Query projection and cosine similarity

- When computing relevance between a query, $\mathbf{q}$, and some document, $\mathbf{d}$, we first must project $\mathbf{q}$ onto the K-dimensional manifold that we computed from the SVD of $\mathbf{X}$:

$$\hat{\mathbf{q}} = \mathbf{q}\mathbf{V}^*\boldsymbol{\Sigma}^{*-1} \quad \in \mathbb{R}^{1\times K}$$

- Document retrieval then amounts to measuring a distance between our projected query, $\hat{\mathbf{q}}$, and document, $\hat{\mathbf{d}}$, in our projected K-dimensional space. In LSA, it is common to use cosine similarity:

$$\text{cosine}(\hat{\mathbf{q}}, \hat{\mathbf{d}}) = \frac{\hat{\mathbf{q}} \cdot \hat{\mathbf{d}}}{\|\hat{\mathbf{q}}\| \|\hat{\mathbf{d}}\|} = \frac{\sum_{i=1}^{K} \hat{q}_i \hat{d}_i}{\sqrt{\sum_{i=1}^{K} \hat{q}_i^2} \sqrt{\sum_{i=1}^{K} \hat{d}_i^2}}$$

- In practice, the above can be computed efficiently using matrix-vector operations between $\hat{\mathbf{u}}$ and a projected corpus of documents $\hat{\mathbf{D}}$.

Non-negative matrix factorization (NMF)

- Directly applying SVD to the term-document matrix enables us to compute distances between documents (and by extension queries) in a latent space, but it comes with a few undesirable properties
    - The right singular vectors (which do the projecting) have the ability to apply negative coefficients. Is this useful? Could we express the same thing using positive coefficients applied to antonyms? For example: one might argue that these two bases convey the same meaning:

$$\mathbf{u} = (0.5)swim + (0.7)sun + (-1.0)rain + (-0.2)dark$$

$$\mathbf{u} = (0.5)swim + (1.4)sun + (0.01)rain + (0.01)dark$$

    - There is no sparsity constraint on $\hat{\mathbf{U}}$, which makes each dimension of the resultant latent feature less interpretable w.r.t. the original feature dimensions.

- Non-Negative Matrix Factorization addresses these issues by imposing positivity and sparsity constraints on the matrix factors.