis vectors:

A close up of a blackboard

Description automatically generated

So that's how you do the reverse process. If you want to take Bear's vector into my world, you need Bear's basis in my coordinate frame, and if you want to do the reverse process, you want my basis in Bear's co-ordinate frame.

Now, this was all prep really for the fun part, which is, we said before in the vectors module that we could do this just by using projections, if the new basis vectors were orthogonal, which these are. So let's see how this works with projections.

A close up of a blackboard

Description automatically generated

So I've used projections here to translate my vector to Bear's vector just using the dot product. Now remember, with the vector product, what I'd have to do is I'd have to remember to normalize when I did the multiplication by Bear's vectors, I'd have to normalize by their lengths. But in this case, their lengths are all one. So it's actually really easy. So we don't have to do the complicated matrix maths, we can just use the dot product if Bear's vectors are orthogonal. Now, there is one last thing.

A close up of a blackboard

Description automatically generated

Since, we don’t know the rotation in the new basis vector space. We convert the vector to our basis vector and rotate it and then convert back to the new basis vector representation. We thought about the coordinate systems and how to do transformations in non-orthogonal coordinate systems. It's been quite hard work, but this really sets us up for example in principal component analysis, to operate in different basis systems.

Orthonormal Basis Vectors:

So what I found is that A\_t times A gives me the identity matrix, and what that means is, is that A\_T is a valid inverse of A. Which is really kind of neat. A set of unit length basis vectors that are all perpendicular to each other are called an **orthonormal basis set,** and the **matrix composed of them is called an orthogonal matrix**

Now in data science what we're really saying here is that wherever possible, we want to use an orthonormal basis vector set when we transform our data. That is, we want our transformation matrix to be an orthogonal matrix. That means the inverse is easy to compute. It means the transmission is reversible because it doesn't collapse space. It means that the projection is just the dot product. Lots of things are nice and pleasant, and easy. If I arrange the basis vectors in right order, then the determinant is one, and that's an easy way to check and if they aren't just exchange a pair of them and actually then they will be determinant one rather than the minus one. So what we've done in this video is look at the transpose and that's led us to find out about the most convenient basis factor set of all which is the orthonormal basis factor set which together make the orthogonal matrix whose inverse is its transpose.

**The Gram-Schmidt Process:**

How to construct an orthonormal basis vector set?

We've said several times now that life is much easier if we can construct an orthonormal basis vector set, but we haven't talked about how to do it. So in this video we'll do that starting from the assumption that we already have some linearly independent vectors that span the space we're interested in.

If you want to check linear independence you can write down their columns in a matrix and check the determinant isn't zero. If they were linearly dependent, that would give you a zero determinant. But they aren't orthogonal to each other or of unit length.

A picture containing person, man, outdoor

Description automatically generated

The Gram-Schmidt process is a method for constructing an orthonormal basis of a space that a set of given vectors span. It can also be used to determine the dimension of that space, which may be different than the dimension of the vectors themselves or the number of vectors provided to span the space.

So we're just going to normalize her and we're going to say that my eventual first basis vector e, is going to be equal to v1, just normalized to be of unit length. Just divided by that, its length. So, e is just going to be some normalized version of v1. And I can now think of v2 as being composed of two things. One is a component, let's do this in orange, a component that's in the direction of e1 like that, plus a component that's perpendicular to e1. But the component that's in the direction of e1, I can find by taking the vector projection v2 onto e1. So I can say v2 is equal to the vector projection of v2 onto e1 dotted together. And if I want to get that actually as a vector, I'll have to take e1, which is of unit length so I'd have to divide by the length of e1 but the length of e1 is one. So, forget it. And if I take that off of v2, then I'll have this guy, and let's call him u2. So I can then say that u2, so plus u2, so I can then rearrange this and say that u2 is then equal to v2 minus this projection v2.e1 times e1. And if I normalize u2, if I take u2 divided by its length, then I'll have a unit vector which is going to be normal to v1. So if I take a normalized version of that, let's say it's that, that will be e2. And that will be at 90 degrees to e1. So it'll actually be there, e2, once I've moved it over. And that will be another unit length vector normal to e1. So that's the first part of taking an orthonormal basis. Now my third vector v3 isn't a linear combination of v1 and v2, so v3 isn't in the plane defined by v1 and v2. So it's not in the plane of e1 and e2 either. So I can project v3 down, let's say something like that, onto the plane of e2 and e1, and that projection will be some vector in the plane composed of e2s and e1s. So I can then write down that v3 minus v3 dotted with e1, e1's. That's going to be the component v3 that's made up of e1's, minus v3 dotted with e2, e2's. That's the component of v3 that's made up of e2's. And then all that's going to be left is going to be this perpendicular guy there, so that's going to be a perpendicular vector which we'll call u3, which is perpendicular to the plane.