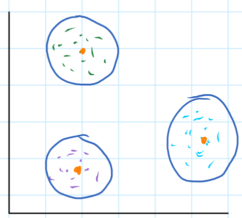
# Unsupervised Learning

what is it?
- no labels
- discover patterns in data /structures
- clustering

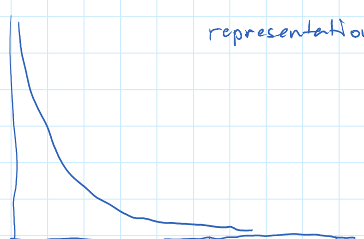
Just a bunch of vectors, What to do with them?.

What are our Goals?
- understand data
- sanity check
- remove uninformative part of data (dimensionality reduction) (compression)
- learning features (and feature selection)
- visualization

clustering

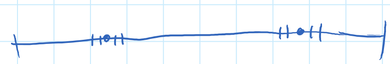


K-means: adds points to clusters




representation: 3 centers,
assignment (lossy compression)

1D:



only displacement, which is small

- new points are likely in the cluster

# Principle Component Analysis (PCA)

- dimensionality reduction method

approximate data set in some dimension $m$, $m < d$

PCA is a linear model

new dimensionaliy

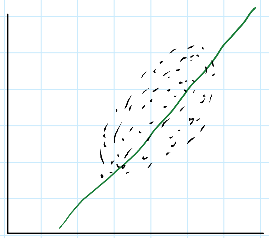
original dimensionality

amount of data

$v \in \mathbb{R}^M$

$V_n = W x_n$ $\quad W \in \mathbb{R}^{M \times D}$ $\quad X$ : whole data set $\mathbb{R}^{D \times N}$

$x \in \mathbb{R}^D$



one dimensional subspace (project)

approximate representation

"maximize variance" of projected values

- equivalent to small L2 error / fitting

$M = 1$ (for now)

$u \in \mathbb{R}^D$ ← learning $u$

$u^T u = 1$

variance of projected data:

$$= \frac{1}{N} \sum_{n=1}^{N} (u^T x_n)^2$$

(sum of squares)

$$= u^T S u,$$

where $S$ is covariance matrix $XX^T$

$$\max_u \quad u^T S u + \lambda (1 - u^T u)$$

lagrange multiplier: ie, $1 = u^T u$

$$S u - \lambda u = 0$$

$$S u = \lambda u$$

(eigenvector of covariance matrix of the data)

- but which eigenvector?

variance is now:

$$u^T S u = u^T \lambda u = \lambda u^T u = \lambda$$

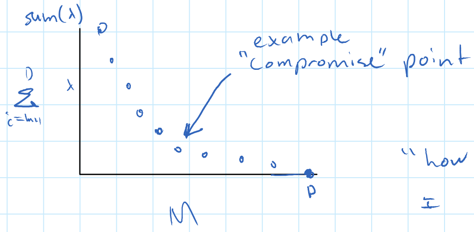
- pick biggest eigenvalue

$m > 1$ ? repeat with remaining variance

Singular Value Decomposition (SVD)

$X = U \Sigma V^T$ eigenvalues in $U$, pick biggest
another way of doing eigenvalue decomposition

How to pick $m$?




"how much of the variance am I throwing away?"

<u>Probabilistic PCA</u> - generative, higher dimension

$x_n = W z_n + \varepsilon$ ← noise ← latent space $\qquad z_n \in \mathbb{R}^m,\ x_n \in \mathbb{R}^D \qquad W \in \mathbb{R}^{D \times M}$

$z_n \sim N(0, \mathbb{I})$ iid

$\varepsilon \sim N(0, \sigma^2 \mathbb{I})$ iid (isotropic noise)

$$p(x \mid z) \sim N(Wz, \sigma^2 \mathbb{I})$$

$$p(x) \sim N(\mu, C)$$

$$\mu = E[x] = E[Wz + \varepsilon] = 0$$

$$C = \text{cov}[x] = E[(Wz + \varepsilon)(Wz + \varepsilon)^T]$$

$$= E[Wzz^TW^T] + E[\varepsilon\varepsilon^T]$$

$$= WW^T + \sigma^2 \mathbb{I}$$

MLE is PCA

$W$ only appears as $WW^T$

$\tilde{W} = WR$ ← orthogonal

$\tilde{W}\tilde{W} = WRR^TW^T = WW^T$

could be considered undesireable, any orthogonal transformation on $W$ gives the same model

Cocktail Party Problem

microphones listening to sum of 5 voices

decompose signal to 5 sources

## Blind Source Seperation

treat as iid data

$$z = \begin{bmatrix} \text{person } 1 \\ \text{person } 2 \\ \vdots \end{bmatrix}$$

infer distances, dissambiguate people

observed

$x = Wz$

$t$ = # time samples

$n_s$ = # of speakers

$n_m$ = # of microphones

# unknowns: $n_s n_m + t n_s$

# equations: $t n_m$

if $t n_m >$ # unknowns, we can hopefully solve it

PCA: disaster! $z$ means 5 people

PCA could be multiplied by some orthogonal matrix

the problem has a "correct answer"

## Independant Component Analysis (ICA)

- linear model (like PCA)
- NOT Gaussian prior on $z$
- $p(z) = \prod_{m=1}^{M} p(z_n)$

## Factor Analysis

- like Probabilistic PCA, but: instead of variance $\sigma^2 I$ variance, each element can have its own.