Problem Statement for Images

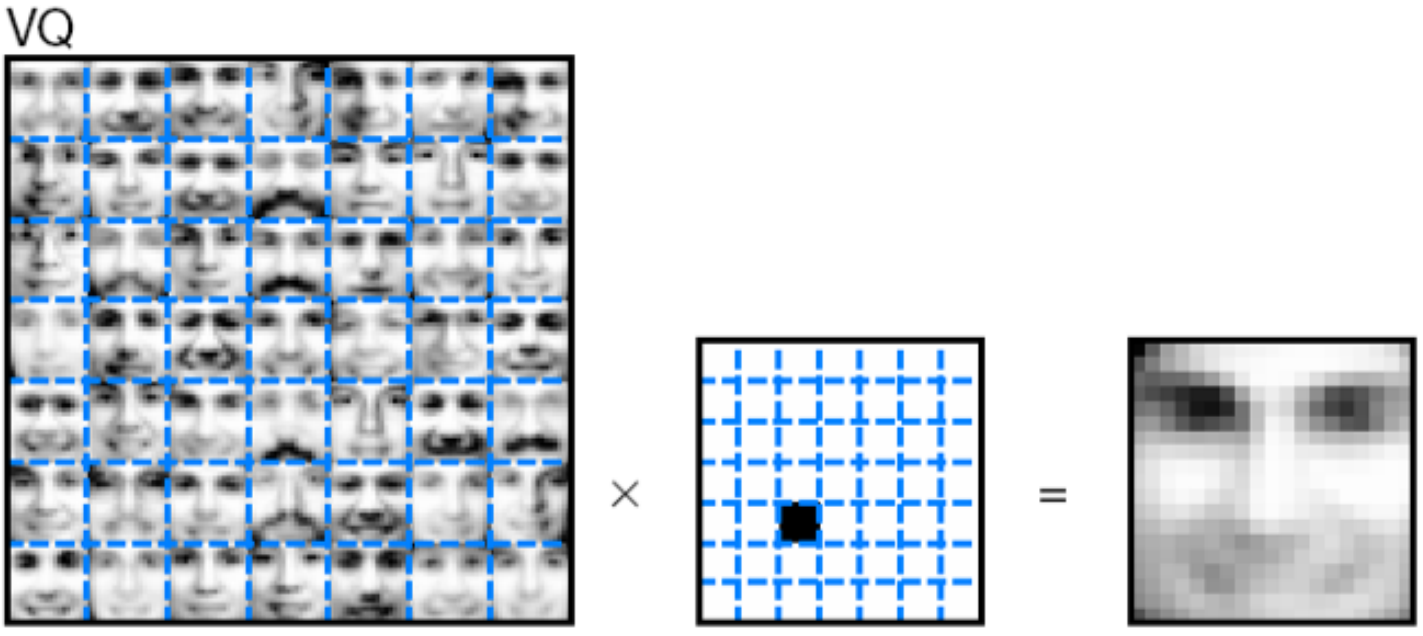
- Given a set of images we want to be able to do the following
- Create a set of basis images from which we can create new images by linear combination
- Find weights that produce any input image from the basis image
  - One set of weights for each input image

slide content adapted from materials from Marshall Tappen

Some possible solutions

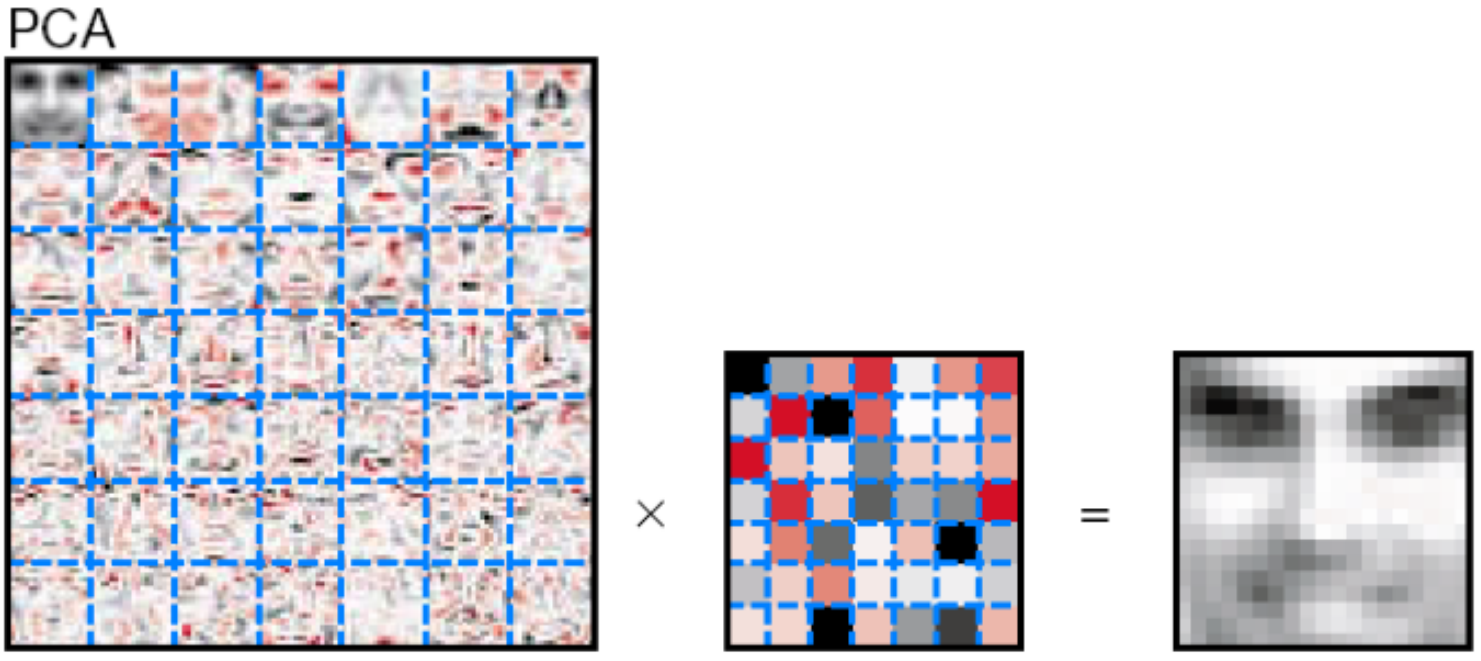
- Vector Quantization
- Principal Components Analysis
- Non-negative Matrix Factorization

Vector Quantization



- Find closest match
- Similar to nearest-neighbor classification
- *Limitations*: scales with number of basis images, provides no analysis

Principal Components Analysis

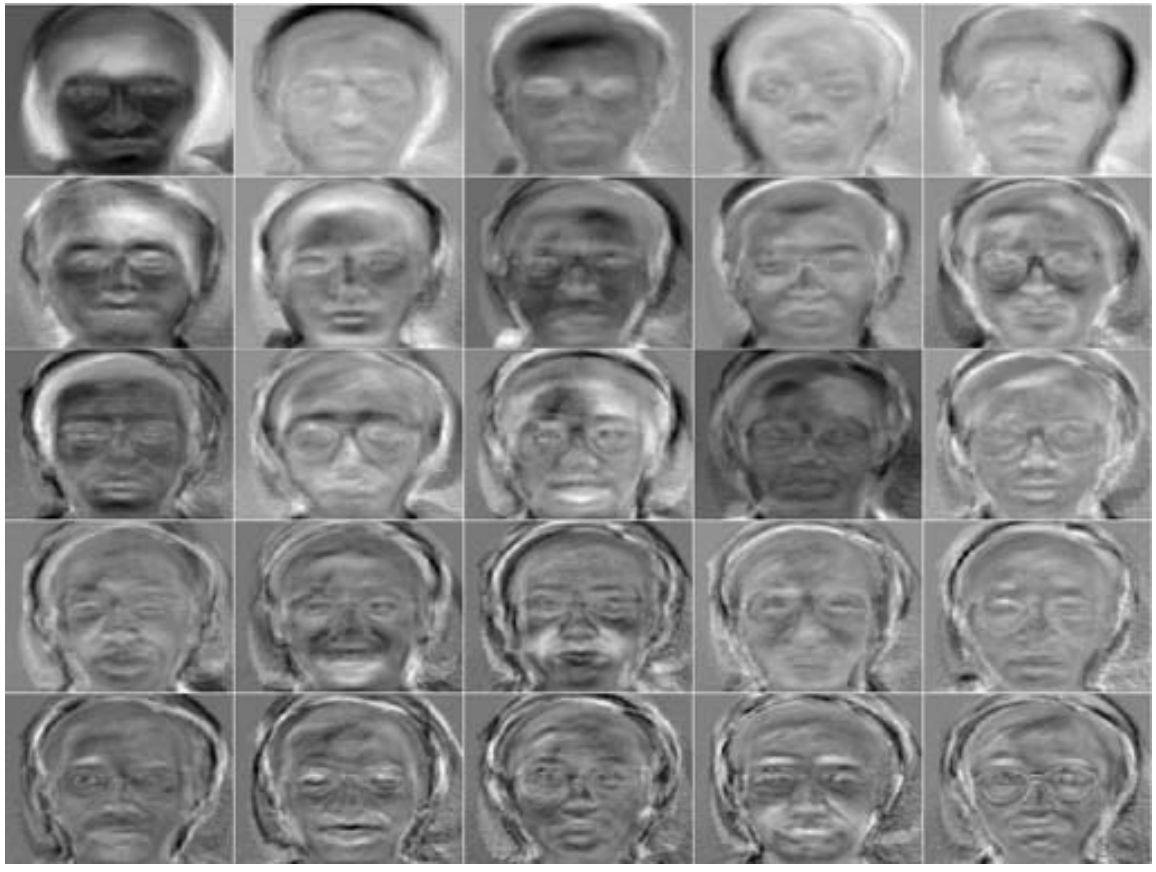


- Basis images are, by construction, orthogonal
- Reconstruction by linear combination

Aspects of PCA

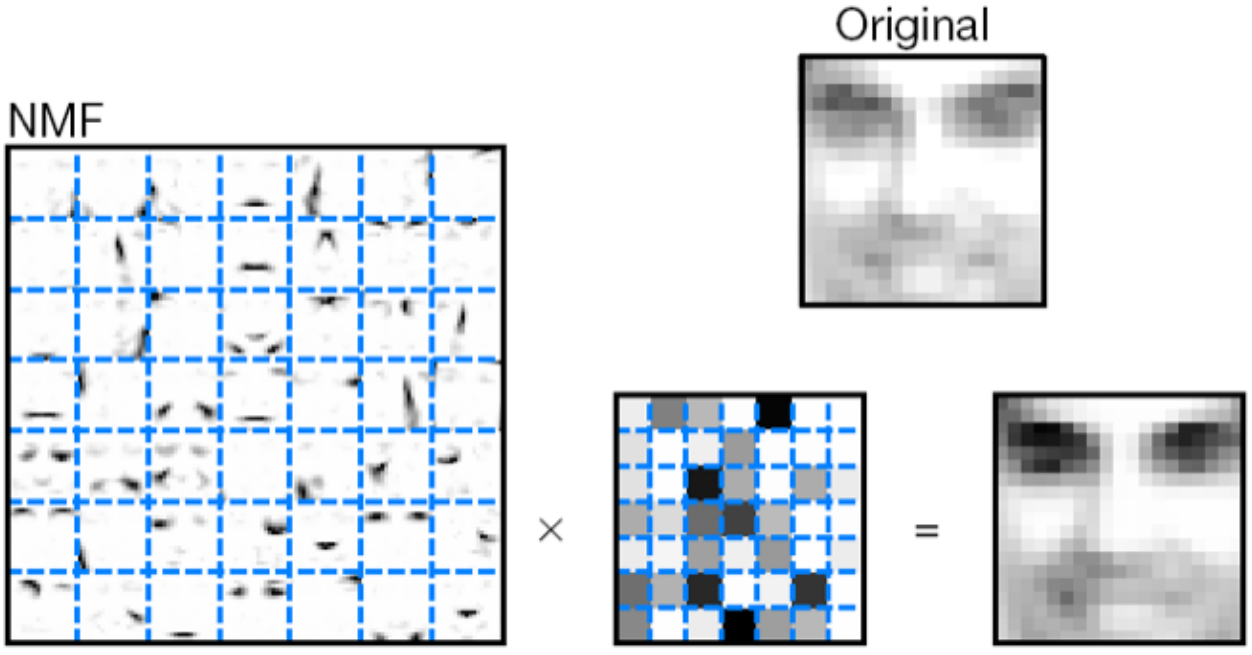
- PCA involves arbitrary linear combinations, we add some and se subtract some
- The basis images obtained do not necessarily correspond with our intuition
- In some contexts subtraction is not a sensible operation
- How is a face subtracted, what is subtraction in the context of document classification.

Eigenfaces Example



- Probably not how faces are represented in the brain

Non-negative matrix factorization

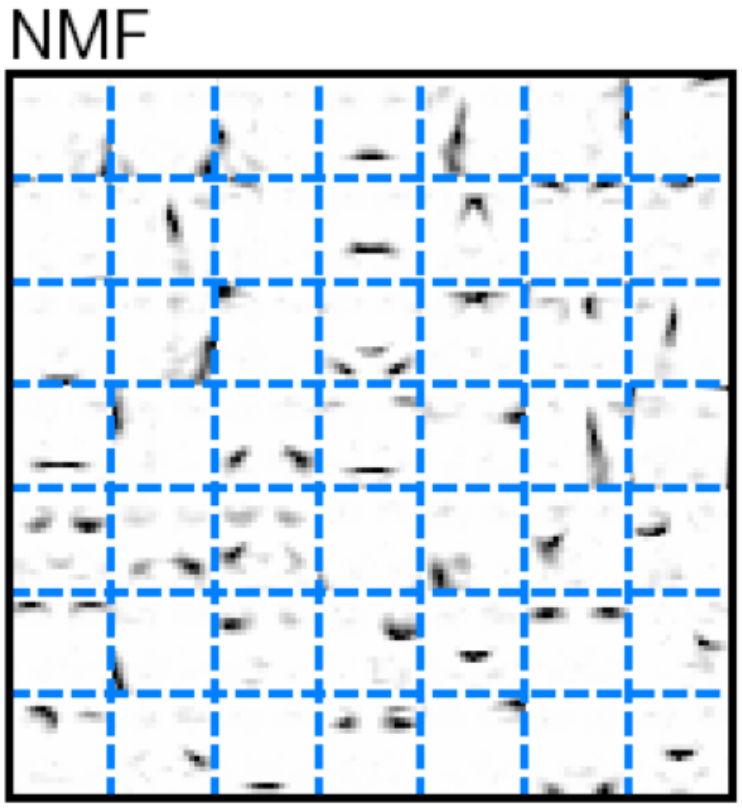


- Similar to PCA, but the coefficients cannot be non-negative

NMF Basis Image Properties

- Only allowing addition makes more intuitive sense in certain contexts and has some correspondence with how neurons operate
- Constraining the reconstruction coefficients to be positive often leads to nice basis images
  - Basis images represent different *parts* of the objects being studied

Non-negative matrix factorization

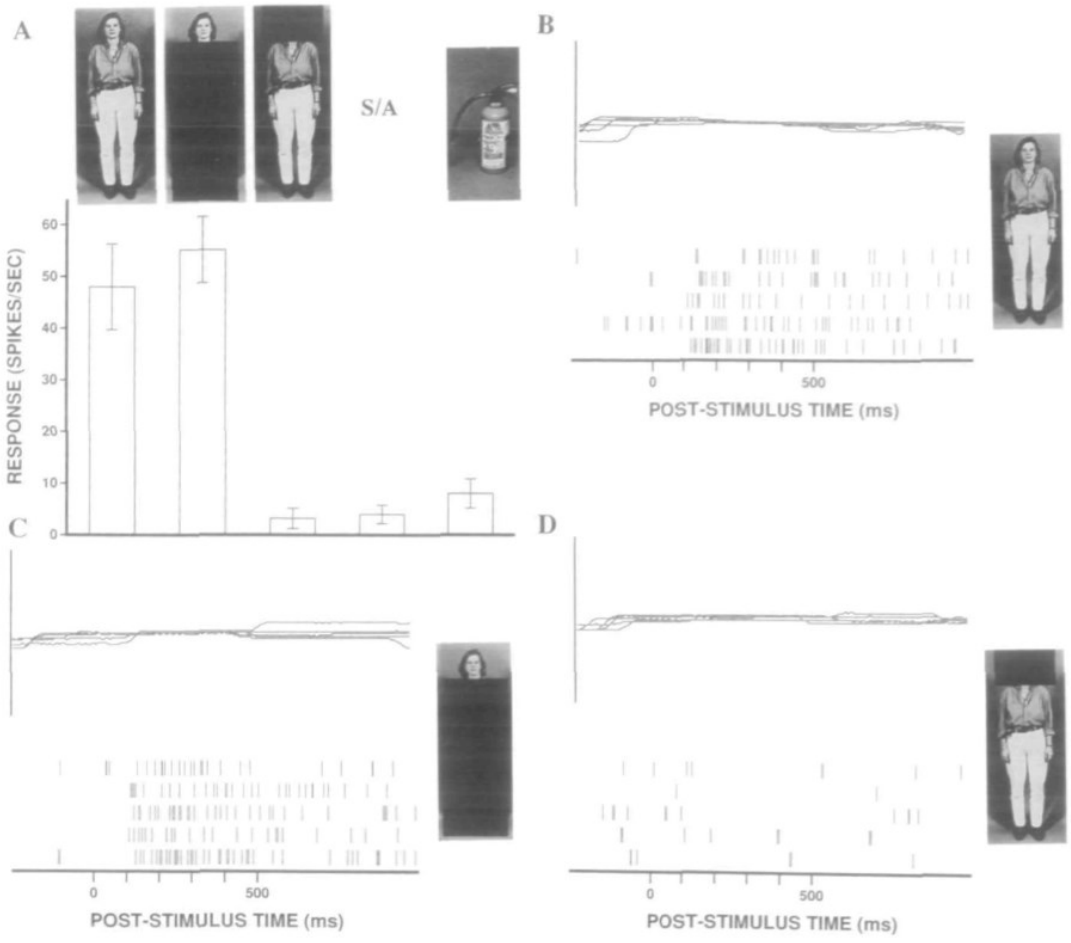


- The factorization has naturally found what we might think of as parts of the faces

Comparison of PCA and NMF

- PCA
  - Produces an optimal set of basis images
  - But this optimality might not be useful for your application
- NMF
  - Produces coefficients with a constraint
  - Can naturally produce a nicer basis, not constrained to be orthogonal

Evidence from Neuroscience



- Different visual stimuli presented to macaque monkey

Wachsmuth et al 1994

- Of 53 neurons,
- 32 percent responded to head only
- 9 percent responded to body only
- 41 percent responded to both the head and the body in isolation
- 17 percent responded to the whole body only
- Suggestive a parts based encoding of figures in the brain

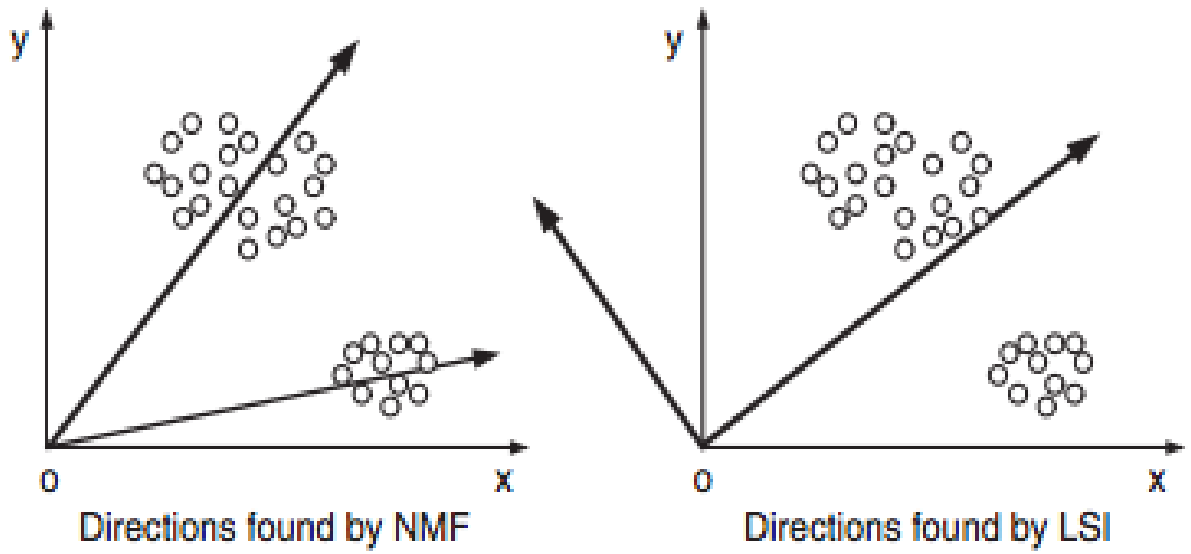
Non-negative Matrix Factorization

- Non-negative Matrix Factorization provides a similar decomposition of the dt matrix
- Naturally additive model since the values are all constrained to be non-negative

$$\arg\min_{W,H} ||X - WH||^2 = \sum_{i,j} X_{ij} - WH_{ij}$$

Document Clustering

- *Document Clustering*: Partitioning a corpus into a predefined number of clusters related to a coherent topic
- NMF applied to text analysis by Xu, Liu, Gong "Document Clustering Base on Non-negative Matrix Factorization"



NMF Document Clustering Algorithm

- *Document Clustering Algorithm*:
  - Construct the term-document matrix $\mathbf{X}$ from the given corpus
  - Find an NMF decomposition of $\mathbf{X}$
  - Normalize the factors U and V
  - Examine each column of V and look for the component with the largest value and assign the corresponding document to cluster k
- Standard Datasets for Document Clustering: NIST Topic Detection and Tracking (TDT2), Reuters dataset

Performing Non-Negative Matrix Factorization in Python

```
import numpy as np
from sklearn.decomposition import NMF
<...>
X = <...>
<...>
model = NMF(n_components=2, init='random', random_state=0)
W = model(fit(X))
H = model.components_
```

- Demonstration of the singular value decomposition routine in the numpy linear algebra package
- This can be used to experiment with LSA on a small document corpus

Task

- Write a script to perform NMF decomposition on a few simple matrices
- Characterize the error between the reconstruction and the original data as you change the rank
- Create a plot of error vs. rank