Unsupervised Learning

In *supervised learning*

- we are given training examples $\langle \mathbf{X}, \mathbf{y} \rangle$
- we find a relationship between features $\mathbf{X}$ and targets/labels $\mathbf{y}$

In *unsupervised learning*

- we are given training examples $\mathbf{X}$
- we find a relationship among the *features* $\mathbf{X}$
    - $\mathbf{y}$ is not usually given (although we may refer to it for informational purposes)

By finding groups of related features we may be able to demonstrate

- Dimensionality reduction: reducing from n original features to $n' \leq n$ *synthetic features*
- Clustering of similar examples
- Cleaning up noisy data

Dimensionality reduction

Why is the relationship among features interesting ?

- Features may be interdependent (redundant)
- Consider the MNIST digits
    - Pairs of features in the 4 corners are highly correlated (e.g., mostly same color)
    - Pairs of features in a vertical line (associated witht the digit "1") are somewhat correlated
        - due to their coocurrence in 10% of the examples corresponding to "1"

Because of the high pair-wise correlation, there may be a *more compact* way of representing the information

- A new "synthetic" feature representing the presence of a "concept"
    - "Rectangle of same pixels"
    - "Vertcal line of pixels"

Let's illustrate with a reduced dimension representation of the MNIST digits

- Original feature vector length $n = 784$
- Reduced feature vector length $n' = 150$

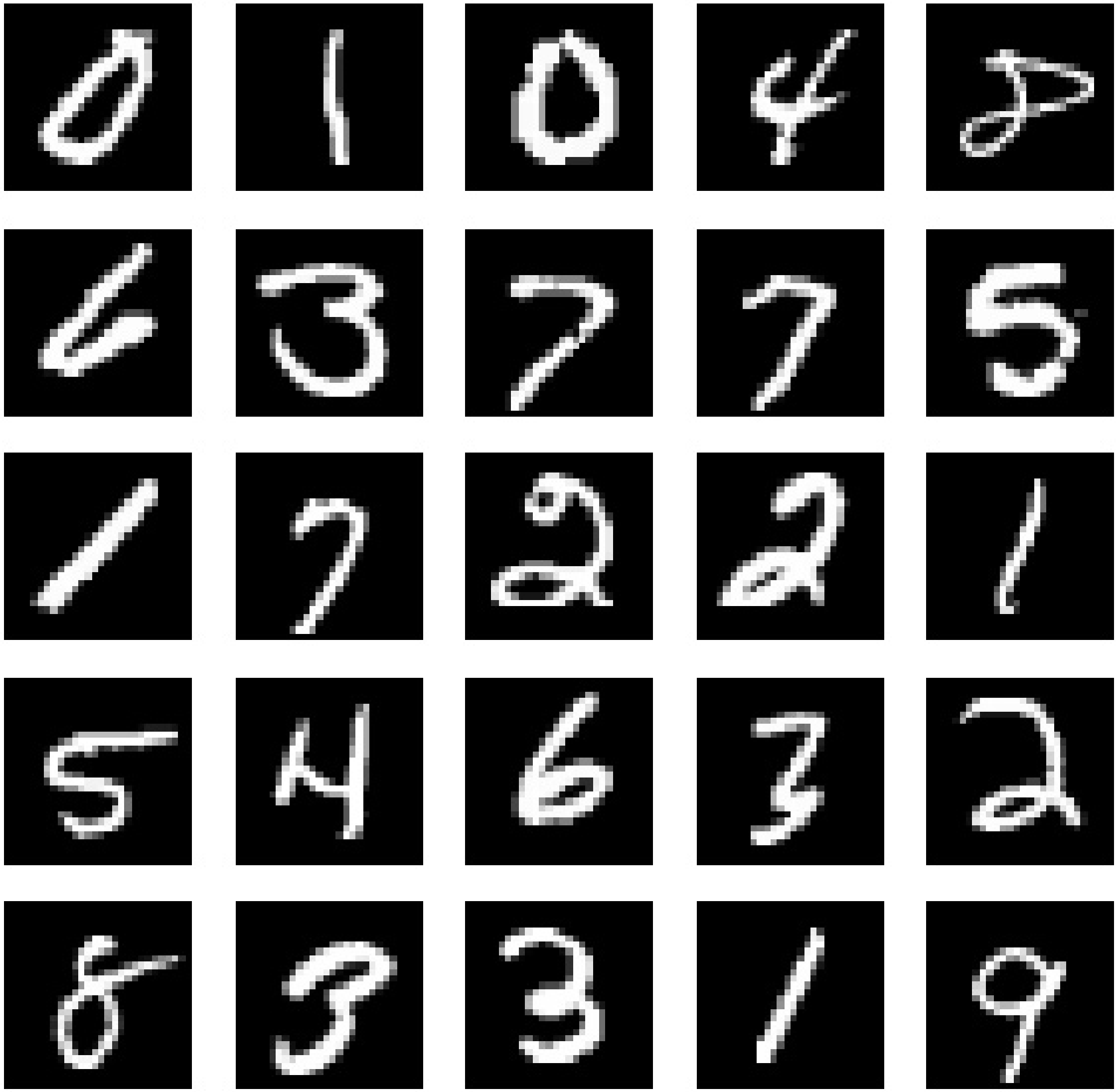
We

- Take the 784 *original* features
- Create 150 *synthetic* features

To demonstrate how little information is lost

- We map *backwards* from the reduced n' dimension representation back to the n dimension represnetation

PCA: reconstructed MNIST digits (95% variance)



The reconstructed $n = 784$ feature digits are a little blurry but still recognizable.

So 80% of the original $n = 784$ features convey little information.

Dimension reduction: examples

Color 3D movie to Black/white still image

- Lose Depth
- Lose color of eyes/hair/clothing
- Lose motion
    - but pose may be informative

For the purpose of recognizing a person, little information is lost

Equity time series

Consider examples with $n = 500$ features

- $\mathbf{x}_j^{(\mathbf{i})}$ is the daily return of stock number j on day i
- Feature $\mathbf{x}_j = [\mathbf{x}_j^{(\mathbf{i})} | 1 \leq i \leq m]$ (returns of equity j)
- Highly correlated with most other features $\mathbf{x}_{j'}$

One way to interpret the high mutual correlation among equity returns

- There is a *common influence* affecting all equities
- e.g., An equity index reflecting the broad market
- Pair-wise correlation of features arises through influence of the shared index

$$
\begin{aligned}
\mathbf{x}_1 &= \beta_1 * \tilde{\mathbf{x}}_{\text{index}} + \epsilon_1 \\
\mathbf{x}_2 &= \beta_2 * \tilde{\mathbf{x}}_{\text{index}} + \epsilon_2 \\
&\vdots \\
\mathbf{x}_{500} &= \beta_{500} * \tilde{\mathbf{x}}_{\text{index}} + \epsilon_{500}
\end{aligned}
$$

If each ϵ_j is small (i.e., $\tilde{\mathbf{x}}_{\text{index}}$ is a close approximation of $\mathbf{x}_j$)

- Then a single feature $\mathbf{x}_{\text{index}}$
- Is an effection way of summarizing $\mathbf{x}$, which has 500 features

Clustering

Are the m examples in the training set

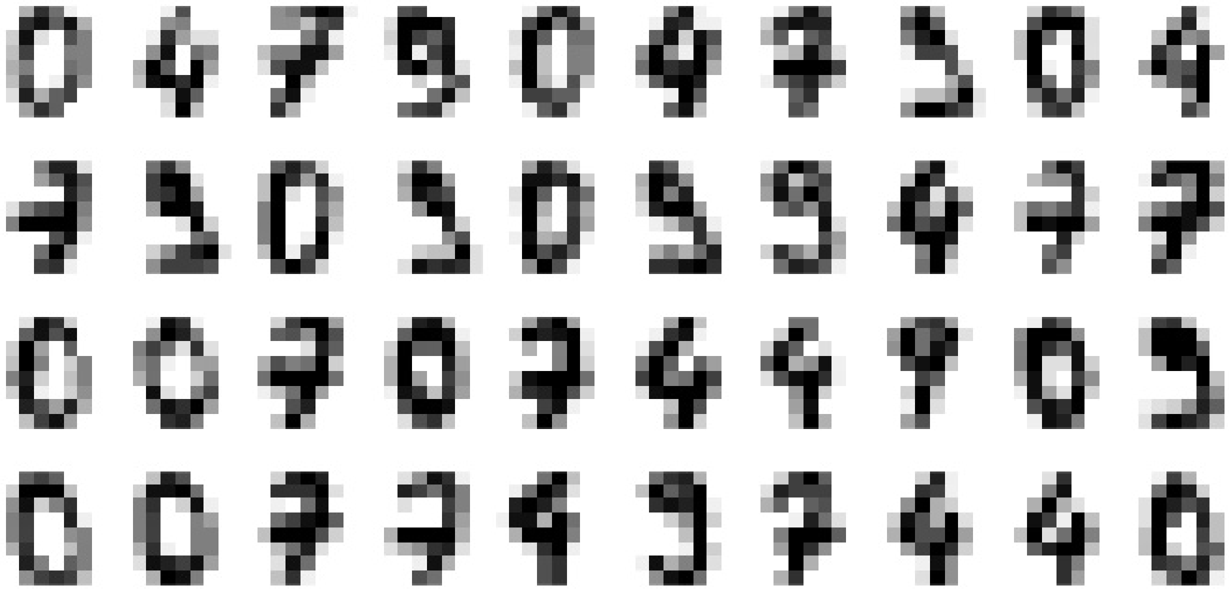
- Uniformly distributed across the n dimensional space ?
- Do they form *clusters* of examples with similar feature vectors ?

Unfortunately: it's hard to visualize n dimensions when n is large.

- By reducing the number of dimensions
- We may be able to visualize related examples
- In such a way that the reduced dimension examples don't lose too much information

Let's illustrate with a limited subset of the smaller (8×8) digits.

8 x 8 digits, subset



It would be difficult to visualize an example in $n = 64$ dimensional space.

By transforming example to a smaller number ($n' = 2$ of synthetic features we *can* visualize

- Each example is a point in two dimensional space

8 x 8 digits, subset clusters

You can see that our $m \approx 700$ examples form 4 distinct clusters.

- The clusters were formed
    - Based solely on features

It turns out that the clusters correspond to examples mostly representing a single digit.

- The clusters organized themselves based on similarity of features
- This is unsupervised ! No targets were used in forming the clusters!
- We use the hidden target merely to color the point, not to form the clusters

This hints that dimensionality reduction may be useful for *supervised* learning as well

- Use commonality of features to reduce dimension
- Reduced dimensions more independent
    - Better mathematically properies (reduced collinearity)
    - More interpretable
- Under assumption that
    - Examples with similar features (i.e., in same cluster) have similar targets

Noise reduction

Consider the MNIST example, where we reduced n by 80% without losing visual information.

This might suggest that the 80% of the features dropped

- Were signficant, but less important (dimension reduction)
- OR that the dropped features were unimportant (noise)

In the latter case, dropping features actually improves data quality by eliminating irrelevant feature.s

Matrix factorization

We will learn how to find the "most important" features by factoring the example matrix $\mathbf{X}$.

The main tool we will introduce is called *Principal Components Analysis (PCA)*.

In [4]:

```
print("Done")
```

```
Done
```