MATH 728D: Machine Learning Lab #17: Projection

John Burkardt

January 29, 2019

The fastest way back to the shore is the perpendicular direction!

We are familiar with the problem of finding the "least squares line" that best approximates a set of data values (x_i, y_i) that doesn't actually fall perfectly on a line. When we do this, we are solving a version of the projection problem, which chooses an approximation to data values that minimizes a measure of the error. In this lab, we will look at some examples of this idea, ranging from the least squares line to the idea of principal component analysis.

1 Projection of a vector onto another vector

Given a pair of vectors u and v, the dot product formula relates the inner product to the lengths of the vectors and the angle θ between them:

$$< u, v >= ||u||_2 \, ||v||_2 \, \cos(\theta)$$

In the following, assume we have:

$$u = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad v = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

Using the vector u as a guide, we want to verify the dot product relation, compute the angle in degrees between the vectors.

The *length* of the projection of v onto u is $\frac{<u,v>}{||u||_2}$. The *direction* of the projection of v onto v is $\frac{u}{||u||_2}$. Therefore, the projection of v onto u is

$$v\_proj = \frac{< u, v >}{||u||_2} \frac{u}{||u||_2} = \frac{< u, v >}{< u, u >} u$$

Thus, we can decompose the vector v into parallel and perpendicular components v_proj and v_perp. These components define a right triangle, so we can also verify the Pythagorean relationship.

Exercise 1:

- calculate `u_dot_v` $= < u, v >$
- calculate `u_norm` $= ||u||_2$
- calculate `v_norm` $= ||v||_2$
- calculate `cos_theta` $= \cos(\theta)$;
- Verify that your data satisfies the dot product formula ;
- calculate `theta_degrees` $= \frac{180}{\pi} \cos^{-1}(\cos(\theta))$;
- Compute `v_proj` $= \frac{<u,v>}{<u,u>} u$;
- Compute `v_perp` $= v - \frac{<u,v>}{<u,u>} u$;
- Verify the Pythagorean relation: $||v||^2 = ||v_{proj}||^2 + ||v_{perp}||^2$

2 Householder Transformations

In order to do more general projection problems, we will need to use the QR factorization, which rewrites an $m \times n$ matrix $A = Q * R$, where Q is $m \times m$ orthogonal and R is $m \times n$ upper triangular. This factorization can be built up by a sequence Q_j of simple orthogonal matrices known as *Householder transformations*, which operate on each column of A, gradually transforming it to upper triangular form:

$$Q_1 * Q_2 * ... * Q_k * A = R$$

If we then multiply both sides of this equation by $Q = Q_k' * ... * Q_2' * Q_1'$, we have our QR factorization.

In this exercise, we construct and apply Householder transformations to compute the QR factorization.

We are going to operate on A one column at a time. In order to handle column j of matrix A, we construct the vector v as follows:

```
v        = A(:,j);                    % copy column j of current version of A
v(1:j-1) = 0;                         % zero out entries 1 to j - 1
v(j) = v(j) + sign(v(j)) * norm(v);   % modify entry j
```

Now we use v to define the corresponding Householder transformation:

$$Q_j = I - \frac{2}{v'v} v\, v'$$

and we compute:

$$A = Q_j * A$$

We carry out this process, for $j = 1, 2, ..., n-1$, when the matrix A should have become upper triangular.

In the following, assume we have the matrix A:

$$A = \begin{pmatrix} 2 & 4 & 4 \\ 2 & -2 & 5 \\ 1 & 7 & 6 \end{pmatrix}$$

Exercise 2:

1. Compute Q_1, the Householder transformation that applies to column 1 of A;
2. Compute $A_1 = Q_1 * A$ and verify that column 1 is now upper triangular;
3. Compute Q_2, the Householder transformation that applies to column 2 of A_1;
4. Compute $A_2 = Q_2 * A_1$ and verify that columns 1 and 2 are now upper triangular;
5. Since A_2 is now actually an upper triangular matrix, define $R = A_2$, $Q = Q_1' * Q_2'$;
6. Verify that $A = Q * R$;

3 Orthonormal basis by QR Method

Suppose we have several vectors v_1, v_2, ..., v_k that are in a linear space $\mathcal{T}$. The set of all linear combinations of the v vectors, also called the *span* of the vectors, forms a linear subspace $\mathcal{S} \subset \mathcal{T}$. The best way to describe the subspace $\mathcal{S}$ is to determine a set of basis vectors, with typical element u, and the best kind of basis vectors are linearly independent, with unit length and pairwise orthogonal. Although we start with k vectors v, the basis set u can be any size from 0 to k.

A natural way to determine the basis vectors u begins by packing the vectors v into a matrix:

$$V = [v_1|v_2|...|v_k]$$

and then computing the QR factorization:

$$V = Q * R$$

R will have the same shape as V, but will be upper triangular. If R(1,1) is nonzero, the Q(:,1) is a useful basis vector, and so on for successive diagonal entries of R, until we run out of diagonal entries, or hit a zero diagonal entry. Because of numerical roundoff, we will actually stop when the entries of R become "very small. The initial columns of Q that we select are the basis for the subspace $\mathcal{S}$.

In the following, assume we have:

$$v_1 = \begin{pmatrix} 3 \\ 0 \\ 4 \end{pmatrix} \quad v_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \quad v_3 = \begin{pmatrix} 4 \\ -2 \\ 7 \end{pmatrix}$$

Exercise 3:

- Form the matrix V from the vectors v_1, v_2, v_3;
- Use the command `[Q,R] = qr(V)` to compute the QR factorization of V;
- Set `dim = 0`;
- Set a tolerance `tol = sqrt(eps)`;
- For each $1 <= j <= 3$, if `tol <= |R(J,J)|`, increment `dim = dim + 1`;
- *You should find that* `dim=2`*, that is, only two columns of Q are needed*;
- Define `U = Q(:,1:dim)`; this is your basis for $\mathcal{S}$;
- Verify that $U'U = I$, that is, the columns of U are orthonormal;

4 Projection of a vector into a subspace

Let us continue the previous exercise. Suppose, then, that we have a vector $v \in \mathcal{T}$, and we wish to determine its projection in the subspace $\mathcal{S}$, for which we have computed the orthonormal basis matrix U.

Our goal is to decompose w as

$$w = w\_proj + w\_perp$$

where $w\_proj \in \mathcal{S}$ and w_perp is perpendicular (has a zero dot product) with every vector in $\mathcal{S}$.

Because U is a basis, every vector $v \in \mathcal{S}$ must be able to be written as

$$v = U * \alpha$$

where α is a set of coefficients of the columns of U.
Because $U' * U = I$, we can determine the α coefficients for any such v by:

$$\begin{aligned} v =& U * \alpha \\ U' * v =& U' * U * \alpha = \alpha \end{aligned}$$

Now if w is not actually in the subspace $\mathcal{S}$, we can still find the coefficients of the projection of w into $\mathcal{S}$ in the same way:

$$\alpha = U' * w$$

and once we have the coefficients, we can construct the projection

$$w_{proj} = U * \alpha = U * U' * w$$

In the following, assume we have the vectors v_1, v_2, v_3, and the U matrix computed from the previous exercise, and let the vector w be defined by:

$$w = \begin{pmatrix} 3 \\ 4 \\ 6 \end{pmatrix}$$

Exercise 4:

- Compute the coefficients α of the projection of w into the space spanned by the v vectors, $\alpha = U' * w$;
- Compute $w_{proj} = U * \alpha$;
- Compute $w_{perp} = w - w_{proj}$;
- Verify $||w||^2 = ||w_{proj}||^2 + ||w_{perp}||^2$;
- Verify $< w_{perp}, v_1 >=< w_{perp}, v_2 >=< w_{perp}, v_3 >= 0$;