Last Time

- Covered broadly the methods of recommendation engines. Matrix Factorization, Neighborhood Models, Graph Diffusion, etc. We missed multi-armed bandits but will cover this after hypothesis testing.

- I believe I've approved all of the project proposals but contact me if I haven't or if you're switching projects.

- Let me know if you're interested in the fellowship but I will most likely reserve decisions until next month when the course is done.

Lecture 7 - Clustering Methods

Lecture Outline

- What is unsupervised learning?
- K means, K Nearest Neighbors.
- Curse of dimensionality - why most clustering fails in high dimensions.
- Support Vector Machines and Kernels - The Kernel Trick.
- Principal Component Analysis.
- Mixture Models (if time permits)
- Topic Models and LDA (if time permits)

What is a “good” clustering?

Theorem 2 (Bennett et al. [1999])

Assumption *Two clusters are pairwise stable, i.e., the between cluster distance dominates the within cluster distance.*

Consequence *We can meaningfully discern “near” neighbors (members of the same cluster) from “far” neighbors (members of the other cluster).*

Curse of Dimensionality

Problems with higher dimensions

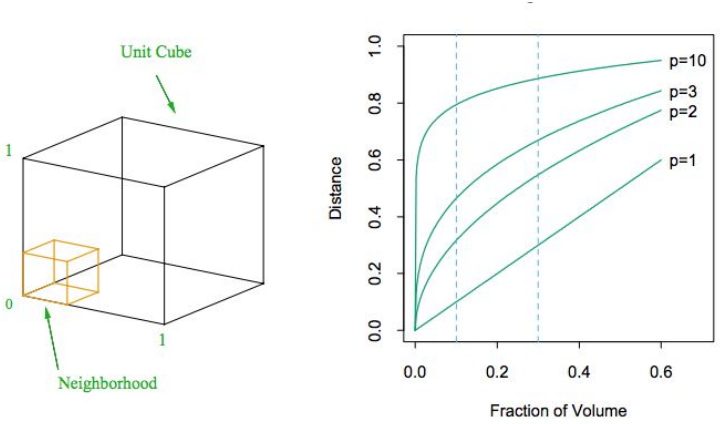


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

Question:

Consider N data points uniformly distributed in a p-dimensional unit ball centered at the origin. Suppose we consider a nearest-neighbor estimate at the origin. What is the median distance to the nearest point?

Problems with higher dimensions

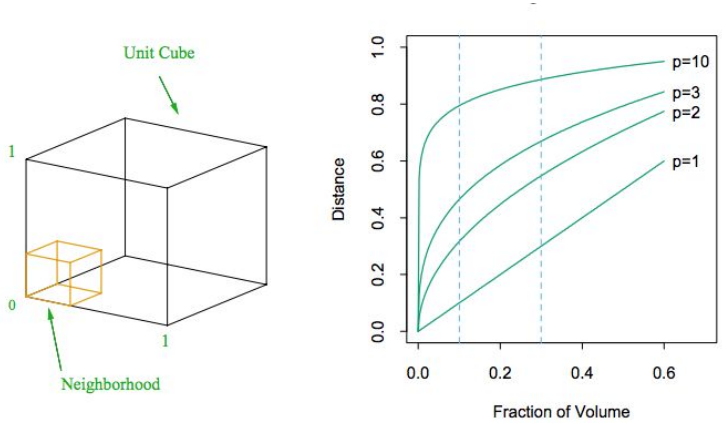


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

Question:

Consider N data points uniformly distributed in a p-dimensional unit ball centered at the origin. Suppose we consider a nearest-neighbor estimate at the origin. What is the median distance to the nearest point?

$$d(p, N) = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/p}$$

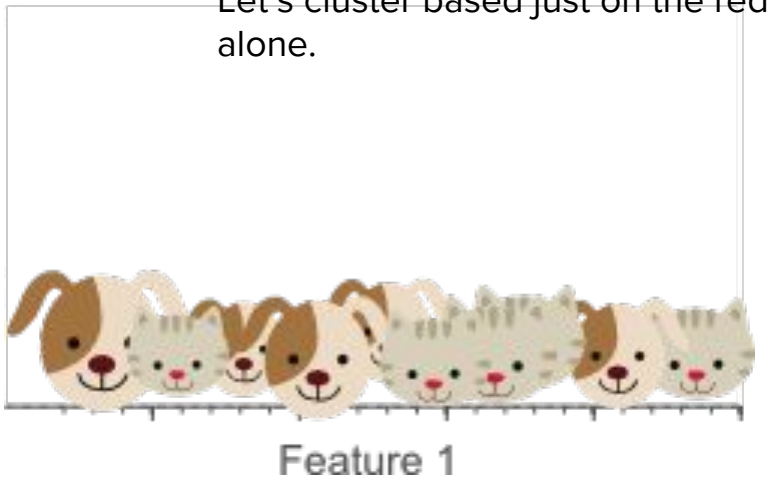
Conclusion:

- Distance to the nearest neighbor takes up most of the domain when the dimension is large.
- All points spread to the boundaries, making clustering based on these distances poor.

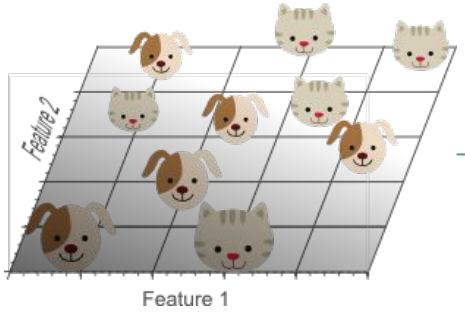
Cats and Dogs

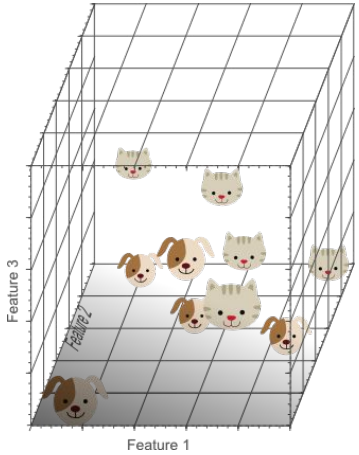
*If 0.5*red + 0.3*green + 0.2*blue > 0.6 : return cat;*
else return dog;

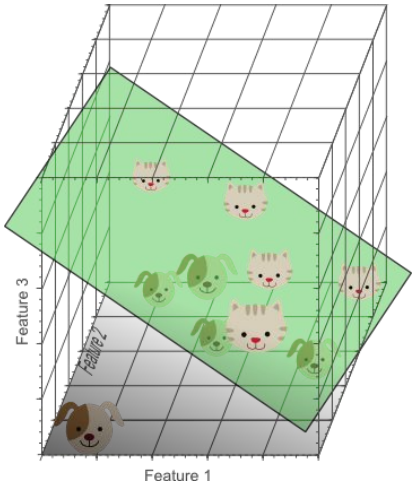
Let's cluster based just on the redness alone.



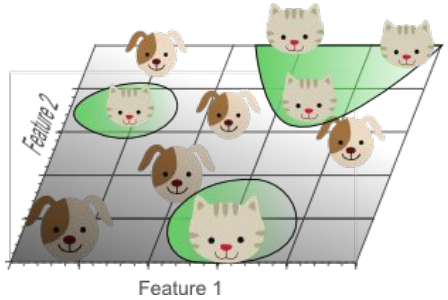
Cats and Dogs Overfitting





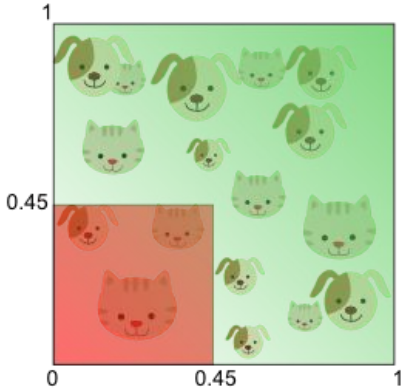


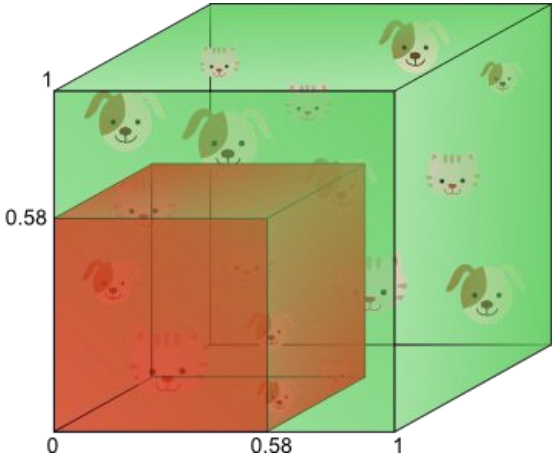
Perfect



Not good!

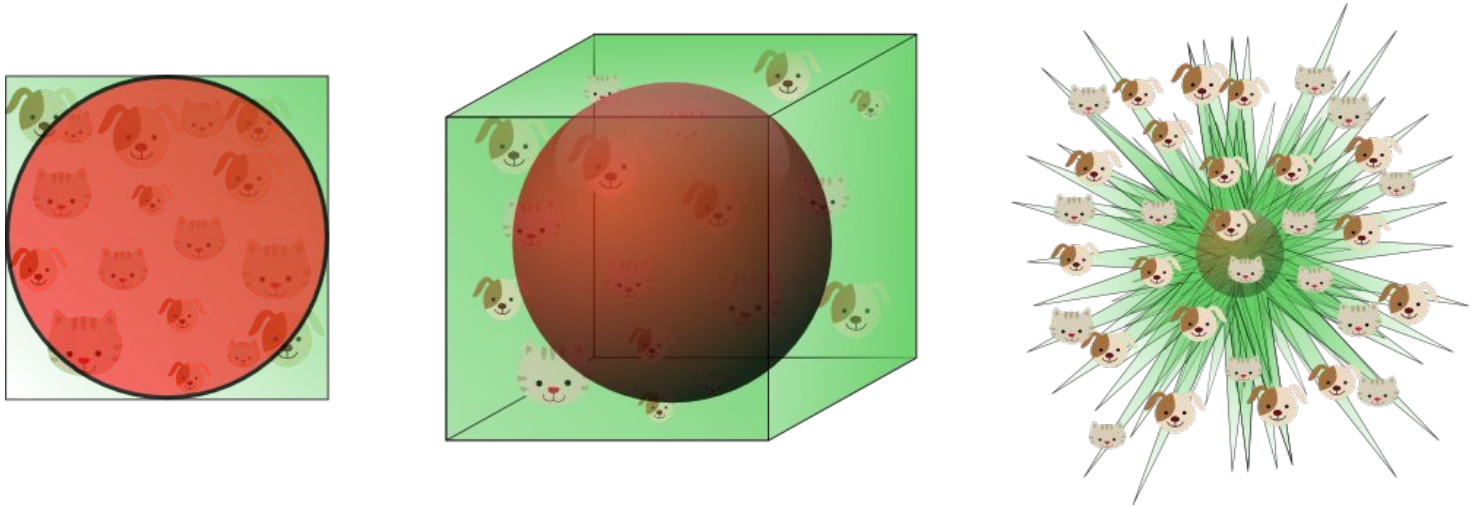
Problems with higher dimensions





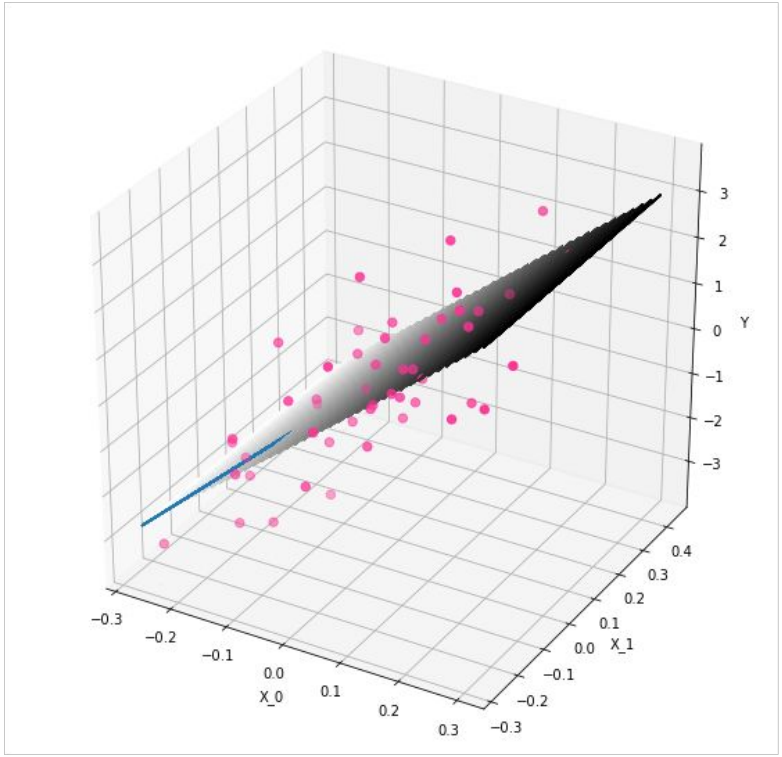
The amount of training data needed to cover 20% of the feature range grows exponentially with the number of dimensions.

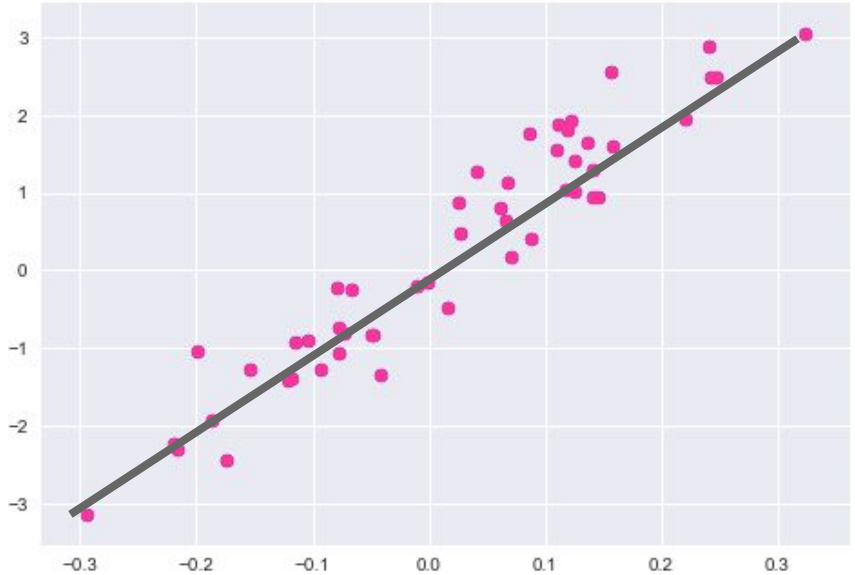
Problems with higher dimensions



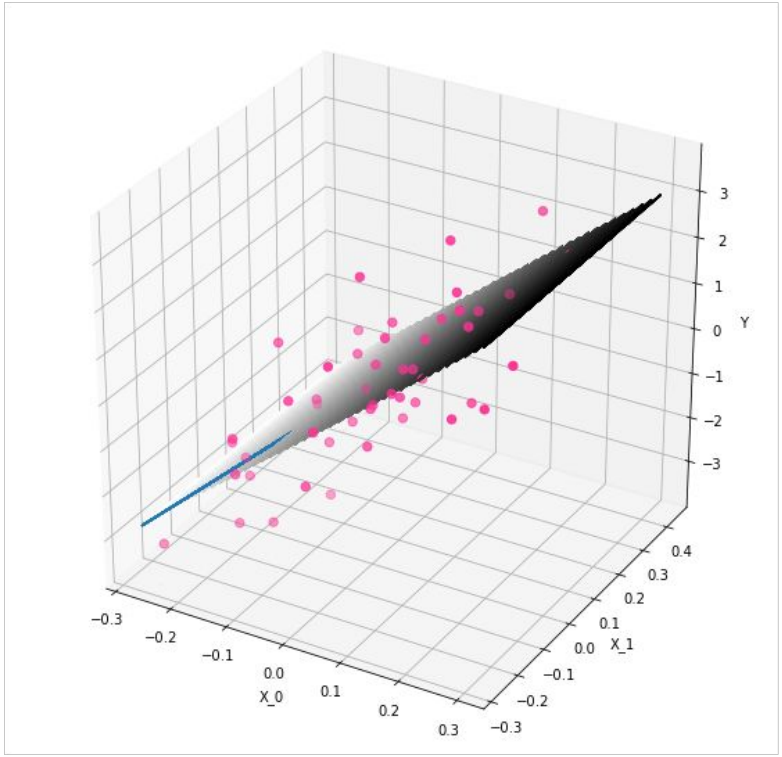
As the dimensionality increases, a larger percentage of the training data resides in the corners of the feature space.

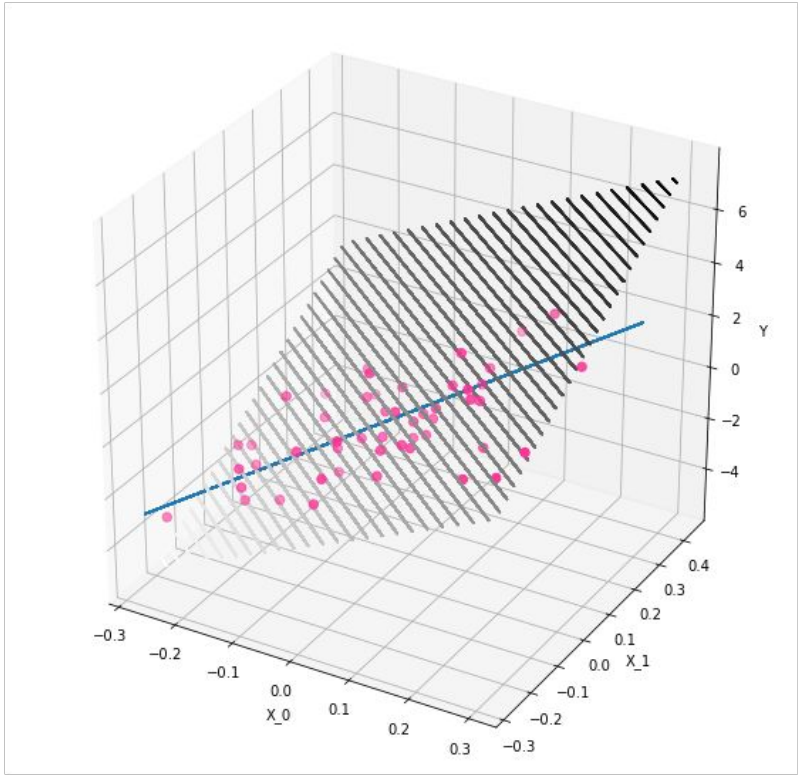
Same thing with linear models





Same thing with linear models

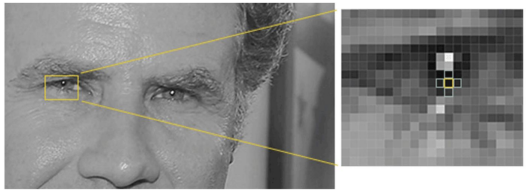




Some solutions

	t = 10		t = 20		t = 50		t = 100	
	P	F	P	F	P	F	P	F
SVD	0.5900	0.0262	0.6750	0.0268	0.7650	0.0269	0.6500	0.0324
LLE	0.6500	0.0245	0.7125	0.0247	0.7725	0.0258	0.6150	0.0337
LS	0.3400	0.0380	0.3875	0.0362	0.4575	0.0319	0.4850	0.0278
HD	0.6255	0.0220	0.6255	0.0220	0.6255	0.0220	0.6255	0.0220
RP	0.4225	0.0283	0.4800	0.0255	0.6425	0.0234	0.6575	0.0219

t= dimension
P = misclassification rate
40 possible faces they're trying to classify

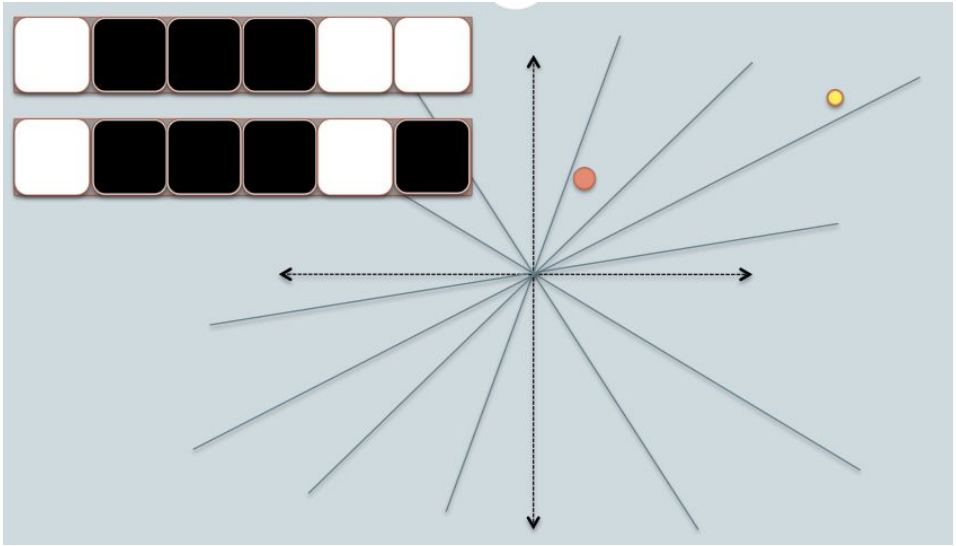


$T_{SVD} = 5.9, T_{LLE} = 4.4, T_{LS} = 0.32, T_{HD} = 0$, and $T_{RP} = 0.03$.

(i) **SVD:** the Singular Value Decomposition (or Principal Components Analysis)
(ii) LLE:
(iii) **LS:** the Laplacian score feature selection.
(v) **HD:** k-means algorithm on the High Dimensional data;
(vi) **RP:** the random projection method (Locally Sensitive Hashing)

http://papers.nips.cc/paper/3901-random-projections-for-k-means-clustering.pdf
(2012

Locality Sensitive Hashing



- We initialize a set of random hyperplanes.
- If a given point lies above a plane, we code it with black. Otherwise we code it with white.
- This gives a hash which is very close for two data points if their cosine similarity is close.
- Very fast searching algorithm and also fast for k means clustering.

How do we deal with this?

1) Project the data into a lower space with random projections,

2) Apply your favourite low-dimensional clustering algorithm in this space (e.g. k-means).

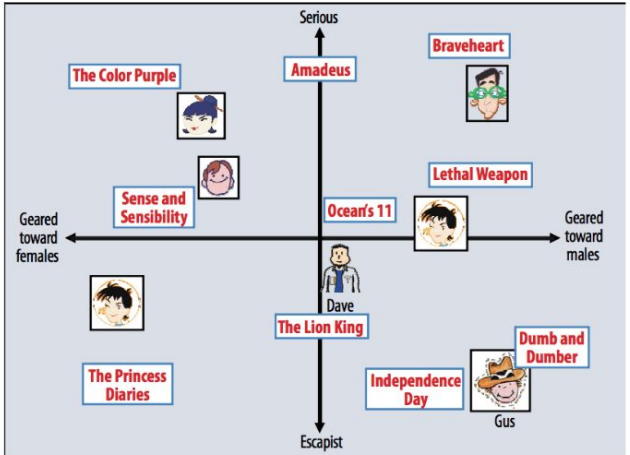


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

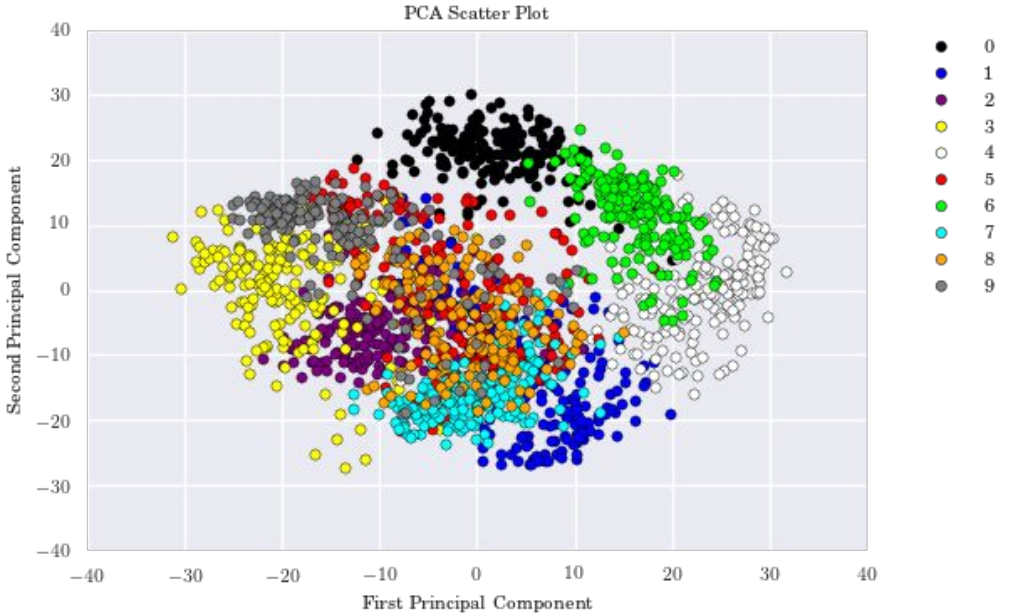
K Means Clustering

K means clustering

$$\operatorname{argmin}_S \sum_{k=1}^{K} \sum_{\mathbf{x} \in S_k} \|\mathbf{x} - \mu_i\|^2$$

$$S = \{S_1, S_2, \cdots, S_K\}$$

- S_k sub region of space.
- μ_i centroid of the set S_k



Break your space into K fixed regions such that the distance (squared) from points to the centroid is of the region is minimized (**you choose K**).

The algorithm

1. Define initial points for the centroids

2. At time step t, assign a point to a cluster if it has the smallest squared distance to the centroid associated with it:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\},$$

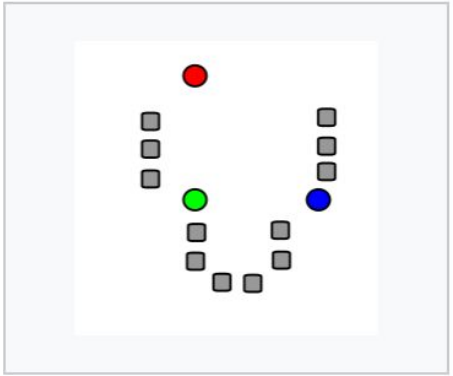
3. Recompute the new centroid based on the updated sets.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

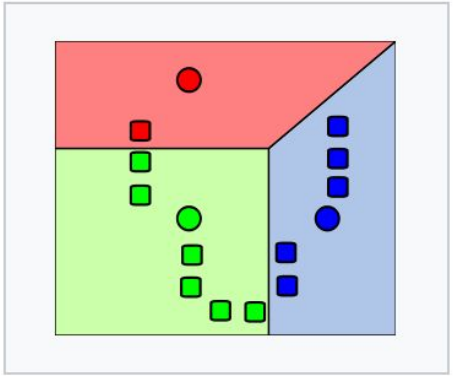
4. Continue 2-4 until regions don't change anymore.

The algorithm

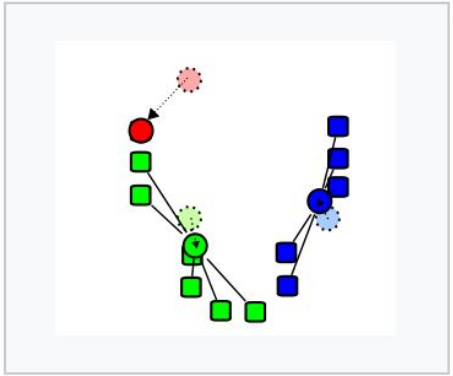
Demonstration of the standard algorithm



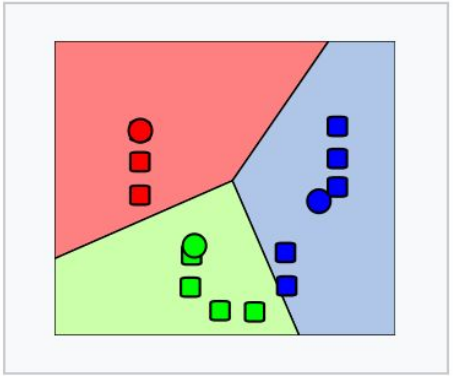
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

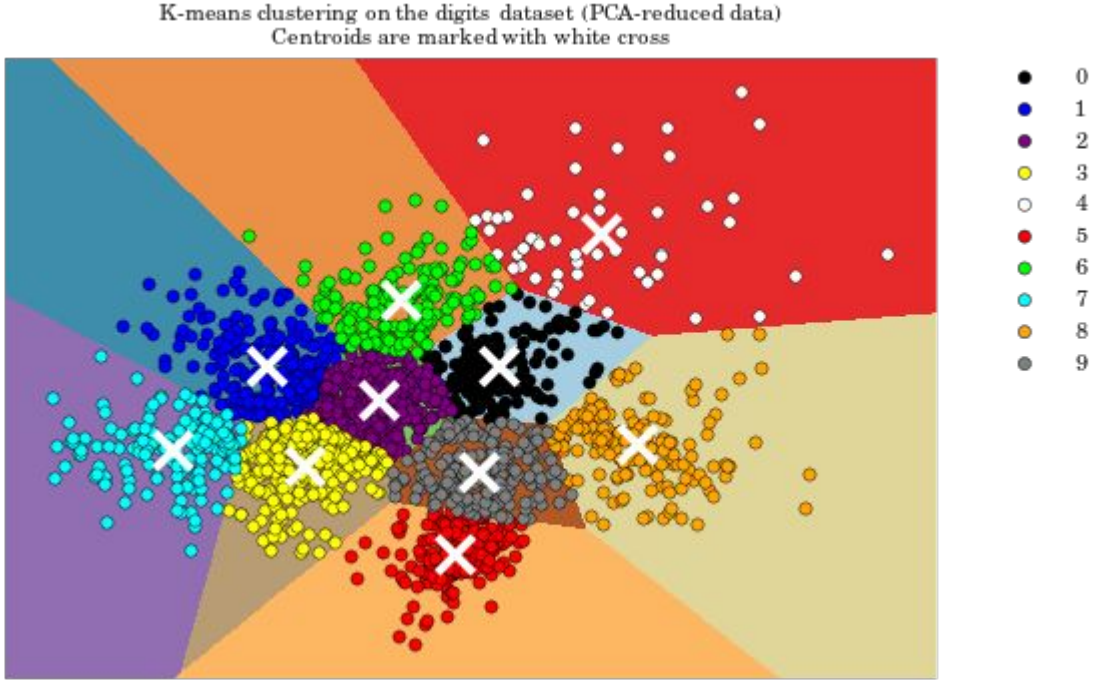


3. The centroid of each of the *k* clusters becomes the new mean.



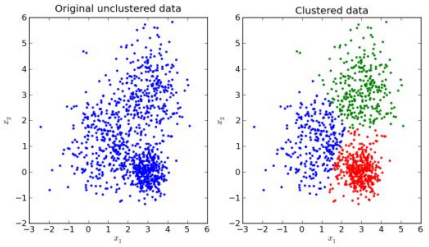
4. Steps 2 and 3 are repeated until convergence has been reached.

Example 1 - Digit Recognition



K means after applying PCA to plot in 2d

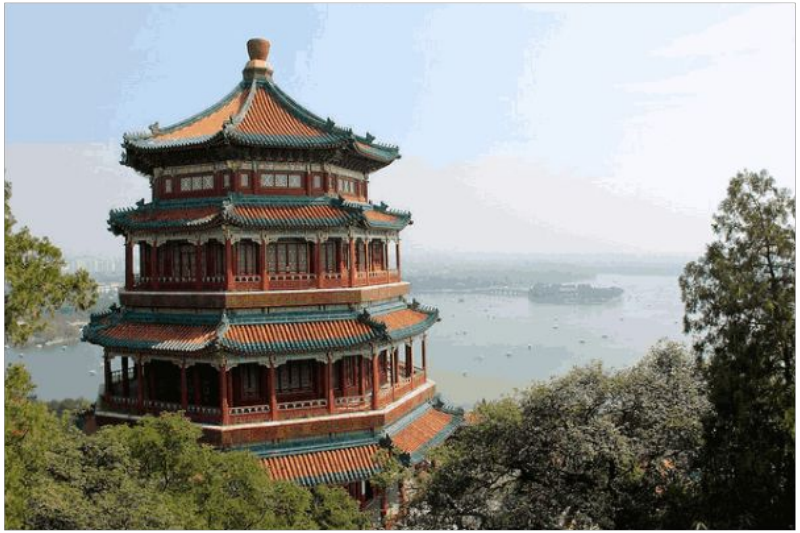
Example 2 - Image Compression



Original image (96,615 colors)

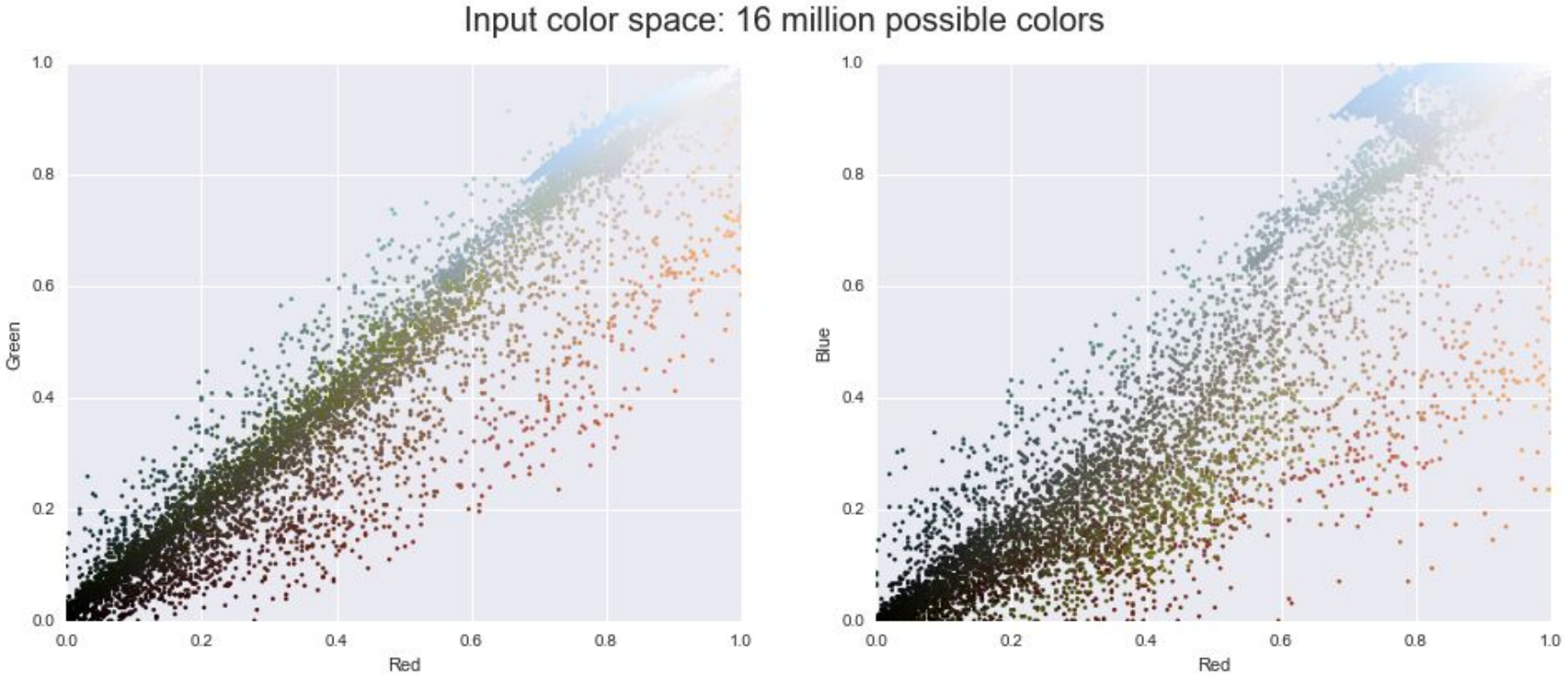
96, 615 colors

Quantized image (64 colors, K-Means)



64 colors

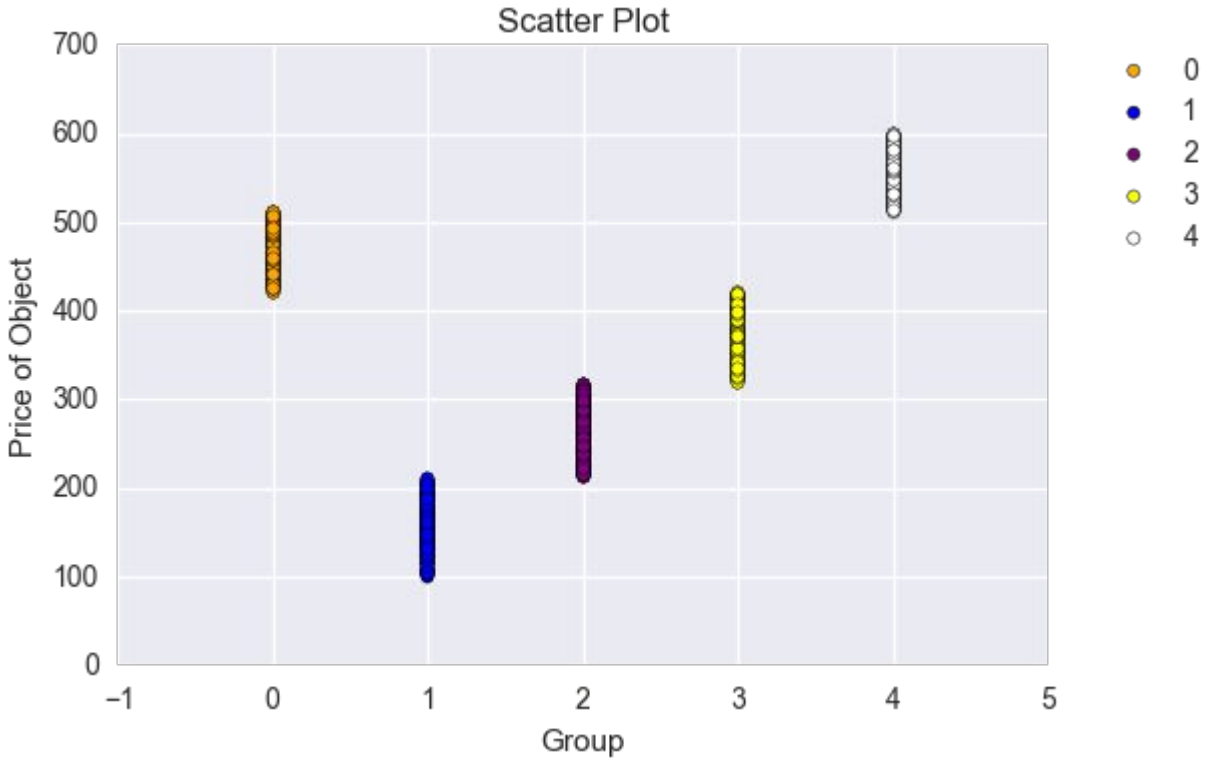
Dimensionality reduction on pixel colors



Example 3

Problem: You have 100 items at a yard sale, and you know the exact price of each item. However you want to simplify matters and have only 5 boxes, with the price of each item being the same in any particular box. How do you choose the prices to put on the boxes?

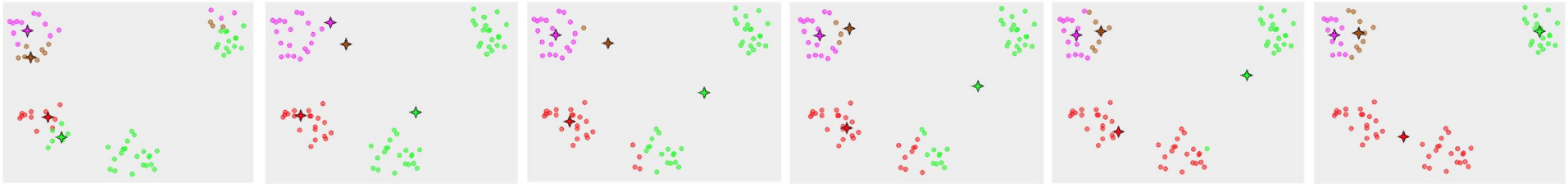
Pricing Groups



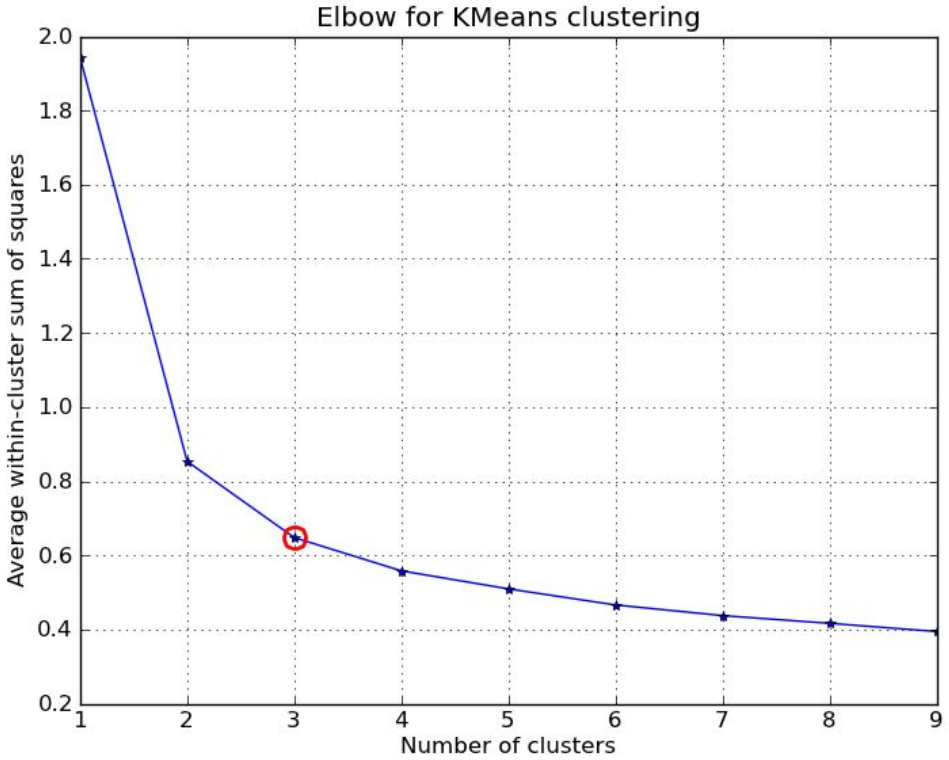
Solution: Break prices into 5 groups. Then for each group, the items that you're selling from that box will have the smallest distance to the 'centroid' or average price.

Properties

- Euclidean distance is used as a metric and variance is used as a measure of cluster scatter.
- The number of clusters *k* is an input parameter: an inappropriate choice of *k* may yield poor results. That is why, when performing k-means, it is important to run diagnostic checks for determining the number of clusters in the data set.
- Convergence to a local minimum may produce counterintuitive ("wrong") results (see example in Fig.).



Elbow method for number of clusters



To choose the ‘optimal’ number of clusters, a typical technique is to choose the ‘elblow point’ on the graph, where the decrease in variance ‘levels off’.

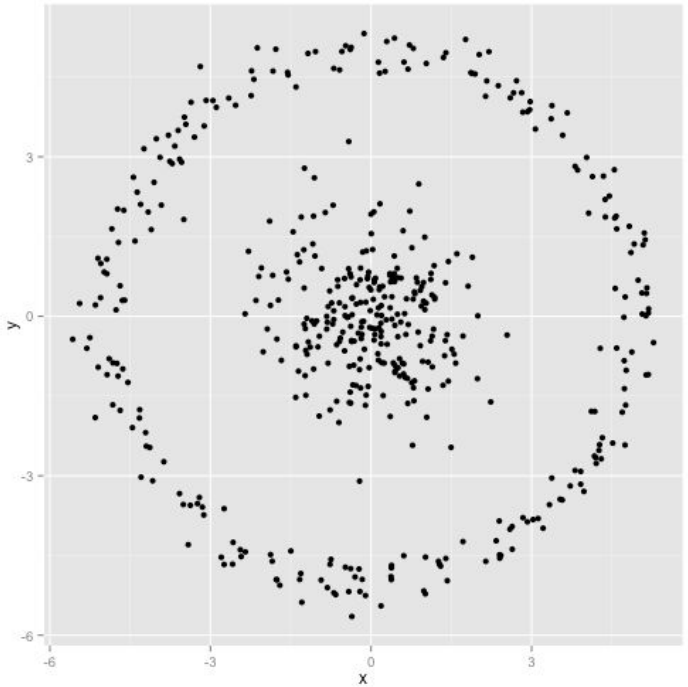
Clustering Continued

- K-means assumptions and limitations revisited.
- Mixture Models - A generalization of K-means.
  - Solving via Expectation Maximization. An optimal lower bound.
  - Getting stuck in local minima - some concrete examples, and how to get out.
  - Deriving K-means as a limit of Gaussian Mixture Models.
- Support Vector Machines
  - Problem formulation and derivation of the dual problem.
  - Convex Duality and the Legendre Transform.
  - What is useful about the dual problem? The Kernel Trick.
- Density Based Clustering.

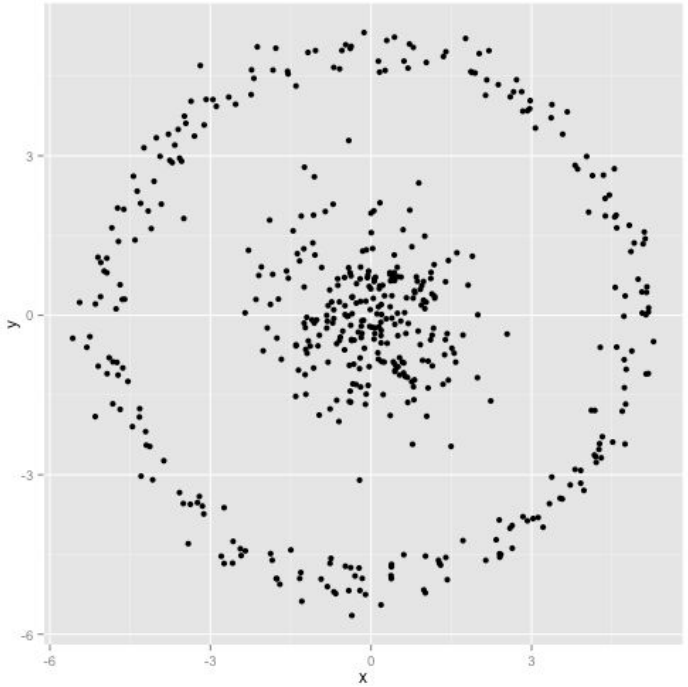
K means assumptions

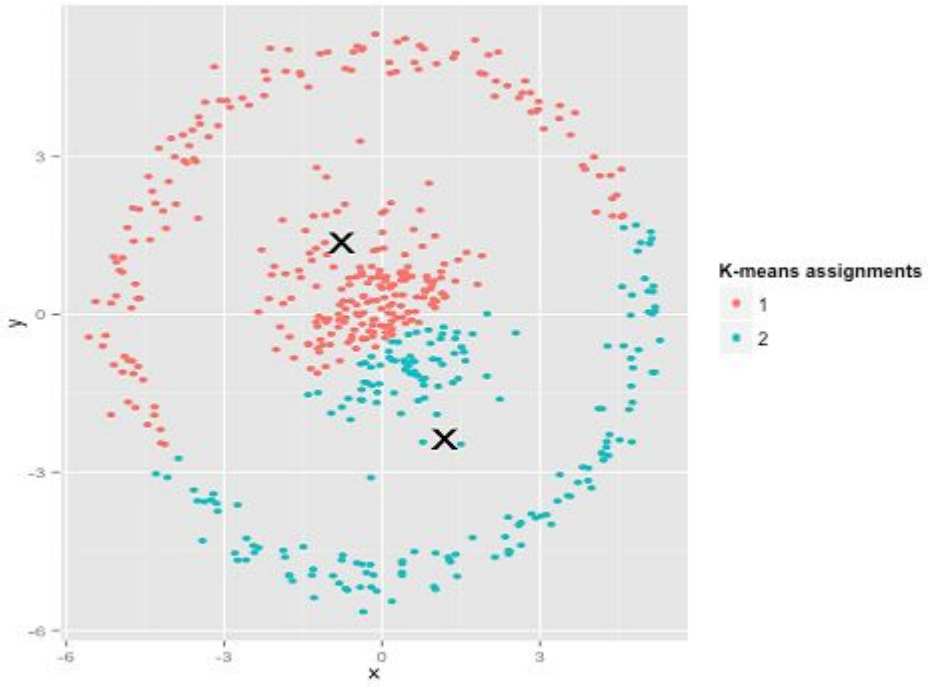
- k-means assumes the variance of the distribution of each attribute (variable) is spherical;
- all variables have the same variance;
- the prior probability for all k clusters is the same, i.e., each cluster has roughly equal number of observations;

K means with spherical data

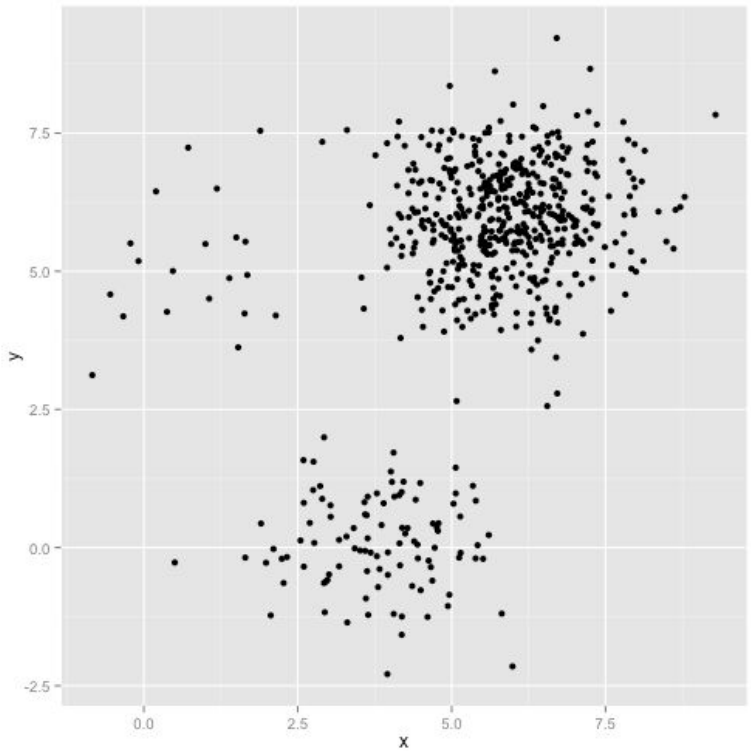


K means with spherical data

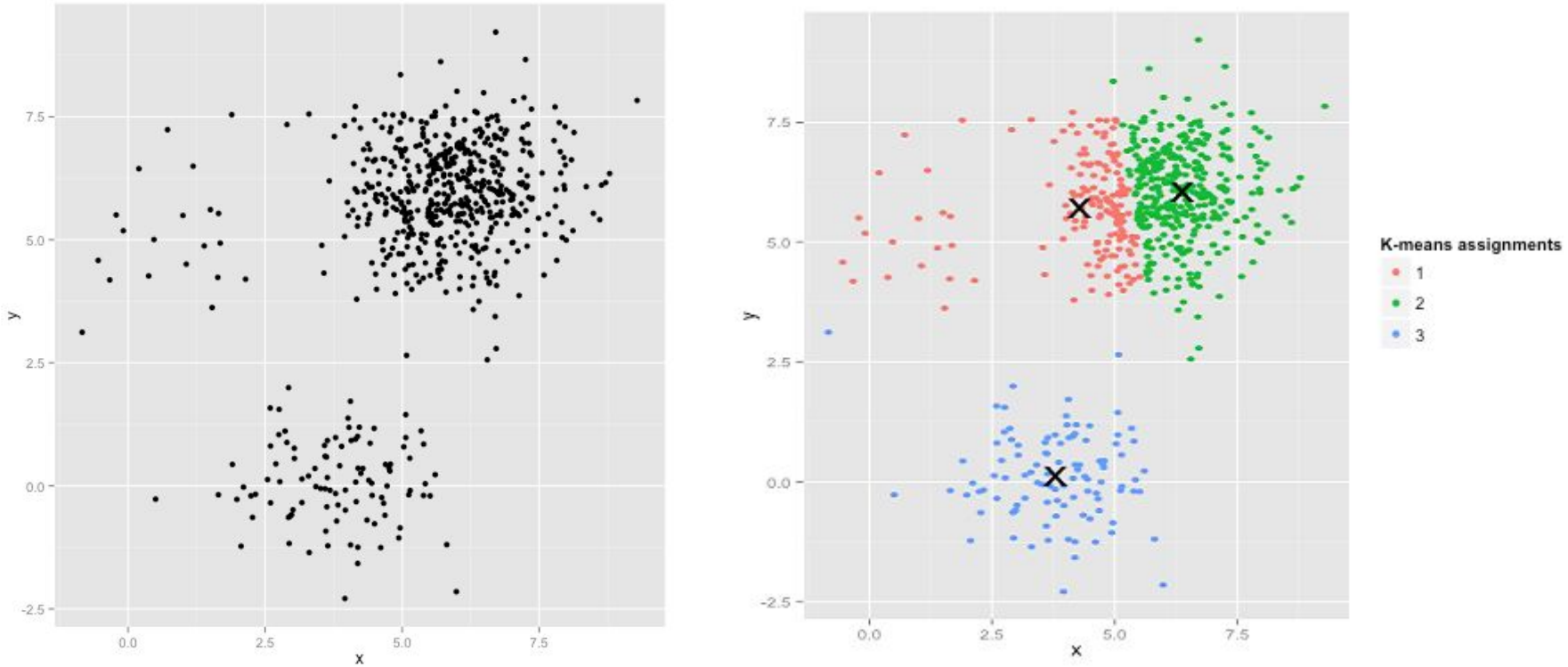




K means bias for groups with more points K=3

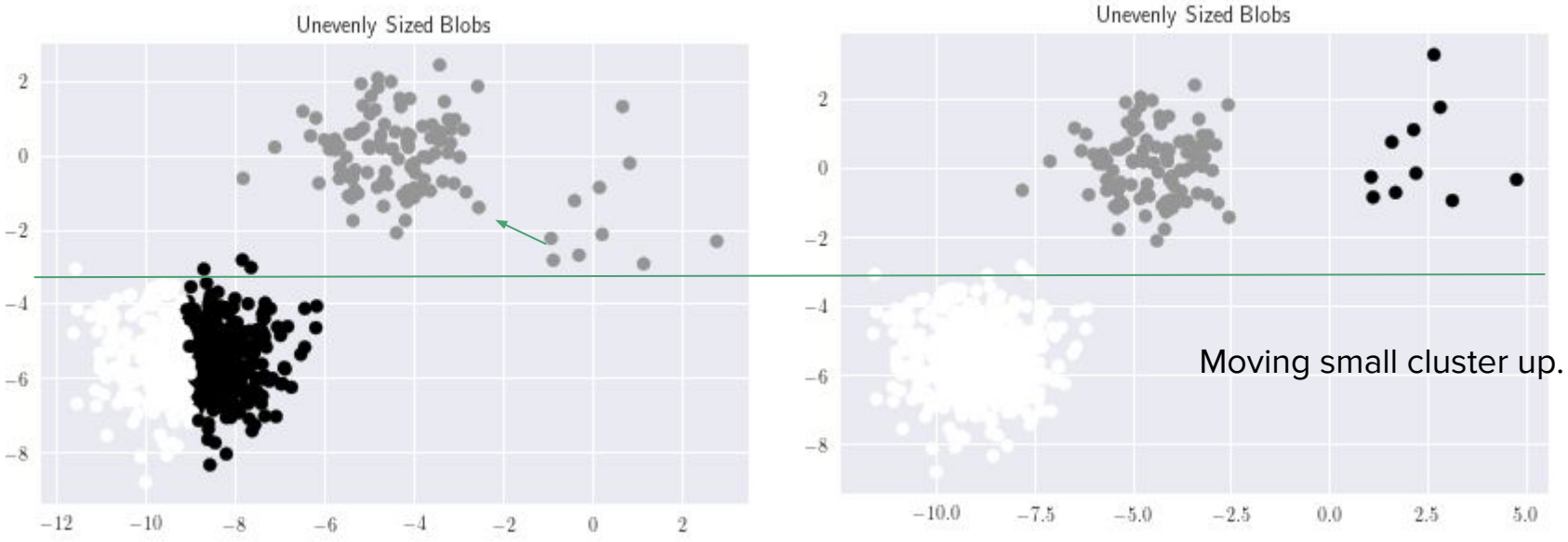


K means bias for groups with more points K=3



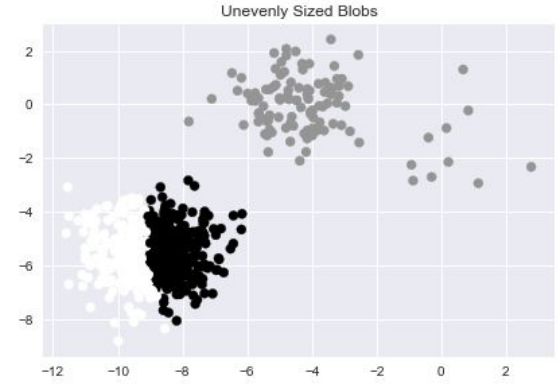
K means wants to break up large clusters in order to minimize total variance.

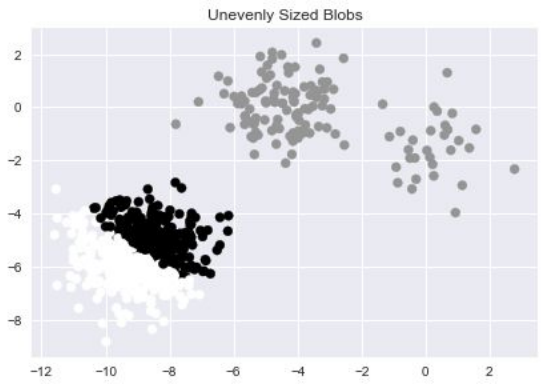
Small clusters can be absorbed in K means

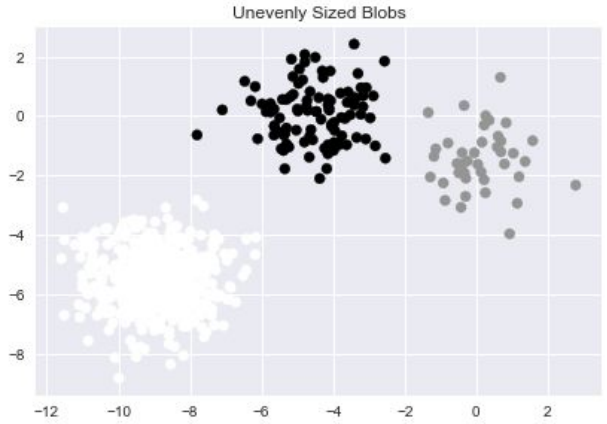


What happened here is a bit subtler. In its quest to minimize the within-cluster sum of squares, the k-means algorithm gives more “weight” to larger clusters. In practice, that means it’s happy to let that small cluster end up far away from any center, while it uses those centers to “split up” a much larger cluster. **The few grey points to the right make up a small fraction of the total cluster size, so they are weighted less. It’s cheaper to absorb these small collection of points.**

Increasing size of small cluster



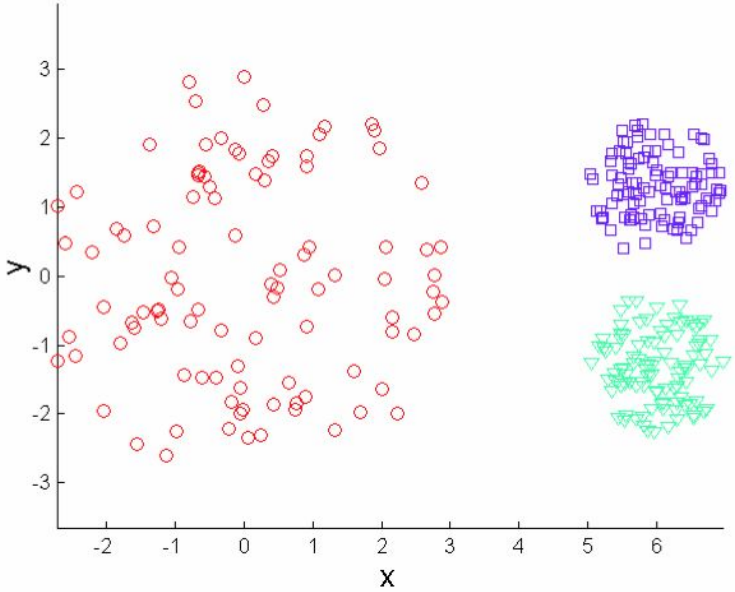


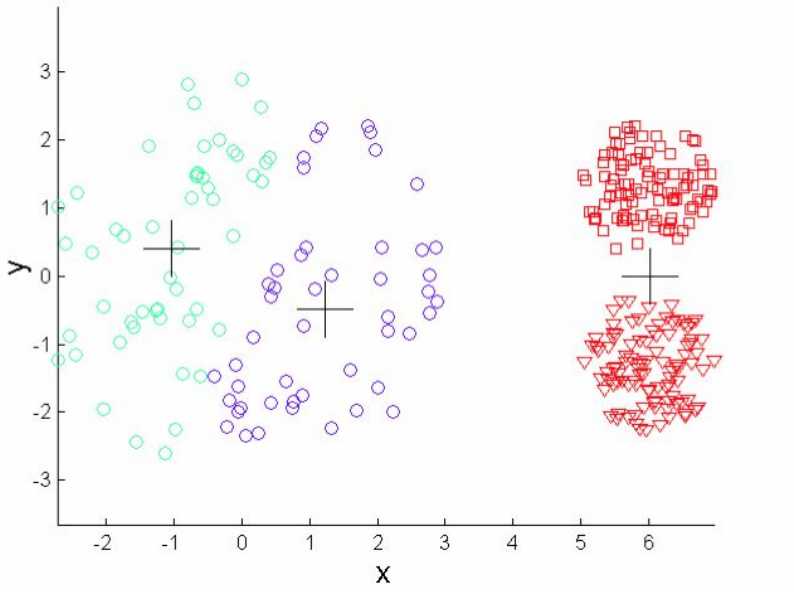


All clusters sampled from Gaussians with unit variance.

Increasing size of small cluster.

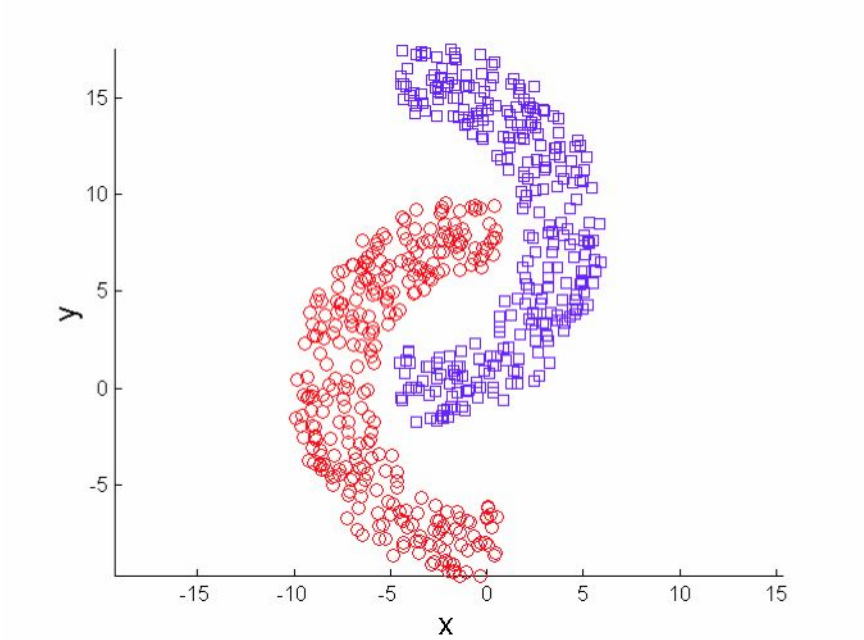
Limitations of K means continued.

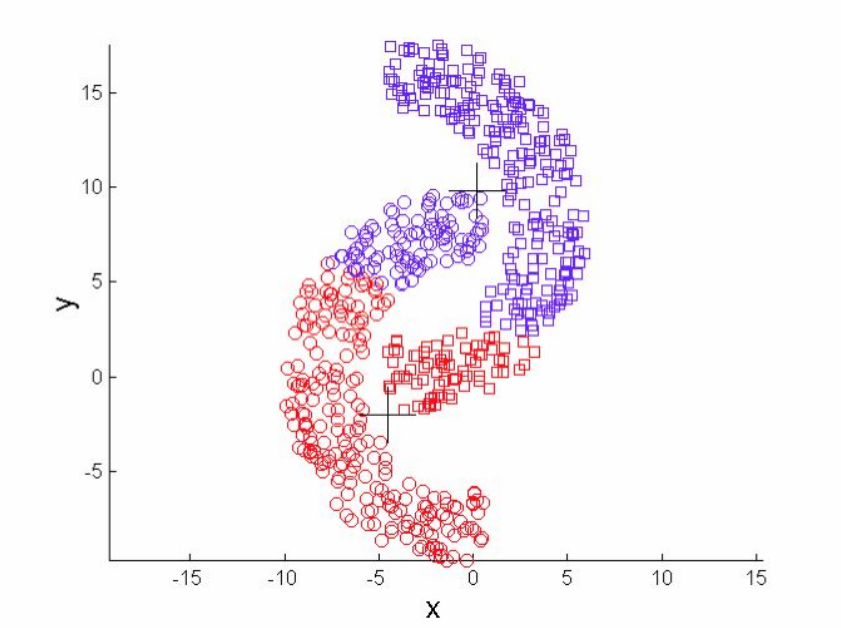




Wants to minimize the spread here. Imagine the red balls were really far away.

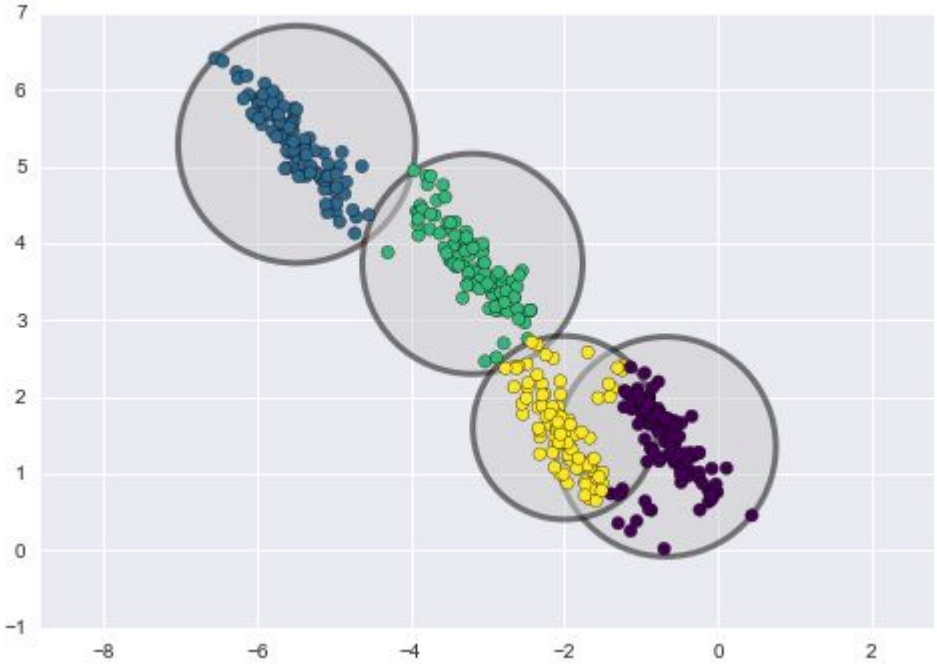
Limitations of K means continued.





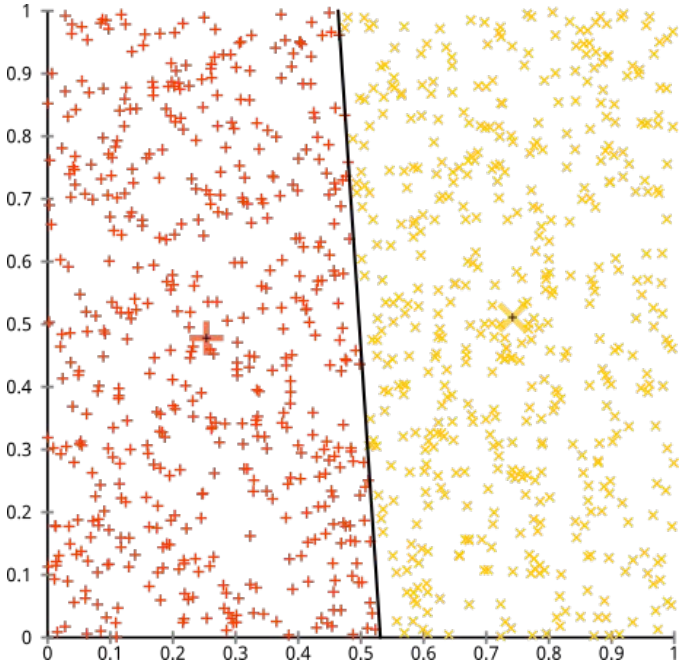
Can’t separate points with nonlinear boundaries.

Limitations of K means



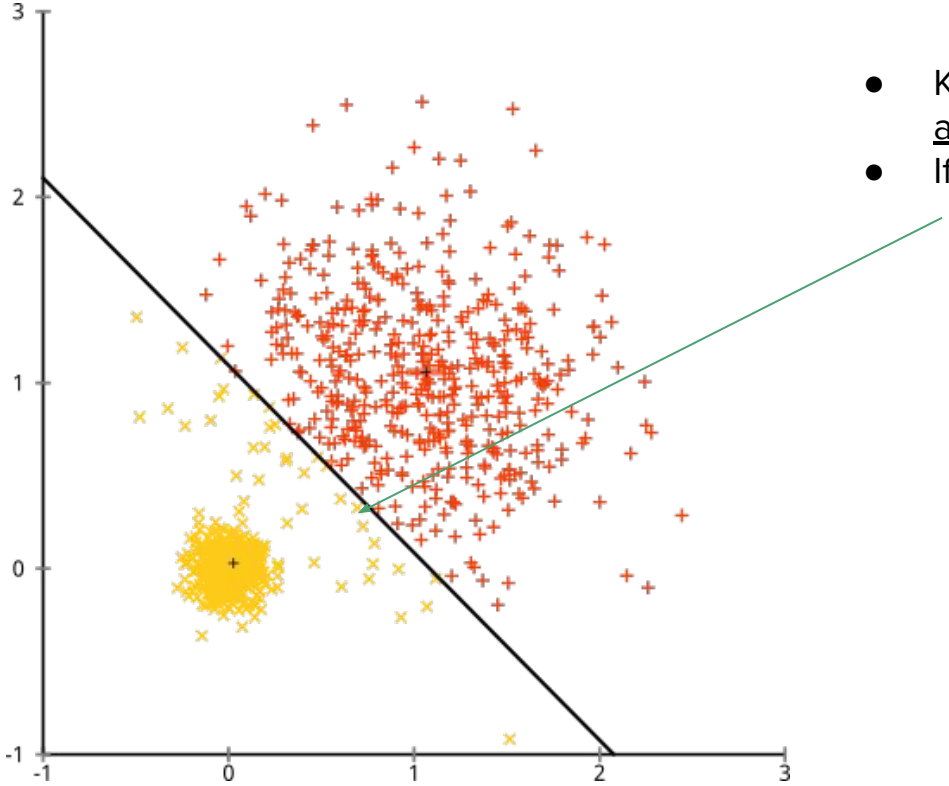
All of the clusters must be circular

K means splits uniform data



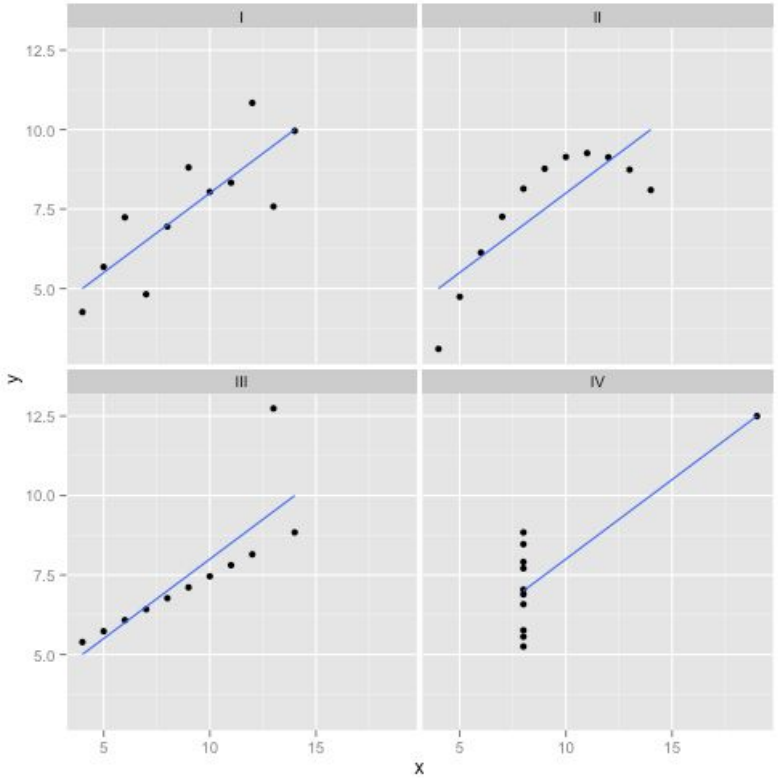
- The goal of K means is to minimize variance, so it will split groups into clusters even if there is random uniform data.

Sensitive to scale



- K means assumes that the variances of the clusters are the same.
- If they're not, incorrect clusterings can be created.

Square pegs into round holes

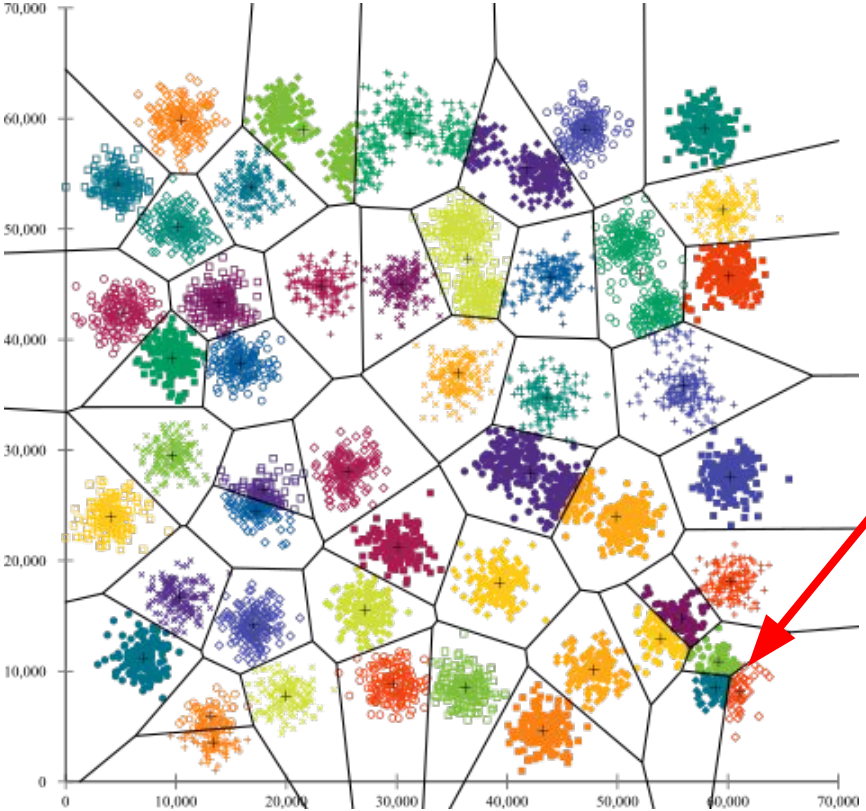


- Created in 1973 by statistician Francis Anscombe, this delightful concoction illustrates the folly of trusting statistical methods blindly. Each of the datasets has the same linear regression slope, intercept, p-value and R2- and yet at a glance we can see that only one of them,

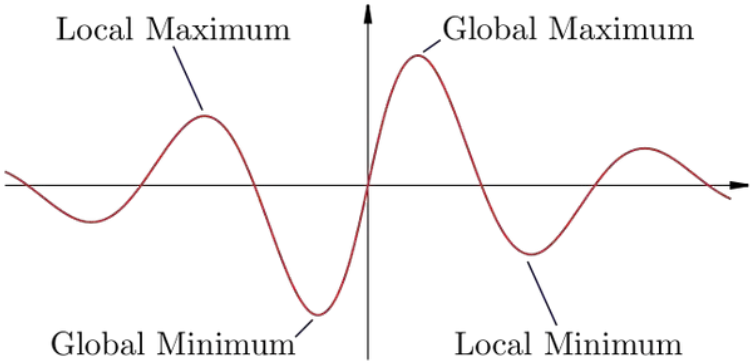
- In **I**, it is appropriate for linear regression.
- In **II** it suggests the wrong shape,
- In **III** it is skewed by a single outlier- and in
- In **IV** there is clearly no trend at all!

Normally we solve this by evaluating on testing data, but we no longer can! Since we don't have labels!

Stuck in local minima anyway



- A3 data set (gamma radiation levels).
- Each cluster is generated by iid gaussians with equal variance.
- The algorithm still gets stuck in local minimums even with perfect data.



Summary

- K means is a simple and effective clustering algorithm in many cases where you need to group things into a small number of categories.

- Like many algorithms, it's very simple to misuse it! It's essential to always have a good understanding of the underlying data before using any model. This is especially true in unsupervised learning, because we can't see how well the model generalizes to test data!

Announcements

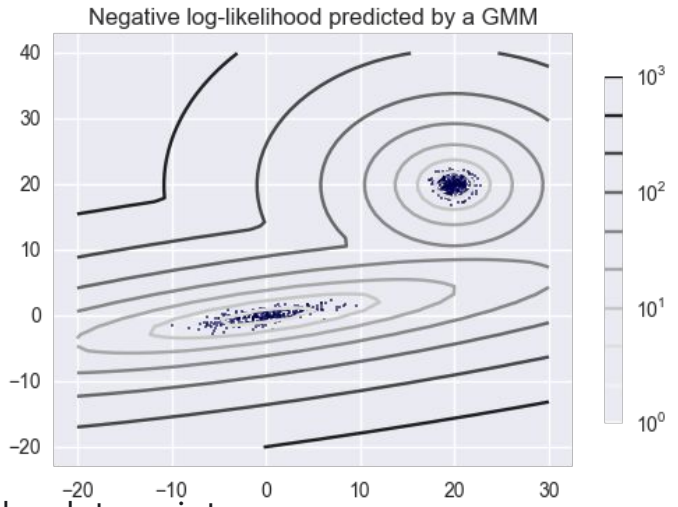
- All projects have been approved (even if you don't see a Y for some reason in the spreadsheet).
- **Final exam:** Thursday May 10, 6:00pm - 10:00pm. I've allotted extra time due to the number of students. **Everyone must include a link to their repo which includes the code base, a notebook with your work, and a pdf with your presentation.**
- **CVN Students:** If you are a CVN student, please submit a 5-10 minute video presenting your work.
- **Next week:** Either office hours or reinforcement learning.
- **Today:** Clustering Continued: Gaussian Mixture Models and Support Vector Machines.

Mixture Models

Expectation Maximization

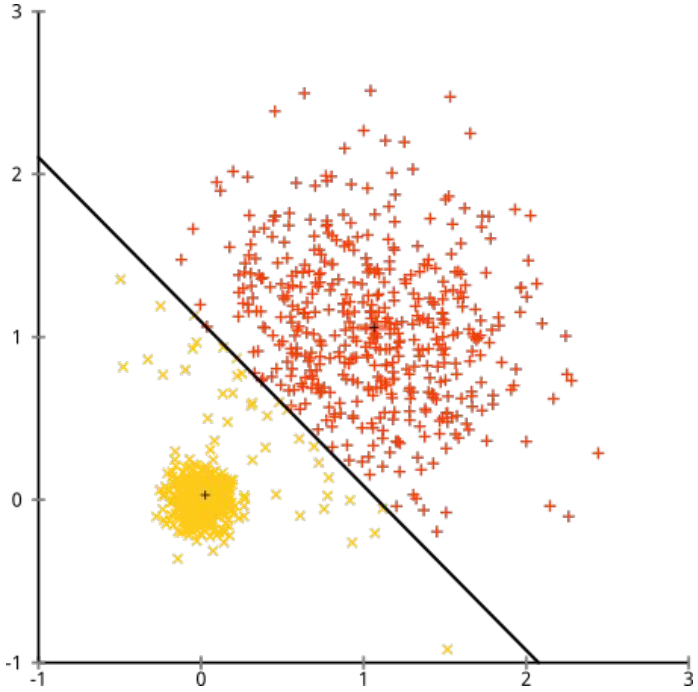
Gaussian Mixture Models

$$X \sim \pi\mathcal{N}(\mu_1, \sigma_1) + (1 - \pi)\mathcal{N}(\mu_2, \sigma_2)$$

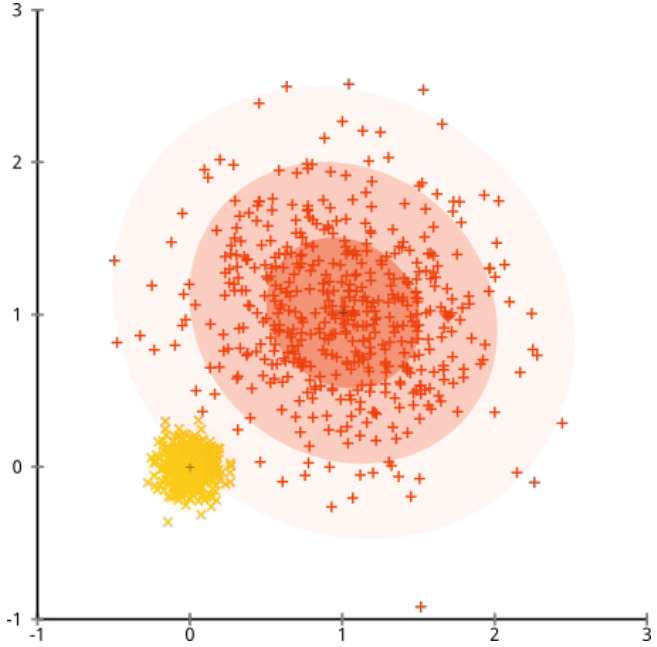


A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. One can think of K means as the small variance limit when all variances are assumed to the same.

K means vs GMM



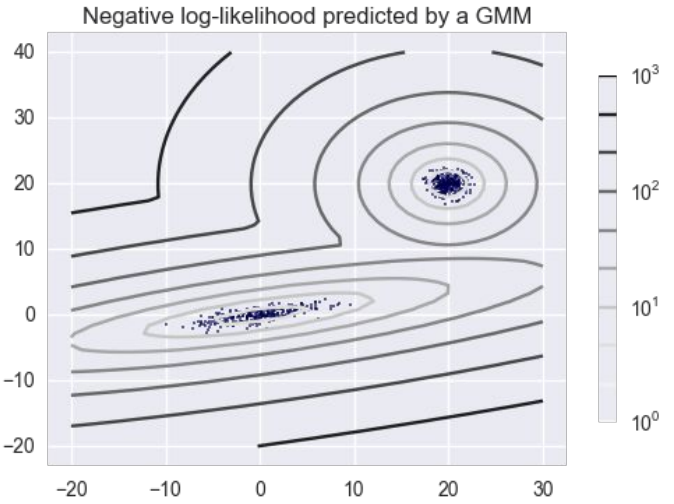
K Means



Gaussian Mixture Model

Gaussian Mixture Models

$$X \sim \pi\mathcal{N}(\mu_1, \sigma_1) + (1 - \pi)\mathcal{N}(\mu_2, \sigma_2)$$



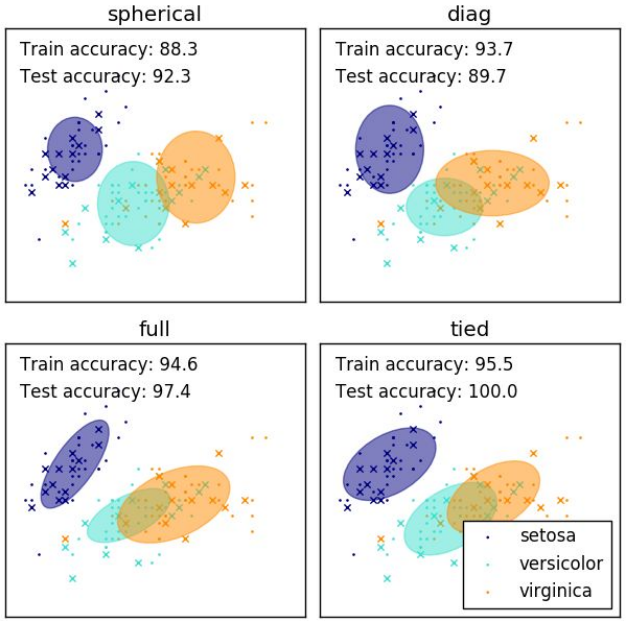
- X is the collection of data (unlabeled).
- Are the two normal distributions that you're trying to fit the data to with means/variances μ_i, σ_i
- π represents the portion of the distributions

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \quad \boldsymbol{\sigma} = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}.$$

Some properties

- MM is *a lot* more **flexible** in terms of **cluster covariance.**
- k-means is actually a special case of GMM in which each cluster's covariance along all dimensions approaches 0.
- Think of rotated and/or elongated distribution of points in a cluster, instead of spherical as in k means.
- Allows for *mixed membership* - ie. a New York Times article may belong to several categories, not just one.

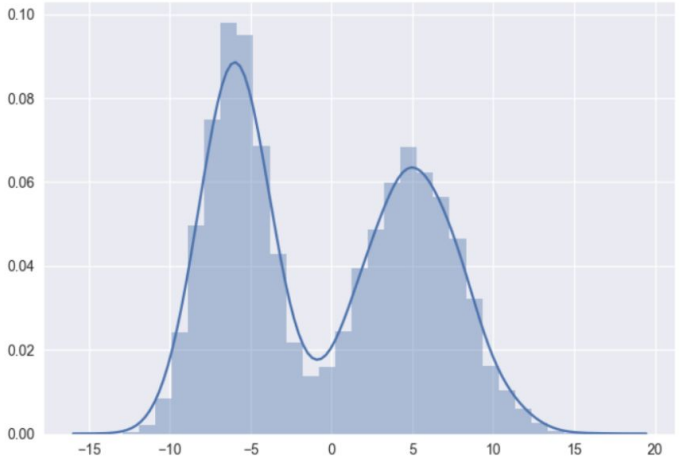
$$\boldsymbol{\mu} = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \quad \boldsymbol{\sigma} = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}.$$




```
'full' (each component has its own general covariance matrix),
'tied' (all components share the same general covariance matrix),
'diag' (each component has its own diagonal covariance matrix),
'spherical' (each component has its own single variance).
```

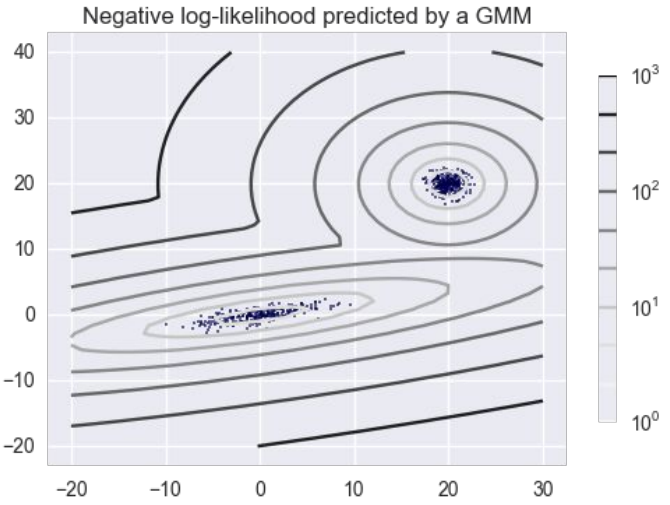
Likelihood function

$$X \sim \pi\mathcal{N}(\mu_1, \sigma_1) + (1 - \pi)\mathcal{N}(\mu_2, \sigma_2)$$



$$p(X|\mu_i, \sigma_i) = \prod_{k=1}^{N} \left(\frac{\pi}{\sqrt{2\pi\sigma_1^2}} e^{-(x_k-\mu_1)^2/\sigma_1^2} + \frac{1-\pi}{\sqrt{2\pi\sigma_2^2}} e^{-(x_k-\mu_2)^2/\sigma_1^2} \right)$$

Likelihood function



Goal:

$$\max_{\mu_i, \sigma_i} \log p(X|\mu_i, \sigma_i)$$

$$p(X|\mu_i, \sigma_i) = \prod_{k=1}^{N} \left(\frac{\pi}{\sqrt{2\pi\sigma_1^2}} e^{-(x_k-\mu_1)^2/\sigma_1^2} + \frac{1-\pi}{\sqrt{2\pi\sigma_2^2}} e^{-(x_k-\mu_2)^2/\sigma_1^2} \right)$$

Gaussian Mixture Models

$$X \sim \pi\mathcal{N}(\mu_1, \sigma_1) + (1-\pi)\mathcal{N}(\mu_2, \sigma_2)$$

$$p(X|\mu_i, \sigma_i) = \prod_{k=1}^{N} \left(\frac{\pi}{\sqrt{2\pi\sigma_1^2}} e^{-(x_k-\mu_1)^2/\sigma_1^2} + \frac{1-\pi}{\sqrt{2\pi\sigma_2^2}} e^{-(x_k-\mu_2)^2/\sigma_1^2} \right)$$

It turns out this is quite difficult to maximize for the parameters, so we try another method. Let's see if we can at least get a lower bound?

The missing link: Jensen's Inequality

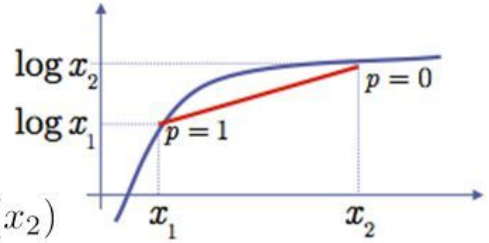
- If f is concave (or convex down): $f(\mathbb{E}\{x\}) \geq \mathbb{E}\{f(x)\}$

- Incredibly important tool for dealing with mixture models.

$$f\left(\sum_i \pi_i p(x|\mu_i, \Sigma_i)\right) \geq \sum_i \pi_i f\left(p(x|\mu_i, \Sigma_i)\right)$$

if f(x) = log(x)

$$\log\left(\sum_i \pi_i p(x|\mu_i, \Sigma_i)\right) \geq \sum_i \pi_i \log\left(p(x|\mu_i, \Sigma_i)\right)$$



$$\log(\pi x_1 + (1-\pi)x_2) \geq \pi \log(x_1) + (1-\pi)\log(x_2)$$

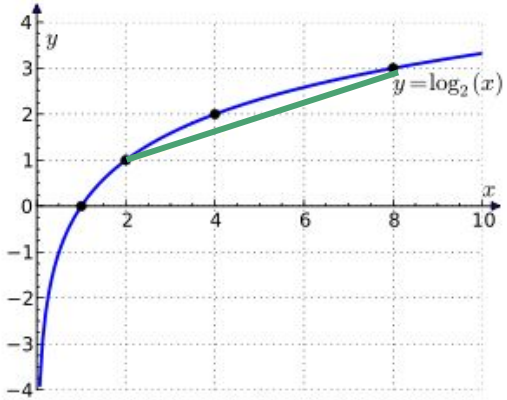
Log special case - easy proof

$$\log\left(\frac{x}{2}+\frac{y}{2}\right)-\frac{1}{2}\log x-\frac{1}{2}\log y = \log\left(\frac{x+y}{2\sqrt{xy}}\right)$$

$$(\sqrt{x}+\sqrt{y})^2 \geq 0$$

$$x+y \geq 2\sqrt{xy}$$

$$\geq \log(1) = 0$$



Why does it work? Concavity!

By the fundamental theorem of Calculus:

$$F(\pi x + (1-\pi)y) - \pi F(x) - (1-\pi)F(y)$$

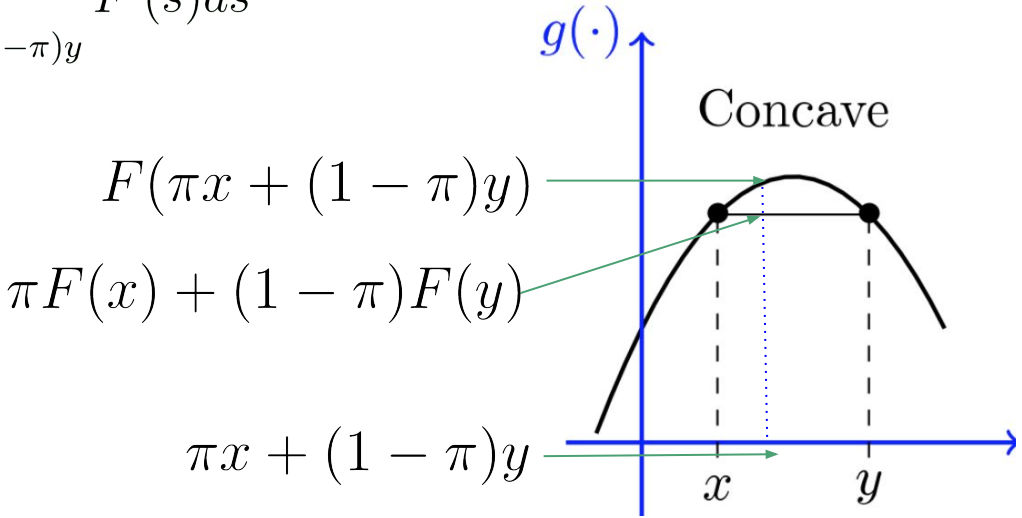
$$= \pi \int_{x}^{\pi x+(1-\pi)y} F'(s)ds - (1-\pi)\int_{\pi x+(1-\pi)y}^{y} F'(s)ds$$

$$= G_1 - G_2 \geq 0$$

Derivative must be **decreasing** as **s increases.**

$$\geq \pi(y-x)^2 \int_0^{\pi} F''(\xi_{s\in[x,y]})dt \geq 0$$

Obtained via **Taylor remainder.**



An optimal lower bound

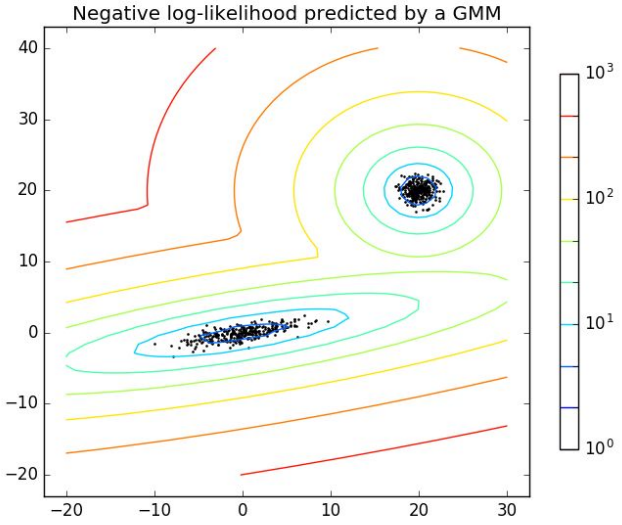
$$L_n(\theta; \mathbf{x}) = \prod_{i=1}^{n} \sum_{j=1}^{2} \tau_j f(\mathbf{x}_i; \mathbf{u}_j, \Sigma_j)$$

$$\log L_n(\theta; \mathbf{x}) = \log \left(\prod_{i=1}^{n} \sum_{j=1}^{2} \tau_j f(\mathbf{x}_i; \mathbf{u}_j, \Sigma_j) \right)$$

$$= \sum_{i=1}^{n} \log \left(\sum_{j=1}^{2} \tau_j f(\mathbf{x}_i; \mathbf{u}_j, \sigma_j) \right)$$

Jensen's inequality

$$\geq \sum_{i=1}^{n} \sum_{j=1}^{2} \tau_j \log f(\mathbf{x}_i; \mathbf{u}_j, \sigma_j)$$



Some example notation

$$p(X|\mu_i,\sigma_i) = \prod_{k=1}^{N}\left(\frac{\pi}{\sqrt{2\pi\sigma_1^2}}e^{-(x_k-\mu_1)^2/\sigma_1^2} + \frac{1-\pi}{\sqrt{2\pi\sigma_2^2}}e^{-(x_k-\mu_2)^2/\sigma_1^2}\right)$$

The full mixed distribution

$$\theta = (\mu_k,\sigma_k)_{k=1,2} = (\mu_1,\mu_2,\sigma_1,\sigma_2)$$

All parameters represented.

$$p(k|\theta) = P(Z=k|\theta) = \pi_*^k = \frac{1}{2}$$

The **real parameters** to be learned.

$$P(\mathbf{x}|k,\theta) = \frac{1}{\sqrt{2\pi\sigma_k^2}}e^{-(\mathbf{x}-\mu_k)^2/\sigma_k^2}$$

The distribution for a given cluster k.

$$P(k|\mathbf{x},\theta) = P(Z=k|\mathbf{x},\theta) = \frac{\pi_{\mathbf{x}}^k e^{-(\mathbf{x}-\theta_k)^2/\sigma_k^2}}{\pi_{\mathbf{x}}^1 e^{-(\mathbf{x}-\theta_1)^2/\sigma_1^2} + \pi_{\mathbf{x}}^2 e^{-(\mathbf{x}-\theta_2)^2/\sigma_2^2}}$$

The probability of being in a given cluster k .

An optimal lower bound - general

$$\log L(\theta;\mathbf{x}) - \log L(\theta_n|\mathbf{x}) = \log P(\mathbf{x}|\theta) - \log P(\mathbf{x}|\theta_n)$$

$$= \log \sum_k P(\mathbf{x}|\theta,k)P(k|\theta) - \log P(\mathbf{x}|\theta_n)$$

$\theta = (\mu_1, \sigma_1, \mu_2, \sigma_2)$

$k = 1, 2$

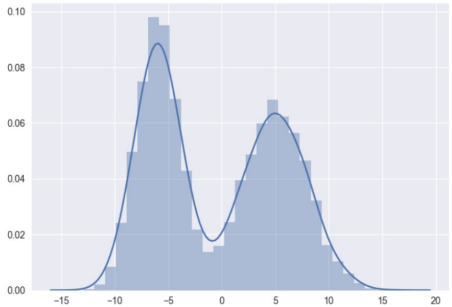
$$= \log \sum_k \frac{P(\mathbf{x}|\theta,k)P(k|\theta)}{P(\mathbf{x}|\theta_n)}$$

Want the expectation w.r.t variables we know

$$= \log \sum_k \frac{P(\mathbf{x}|\theta,k)P(k|\theta)P(k|\mathbf{x},\theta_n)}{P(\mathbf{x}|\theta_n)P(k|\mathbf{x},\theta_n)}$$

$$\geq \log \sum_k P(k|\mathbf{x},\theta_n)\frac{P(\mathbf{x}|\theta,k)P(k|\theta)}{P(\mathbf{x}|\theta_n)P(k|\mathbf{x},\theta_n)}$$

Jensen inequality



An optimal lower bound - general

$$\geq \log \sum_k P(k|\mathbf{x}, \theta_n) \frac{P(\mathbf{x}|\theta, k)P(k|\theta)}{P(\mathbf{x}|\theta_n)P(k|\mathbf{x}, \theta_n)}$$

$$= \mathbb{E}_{k|\mathbf{x},\theta_n} \log \frac{P(\mathbf{x}|\theta, k)P(k|\theta)}{P(\mathbf{x}|\theta_n)P(k|\mathbf{x}, \theta_n)}$$

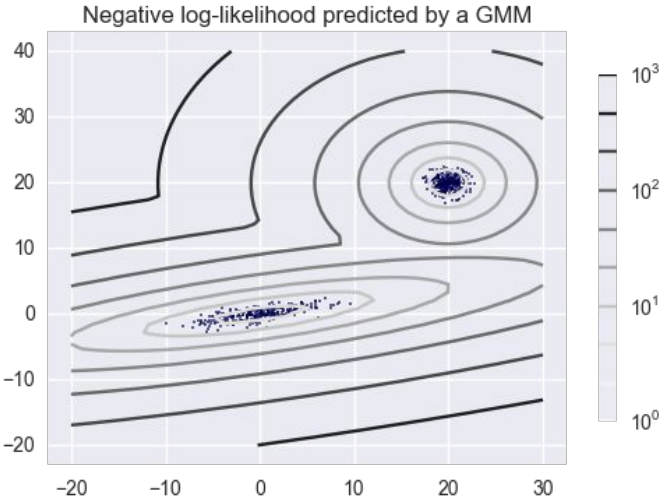
Properties of lower bound

- $\Delta(\theta_n, \theta_n) = 0$
- Always non-decreasing.

Expectation

$$=: \Delta(\theta, \theta_n) \longrightarrow \theta_{n+1} = \text{argmax}_\theta \Delta(\theta, \theta_n)$$ **Maximization**

Gaussian Case



$$(\boldsymbol{\mu}_1^{(t+1)}, \Sigma_1^{(t+1)}) = \arg\max_{\boldsymbol{\mu}_1, \Sigma_1} Q(\theta|\theta^{(t)})$$
$$= \arg\max_{\boldsymbol{\mu}_1, \Sigma_1} \sum_{i=1}^{n} T_{1,i}^{(t)} \left\{ -\tfrac{1}{2} \log|\Sigma_1| - \tfrac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_1)^\top \Sigma_1^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) \right\}$$

$$\boldsymbol{\tau}^{(t+1)} = \arg\max_{\boldsymbol{\tau}} Q(\theta|\theta^{(t)})$$
$$= \arg\max_{\boldsymbol{\tau}} \left\{ \left[\sum_{i=1}^{n} T_{1,i}^{(t)} \right] \log \tau_1 + \left[\sum_{i=1}^{n} T_{2,i}^{(t)} \right] \log \tau_2 \right\}$$

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{j,i}^{(t)}}{\sum_{i=1}^{n} (T_{1,i}^{(t)} + T_{2,i}^{(t)})} = \frac{1}{n} \sum_{i=1}^{n} T_{j,i}^{(t)}$$

Solutions to minimization problem

$$\boldsymbol{\mu}_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{1,i}^{(t)}} \qquad \boldsymbol{\mu}_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{2,i}^{(t)}}$$

$$\Sigma_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})^\top}{\sum_{i=1}^{n} T_{1,i}^{(t)}}$$

$$\Sigma_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})^\top}{\sum_{i=1}^{n} T_{2,i}^{(t)}}$$

These are just the empirical means and variances of the data!

Termination Step

$$E_{Z|\theta^{(t)},\mathbf{x}}[\log L(\theta^{(t)};\mathbf{x},\mathbf{Z})] \leq E_{Z|\theta^{(t-1)},\mathbf{x}}[\log L(\theta^{(t-1)};\mathbf{x},\mathbf{Z})] + \epsilon$$

Choose ϵ and continue until the inequality is satisfied.

Log-likelihood landscape

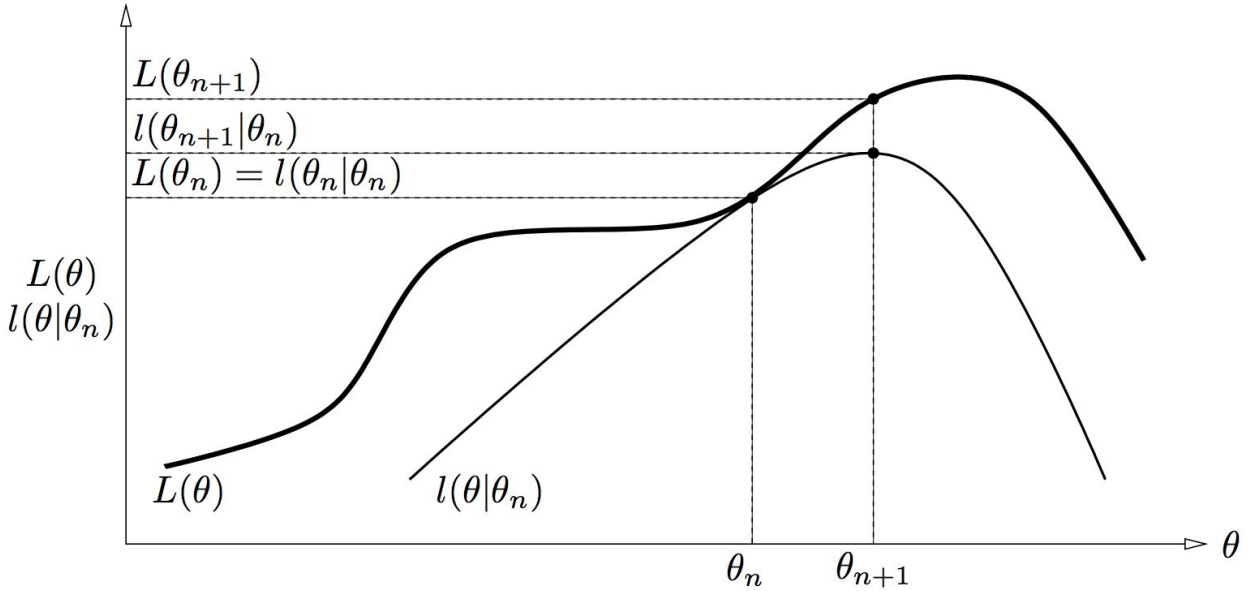
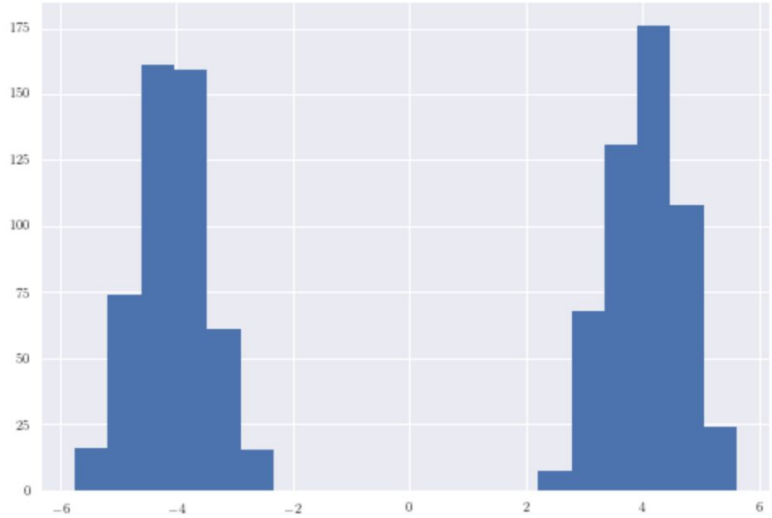


Figure 2: Graphical interpretation of a single iteration of the EM algorithm: The function $L(\theta|\theta_n)$ is upper-bounded by the likelihood function $L(\theta)$. The functions are equal at $\theta = \theta_n$. The EM algorithm chooses θ_{n+1} as the value of θ for which $l(\theta|\theta_n)$ is a maximum. Since $L(\theta) \geq l(\theta|\theta_n)$ increasing $l(\theta|\theta_n)$ ensures that the value of the likelihood function $L(\theta)$ is increased at each step.

In [352]:

```
#In 1-D
# True parameter values
mu_true = [-4, 4]
sigma_true = [0.6, 0.6]
lambda_true = .5
n = 1000

# Simulate from each distribution according to mixing proportion psi
z = np.random.binomial(1, lambda_true, n)
x = np.array([np.random.normal(mu_true[i], sigma_true[i]) for i in z])

plt.hist(x, bins=20)
plt.show()
```



Let's generate data from a mixture of two Gaussians and see what happens when we start close to the saddle point

Example of two Gaussians

$$\sigma_1 = \sigma_2 = 1$$
$$\mu_2 = -4$$
$$\mu_1 = 4$$

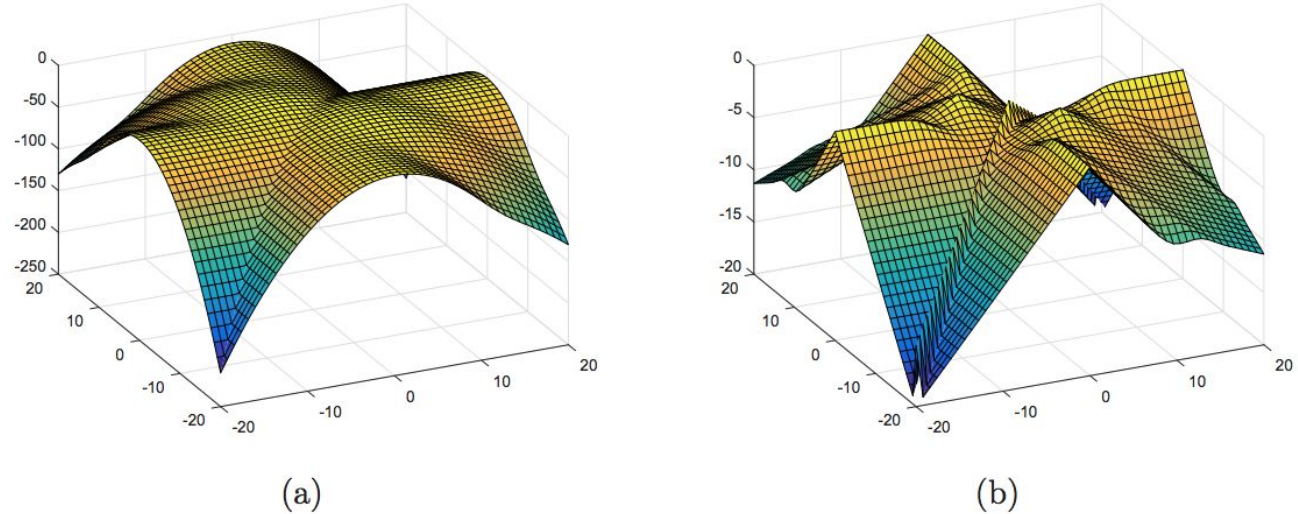
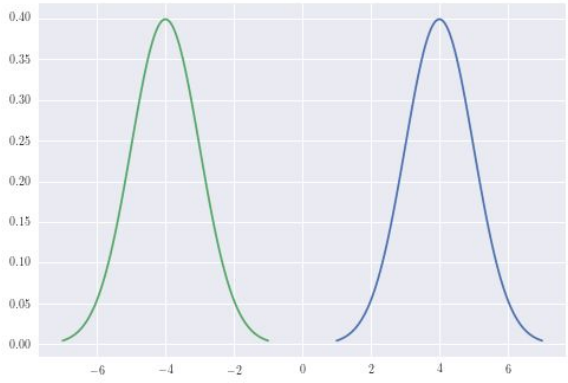
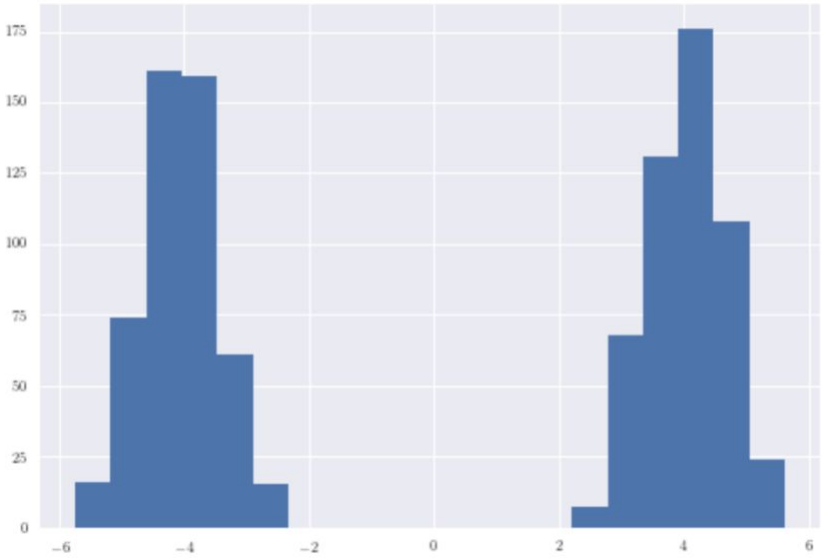


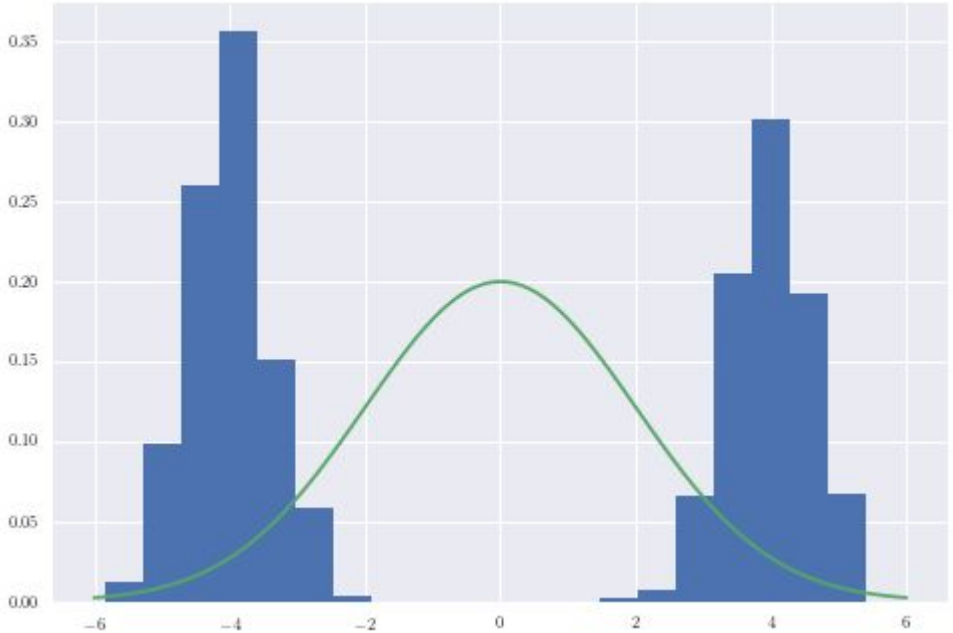
Figure 1. Illustration of the likelihood and gradient maps for a two-component Gaussian mixture. (a) Plot of population log-likehood map $\boldsymbol{\mu} \mapsto \mathcal{L}(\boldsymbol{\mu})$. (b) Plot of the negative Euclidean norm of the gradient map $\boldsymbol{\mu} \mapsto -\|\nabla \mathcal{L}(\boldsymbol{\mu})\|_2$.



Global maximum is at (-4,4) but saddle at 0

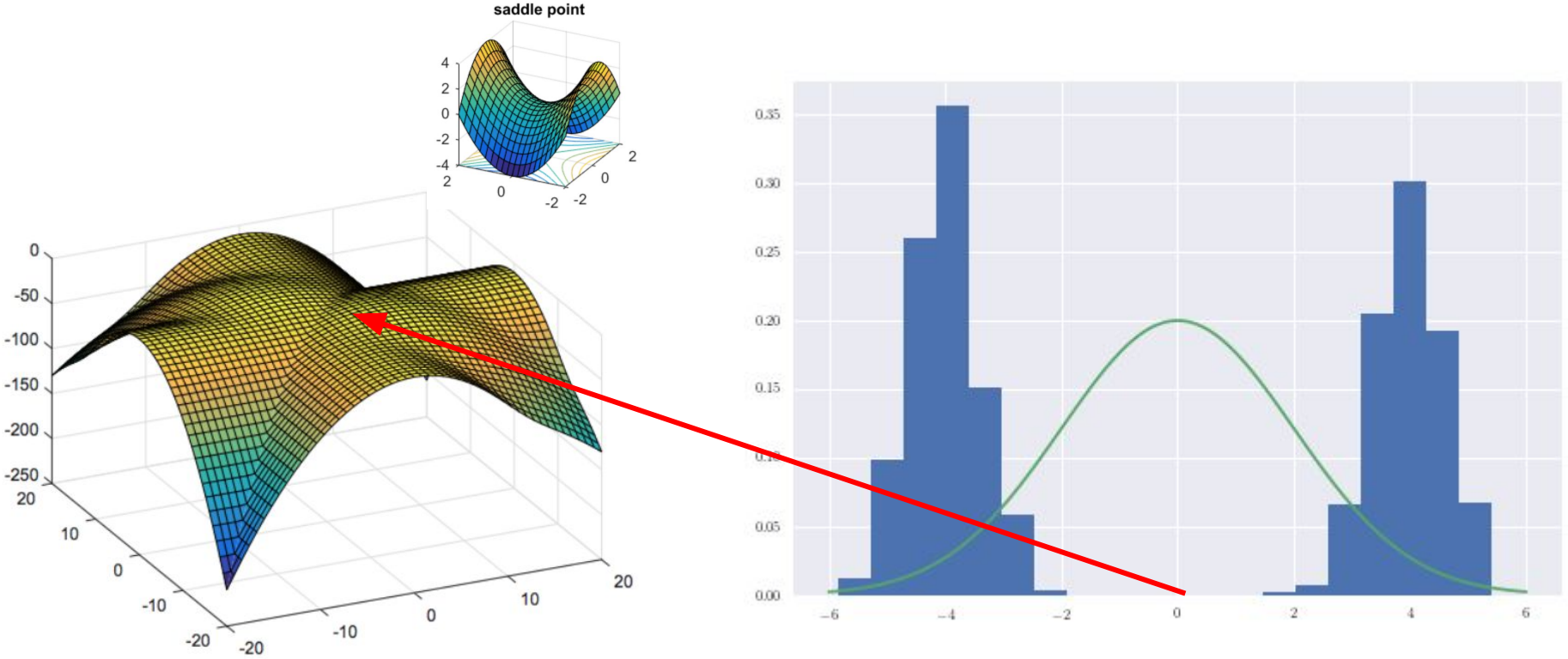
Converging to the saddle point





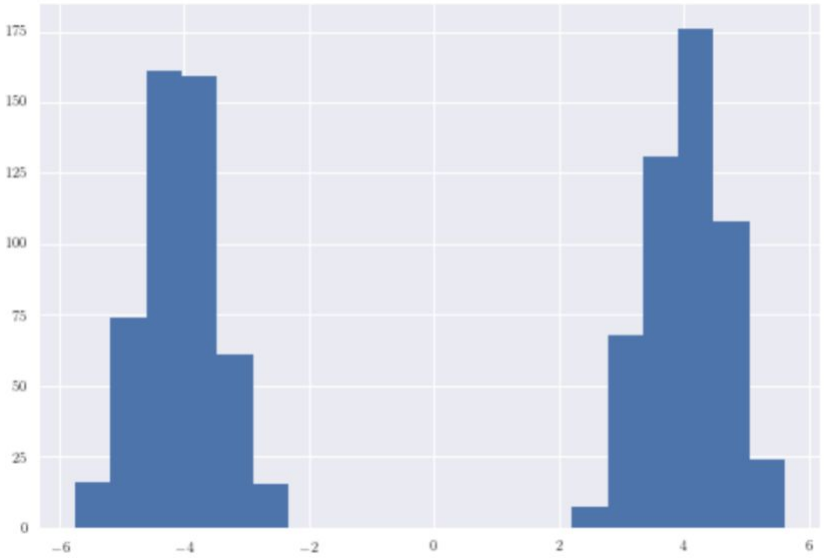
Initial conditions chosen close to saddle point

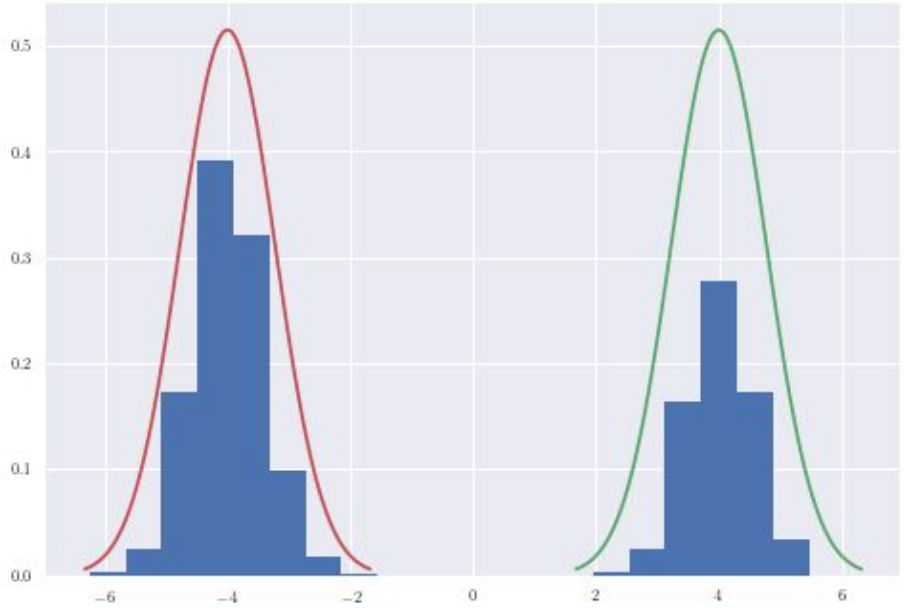
Converging to the saddle point



Initial conditions chosen close to saddle point

Converging to the right values





- Initial conditions **chosen randomly.**
- If we weren't careful though, we could have easily converged to our saddle point if we didn't try other initial values!

Gaussian Case and K means

$$T_{j,i}^{(t)} := \mathrm{P}(Z_i = j | X_i = \mathbf{x}_i; \theta^{(t)}) = \frac{\tau_j^{(t)} \ f(\mathbf{x}_i; \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)})}{\tau_1^{(t)} \ f(\mathbf{x}_i; \boldsymbol{\mu}_1^{(t)}, \Sigma_1^{(t)}) + \tau_2^{(t)} \ f(\mathbf{x}_i; \boldsymbol{\mu}_2^{(t)}, \Sigma_2^{(t)})}$$

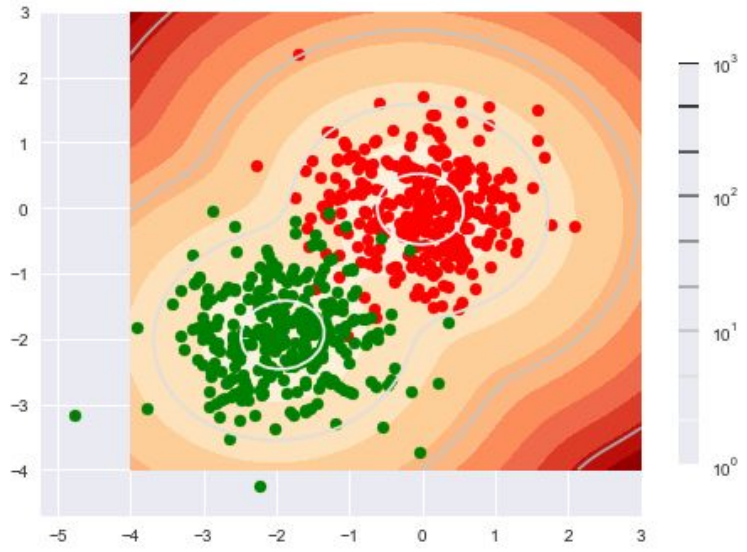
K means special case

- Assume variances in each cluster are equal and very small (no cloud effect).
- Then each point belongs to one cluster.

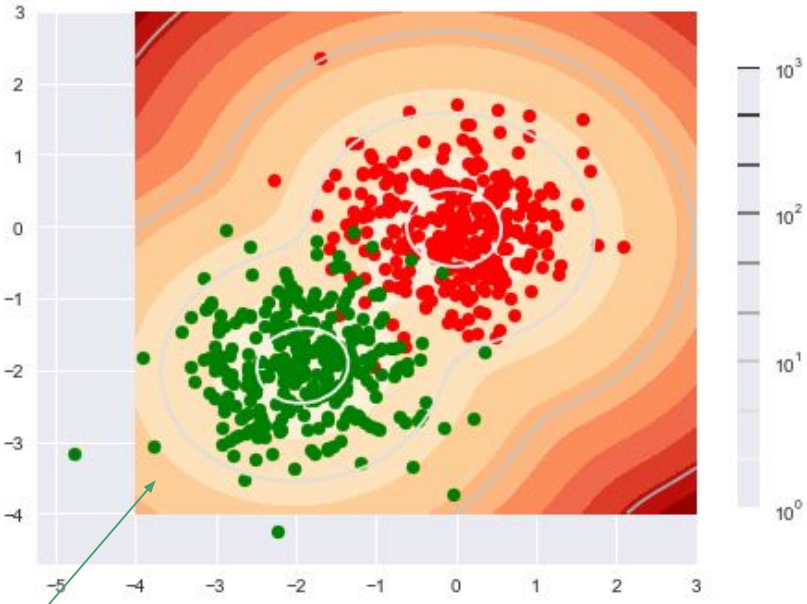
$$\lim_{\epsilon \to 0} \frac{\tau_i e^{-(x_i - \mu_j)^2/\epsilon}}{\tau_1 e^{-(x_i - \mu_1)^2/\epsilon} + \tau_2 e^{-(x_i - \mu_2)^2/\epsilon}} = \delta(Z = j)$$

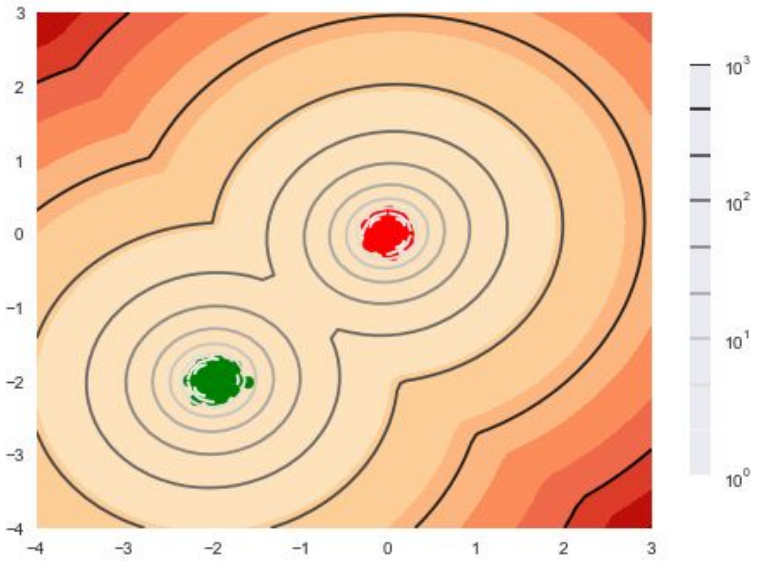
$$L(\theta|\mathbf{x}) = \sum_z \log P(\mathbf{x}|\theta, z) P(z|\mathbf{x}, \theta)$$

$$= \log P(\mathbf{x}|\theta, z = j) \sim \sum_{\mathbf{x}_i \in K_j} (\mathbf{x}_i - \mu_j)^2$$



Shrinking variance leads to K means



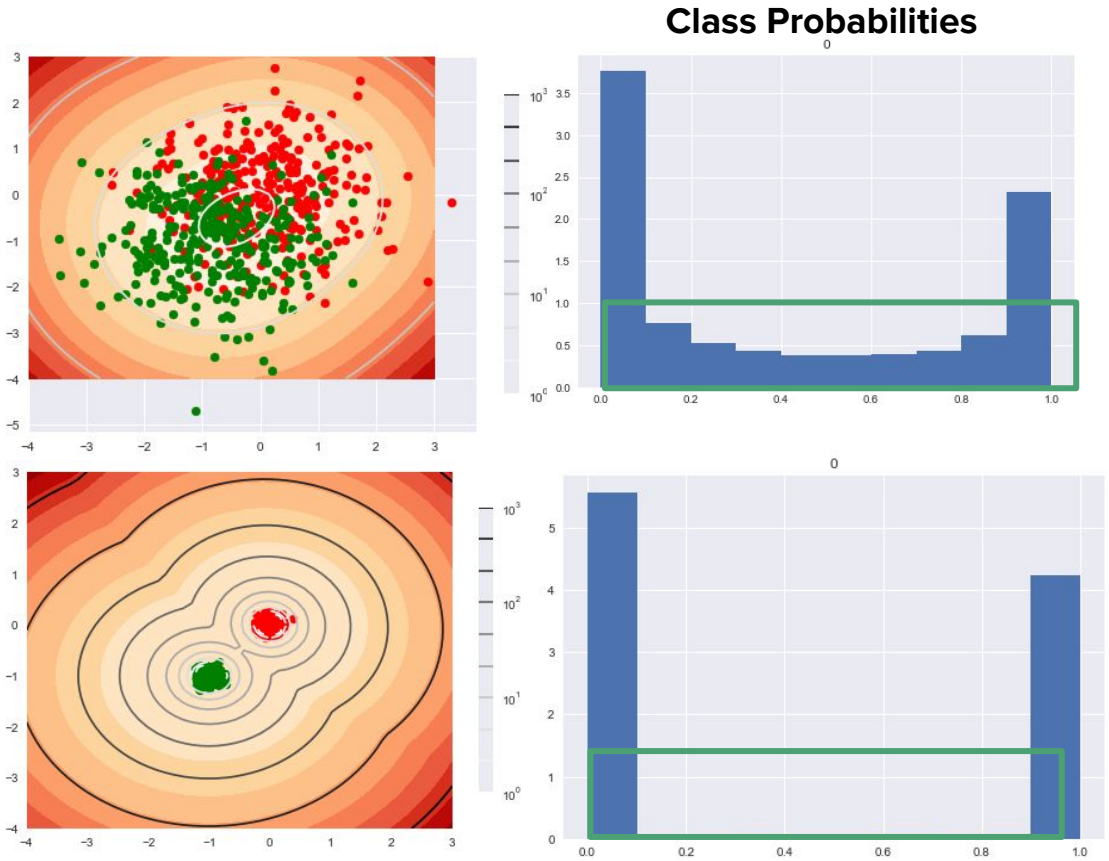


Darker red = lower likelihood.

Variance shrinking ->

As the variance shrinks, the probability of being in a cluster tends towards to 1 which is K means.

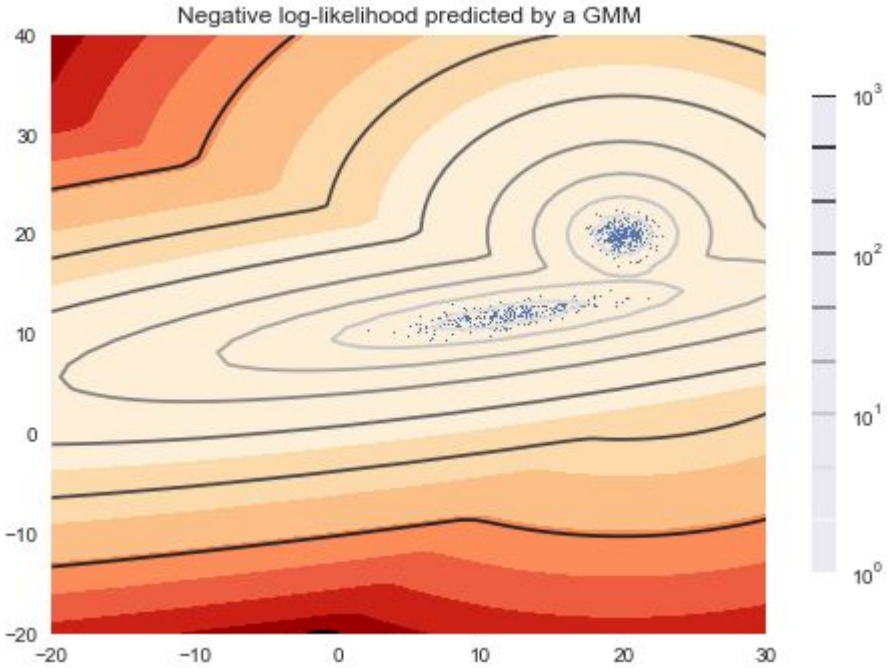
K means vs GMM

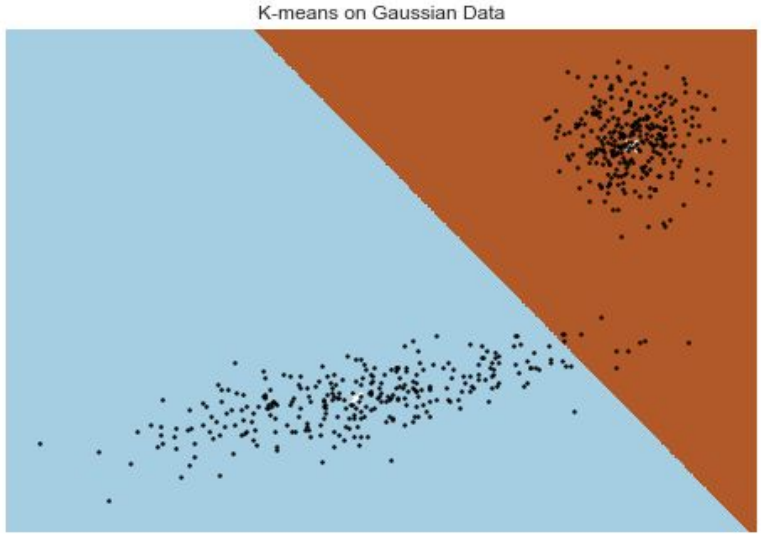


Gaussian Mixture

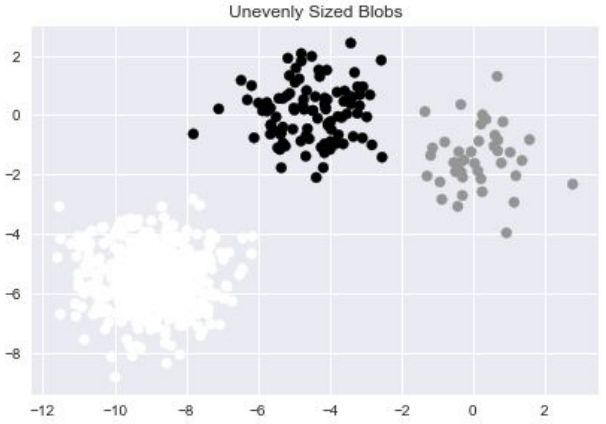
Low variance Gaussian Mixture - results in pure classes.

Where K means assumptions fail



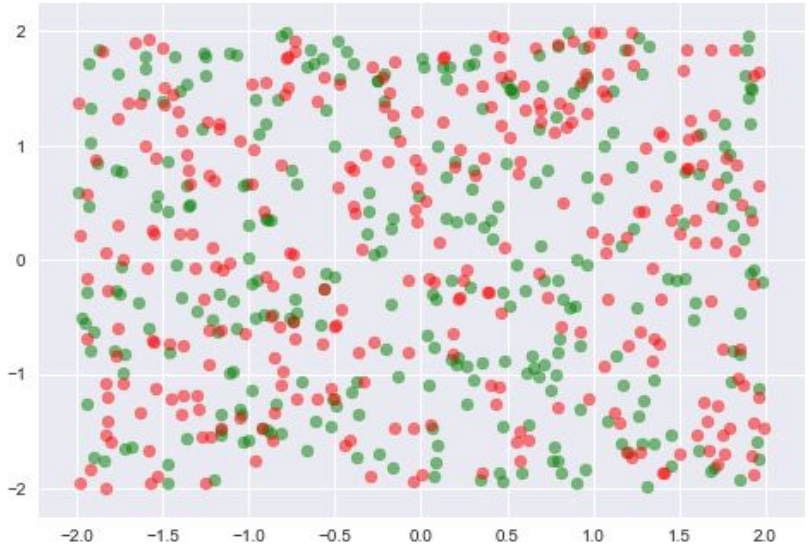


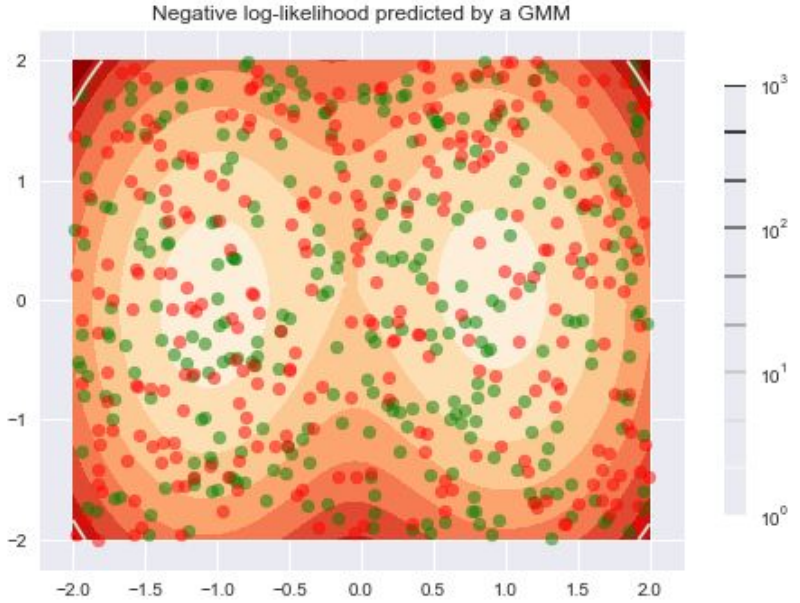
Same issue here with GMM



In [197]:
```
# fit a Gaussian Mixture Model with two components
# Unevenly sized blobs
# Different variance
for n in range(10,50,10):
    X,y = make_blobs(n_samples=n_samples,
                                    cluster_std=[1, 1, 1],
                                    random_state=random_state)


    # Unevenly sized blobs
    X_filtered = np.vstack((X[y == 0][:500], X[y == 1][:100], X[y == 2][0:n]-2))

    plt.figure(figsize=(15,10))

    clf = mixture.GaussianMixture(n_components=3, covariance_type='full')
    clf.fit(X_filtered)
    y_pred = clf.predict(X_filtered)
    plt.subplot(224)
    plt.scatter(X_filtered[:, 0], X_filtered[:, 1], c=y_pred)
    plt.title("Unevenly Sized Blobs")
    plt.show()
```

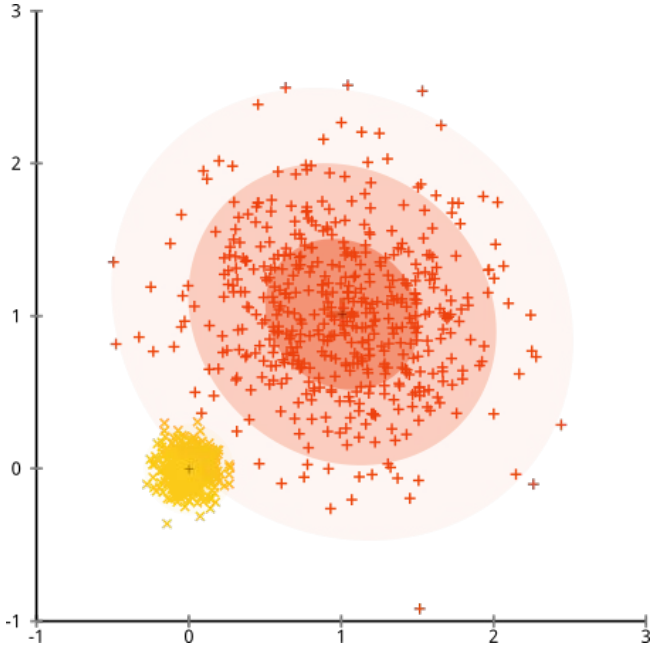
Same issue with GMM - Uniform data





When to use it?

- From what we've seen, the best use case for GMM is **when you don't expect your data to have constant variance in each cluster, but it generally follows a Gaussian with comparable number of points.**
- In many other situations, this still fails as we've seen.



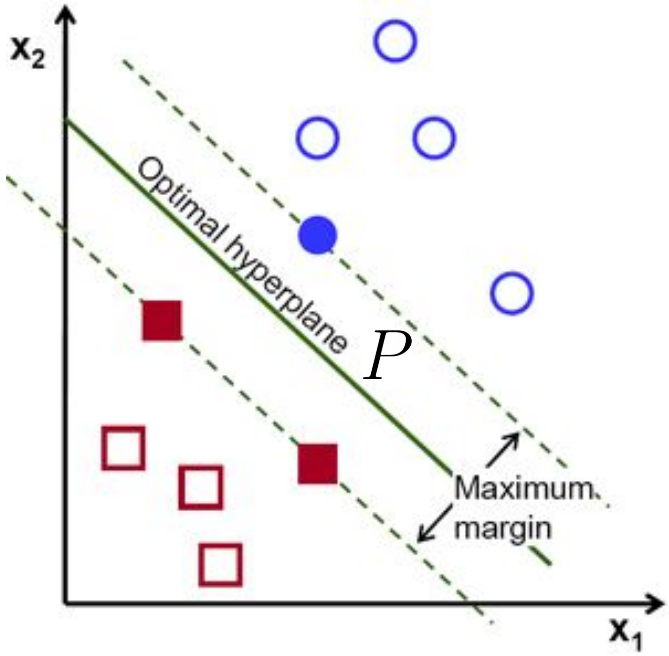
Support Vector Machines

Support Vector Machines

$$f(x) = \omega_0 + \omega^T \cdot x$$

$$|\omega_0 + \omega^T \cdot x| = 1$$

$$d(x, P) = \frac{|\omega_0 + \omega^T \cdot x|}{\|\omega\|} = \|\omega\|^{-1}$$



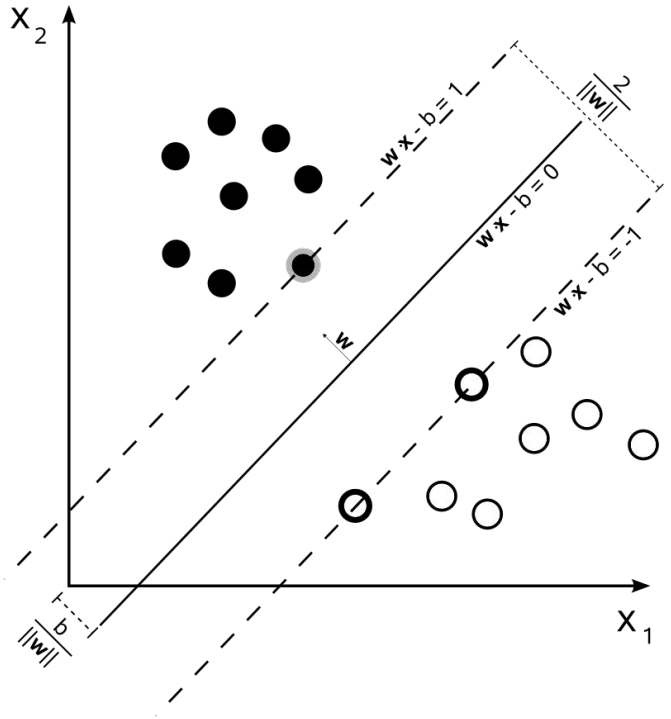
Goal: Create a maximally separating hyperplane between two classes, ie. choose P as to maximize d. Thus we which to maximize the norm of w.

Support Vector Machines

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n}\|w\|^2$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1$$

- Note that maximizing $\|\omega\|^{-1}$ is the same as minimizing $\|\omega\|$
- This is a convex linear program and thus has a global unique solution.
- But we have just one problem! The constraint is probably not satisfied.



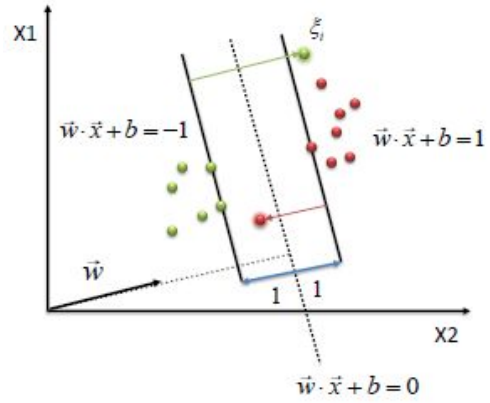
$$d(x, P) = \frac{|\omega_0 + \omega^T \cdot x|}{\|\omega\|} = \|\omega\|^{-1}$$

Support Vector Machines With Slack

$$\text{minimize } \|w\|^2 + C\frac{1}{n}\sum_{i=1}^{n}\zeta_i$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i.$$

- Note that maximizing $\|\omega\|^{-1}$ is the same as minimizing $\|\omega\|$
- ζ_i is called the ***slack variable***. It accounts for the fact that data will never have perfect separation, so it relaxes the above constraint.
- C is the ***regularization strength***. If C = 0, the above problem doesn't care about getting the right answer at all.



slack variable:

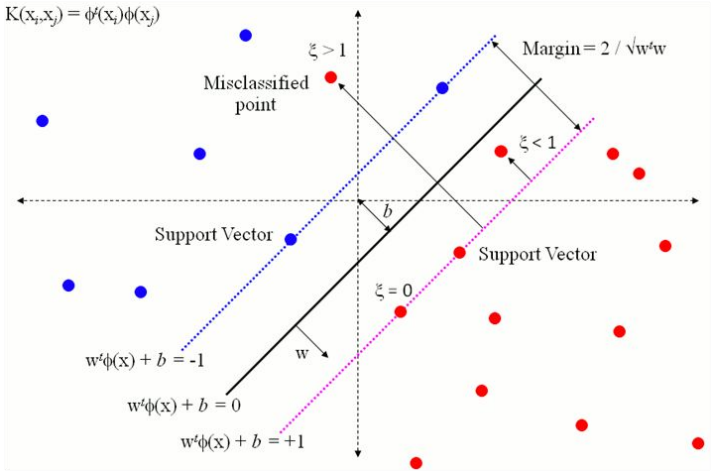
ξ_i

Allow some instances to fall off the margin, but penalize them

Support Vector Machines With Slack

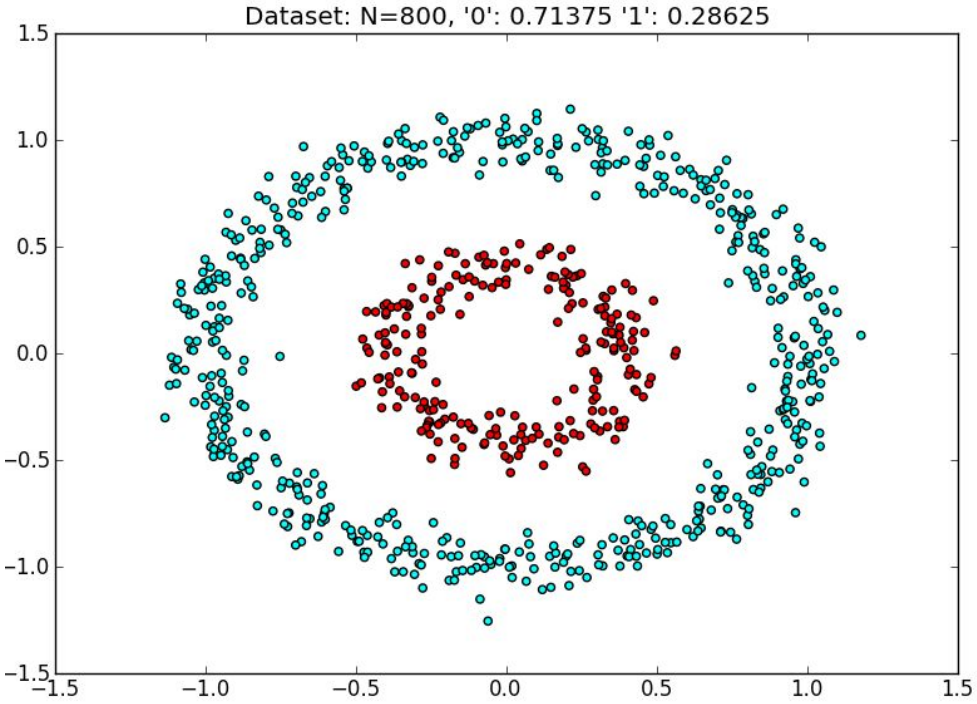
$$\text{minimize } \|w\|^2 + C\frac{1}{n}\sum_{i=1}^{n}\zeta_i$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, \text{ for all } i.$$



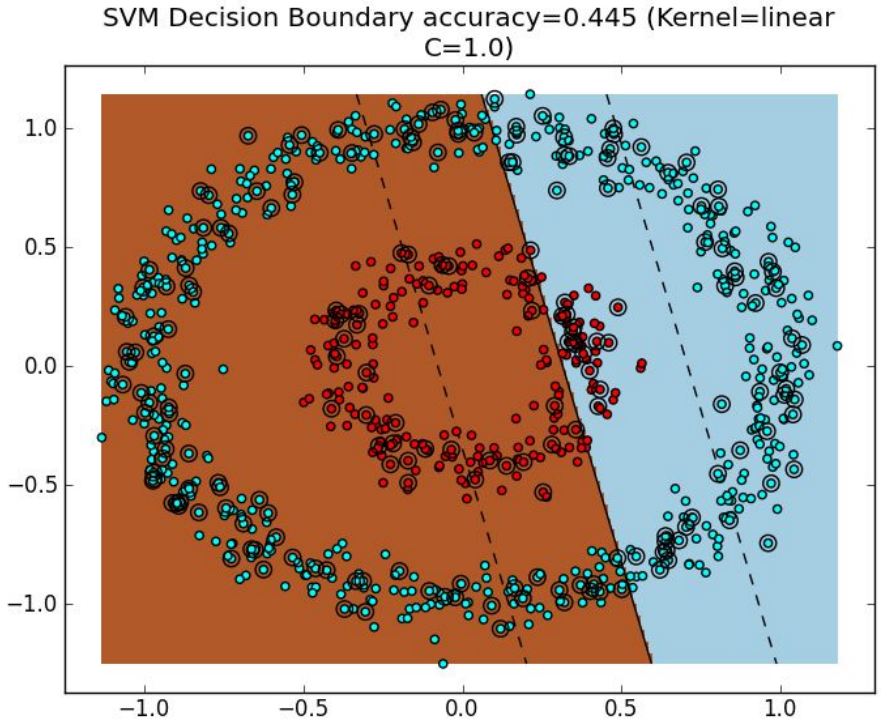
- If $0 \leq \zeta_i \leq 1$ then the point is on the right side, but not as far from the plane as we originally wanted.
- If $\zeta_i > 1$ then the point is incorrectly classified. It will be penalized more heavily.
- How much you want to penalize wrong answers depends on C.

Nonlinear example



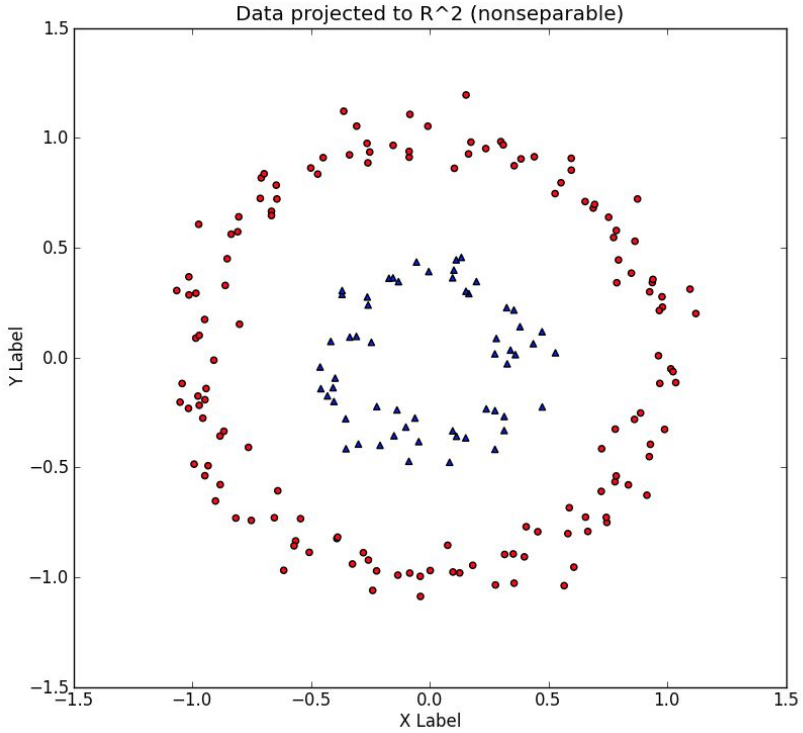
- In this example, a separating hyperplane won't work. So we're out of luck right?

Nonlinear example



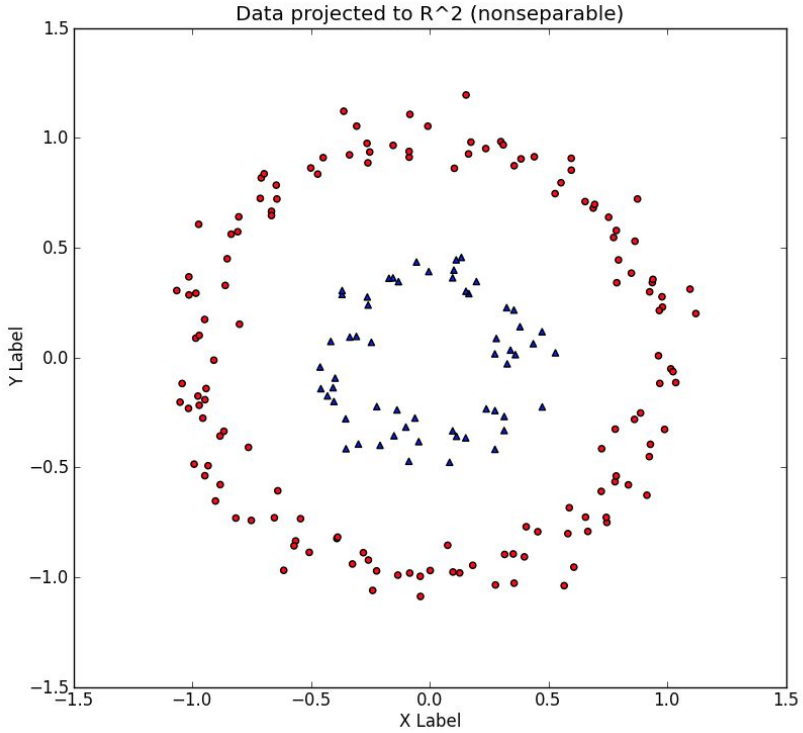
- In this example, a separating hyperplane won't work. So we're out of luck right?

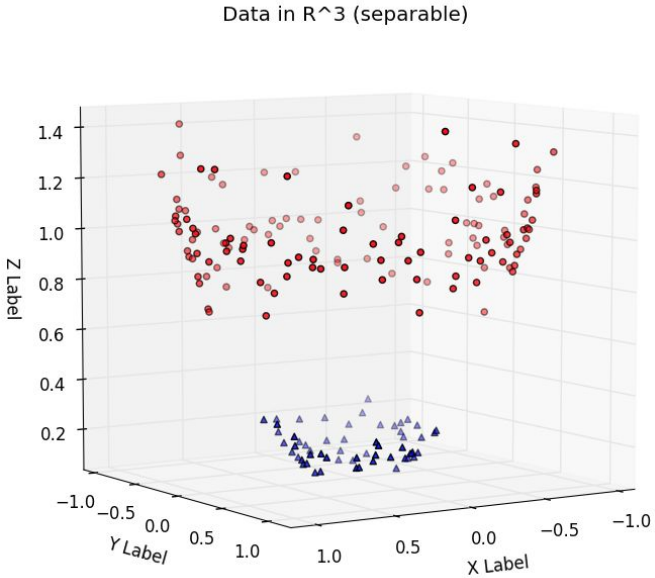
Kernel Trick



- What if we could learn a nonlinear decision boundary?
- What about mapping this to a higher dimensional space?

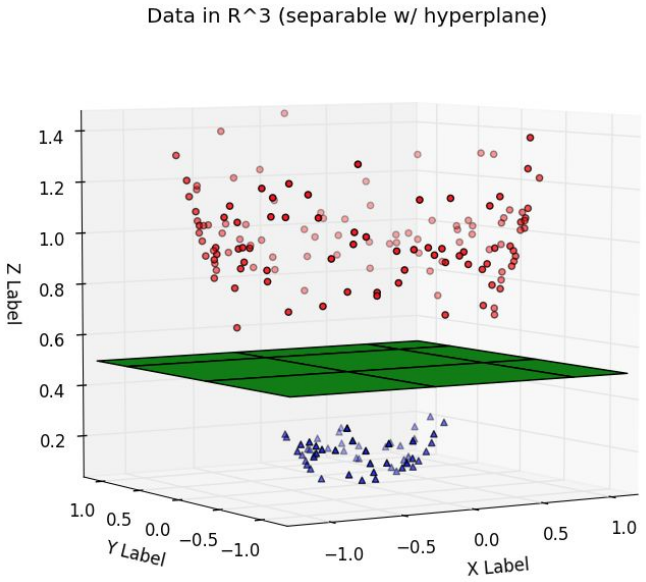
Kernel Trick

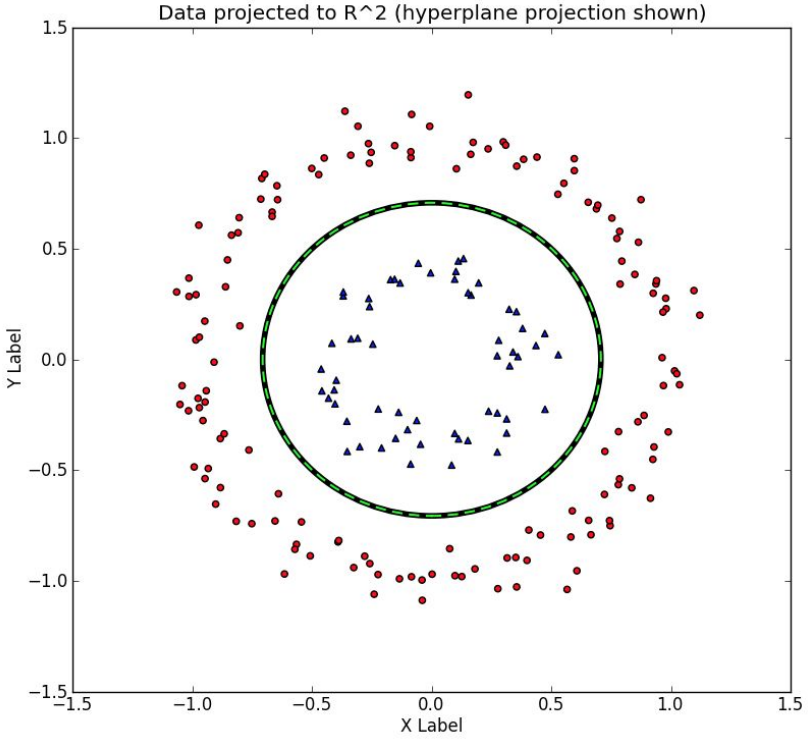




$$[x, y, x^2 + y^2]$$

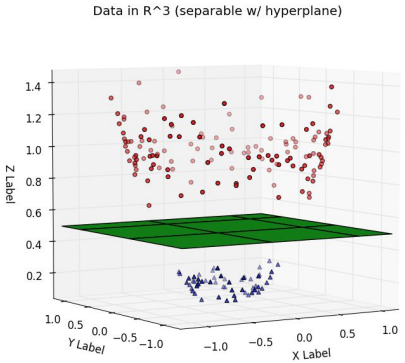
A solution in dimension 3

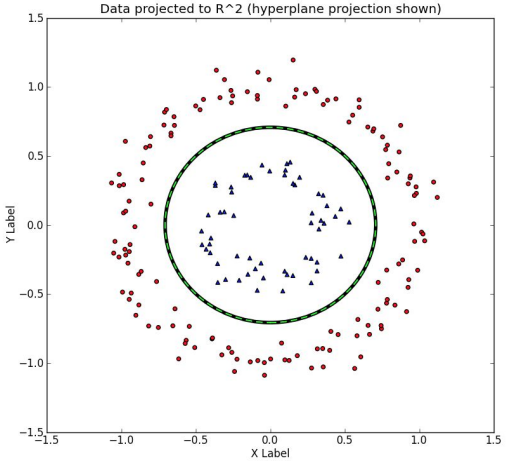




$$\phi((x, y)) = (x, y, x^2 + y^2)$$

A solution in dimension 3





$$\phi((x, y)) = (x, y, x^2 + y^2)$$

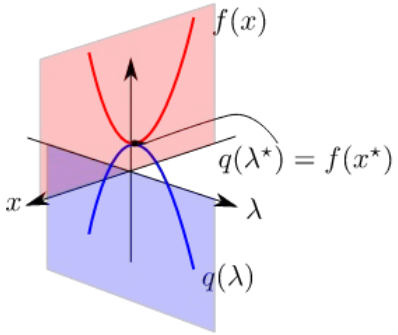
- One maps to a higher dimensional space where linear separation can be possible. Then find a solution, map back to lower dimensional space.
- But the trick is, we don't actually have to do this!
- How do we avoid mapping to higher dimensions to solve our problem?

Dual Problem

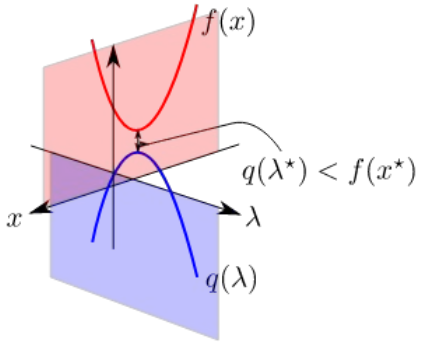
Original:

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\|w\|^2$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1$$



strong duality



weak duality

.

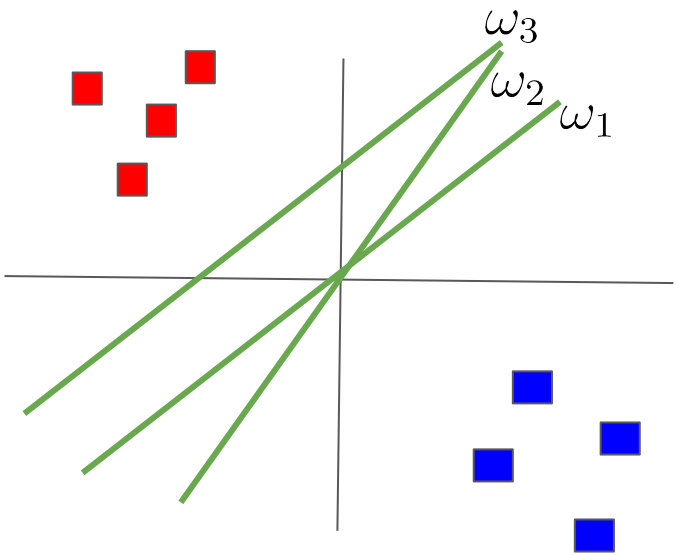
Dual Problem:

$$\text{maximize}_\alpha \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$\text{s.t.} \quad \forall i : \alpha_i \geq 0 \wedge \sum_{i=1}^{n} y_i \alpha_i = 0 \qquad w = \sum_i \alpha_i y_i \mathbf{x_i}$$

Why do we care about this at all?

Convex Duality



$$\min_{\omega} \frac{1}{2}\|\omega\|^2$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1$$

$$= \min_{\omega} \max_{\lambda_i > 0} \frac{1}{2}\|\omega\|^2 - \sum_i \lambda_i (y_i \omega \cdot x_i - 1)$$

- If the constraint isn't satisfied, the max over lambda will blow up, contradicting the minimum we have.
- There is no range of w which won't "need" this constraint, due to rotations.

Dual Problem

Original:

$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n} \|w\|^2$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1$$

Lagrangian:

$$\mathcal{L}(\omega, \omega_0, \alpha) := \sum_{k=1}^{d} \omega_k^2 - \sum_{i=1}^{N} \alpha_i [y_i(\omega \cdot \mathbf{x}_i + \omega_0) - 1]$$

$$\nabla_\omega \mathcal{L} = 0$$

$$\nabla_{\omega_0} \mathcal{L} = 0$$

We want to minimize the Lagrangian over the original variables, and maximize it over the Lagrange multipliers (since we expect the term in brackets to be positive).

Dual Problem

Lagrangian:

$$\mathcal{L}(\omega, \omega_0, \alpha) := \frac{1}{2}\sum_{k=1}^{d} \omega_k^2 - \sum_{i=1}^{N} \alpha_i [y_i(\omega \cdot \mathbf{x}_i + \omega_0) - 1]$$

$$\nabla_\omega \mathcal{L} = 0 \longrightarrow w = \sum_i \alpha_i y_i \mathbf{x_i} \qquad (1)$$

$$\nabla_{\omega_0} \mathcal{L} = 0 \longrightarrow \sum_i \alpha_i y_i = 0 \qquad (2)$$

Substituting (1) and (2) into the Lagrangian , we get, maximizing over alpha:

$$\text{maximize}_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

Dual Problem - Who Cares?

Dual Problem:

$$\text{maximize}_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$\text{s.t.} \quad \forall i : \alpha_i \geq 0 \wedge \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$w = \sum_i \alpha_i y_i \mathbf{x_i}$$

This gives us an explicit expression for w.

Question: Why is this useful?

Dual Problem - Written in terms of Inner Products

Dual Problem:

$$\text{maximize}_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$\text{s.t.} \quad \forall i : \alpha_i \geq 0 \wedge \sum_{i=1}^{n} y_i \alpha_i = 0$$

Let's plug this in and see what happens:

$$w = \sum_i \alpha_i y_i \mathbf{x_i} \qquad \omega^T \cdot \mathbf{x} + \omega_0 = \sum_i \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle_V + \omega_0$$

$$\tilde{w} = \sum_i \alpha_i y_i \phi(\mathbf{x_i}) \qquad \omega^T \cdot \phi(\mathbf{x}) + \omega_0 = \sum_i \alpha_i y_i \langle \phi(\mathbf{x_i}), \phi(\mathbf{x}) \rangle_V + \omega_0$$

Dual Problem

$$\omega^T \cdot \phi(\mathbf{x}) + \omega_0 = \sum_i \alpha_i y_i \langle \phi(\mathbf{x_i}), \phi(\mathbf{x}) \rangle_V + \omega_0$$

$$\omega^T \cdot \mathbf{x} + \omega_0 = \sum_i \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle_V + \omega_0$$

Recall we defined K as:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_1) \rangle_V$$

This is only in terms of inner products! Hmmm! But wait, we defined K as a Kernel on our original variables.

Therefore:

$$\omega^T \cdot \phi(\mathbf{x}) + \omega_0 = \sum_i \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + \omega_0$$

The Kernel Trick

$$K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_1) \rangle_V$$

$$\omega^T \cdot \phi(\mathbf{x}) + \omega_0 = \sum_i \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + \omega_0$$

- It now remains to optimize over $\{\alpha_i\}$.
- But note that we are now solving the problem with the same number of variables, but in a higher dimension!
- This is known as the **kernel trick**. It holds as long as K can be realized as an inner product on some larger space (maybe infinite dimensional!). The precise conditions are in Mercer's Theorem.

Examples of Kernels

- Linear $\mathbf{x} \cdot \mathbf{v}$
- Polynomial $(r + \mathbf{x} \cdot \mathbf{v})^d \text{ for } r \geq 0, d \geq 0$
- Radial Basis Function $\exp(-\gamma \|\mathbf{x} - \mathbf{v}\|^2), \text{ for } \gamma > 0$
- Gaussian $\exp(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{v}\|^2)$

We only need the dot products!

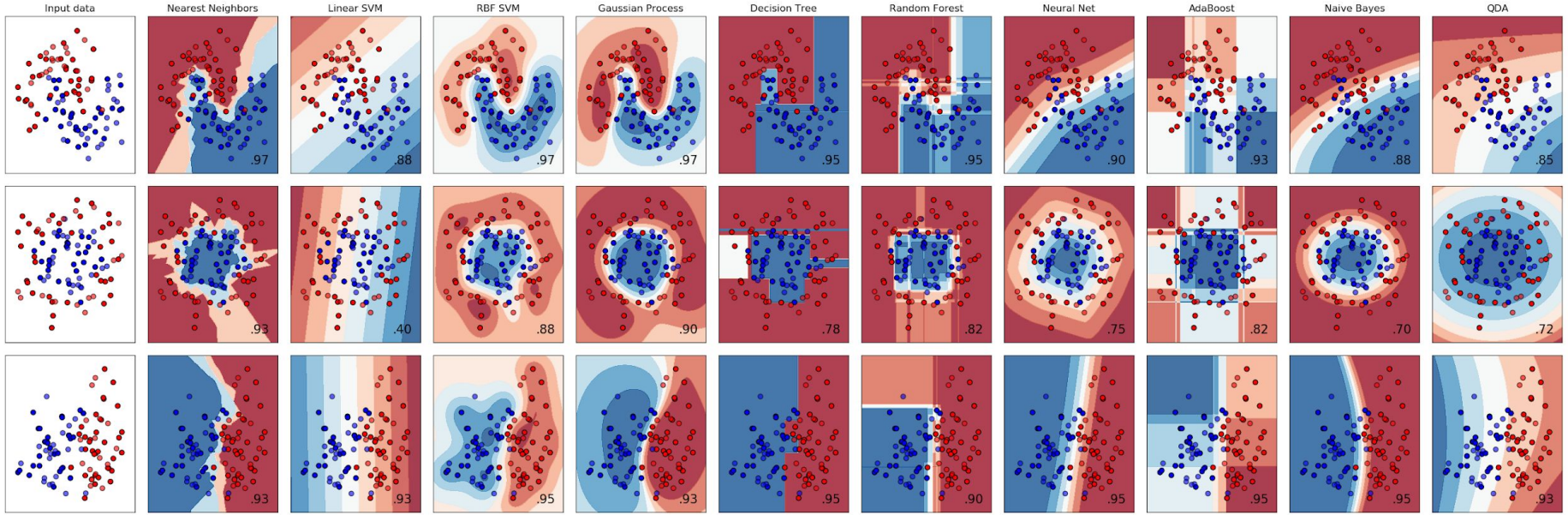
It turns out that the SVM has no need to explicitly work in the higher-dimensional space at training or testing time. One can show that during training, the optimization problem only uses the training examples to compute **pairwise** dot products.

Why is this significant? It turns out that there exist functions that, given two vectors v and w in , they implicitly computes the dot product between v and w in a higher-dimensional data **without explicitly transforming v and w to higher dimensions** . Such functions are called **kernel** functions,

<u>The implications are:</u>

1. By using a kernel , we can implicitly transform datasets to a higher-dimensional using no extra memory, and with a minimal effect on computation time.
2.
    - The only effect on computation is the extra time required to compute . Depending on , this can be minimal.
    -
3. By virtue of (1), we can efficiently learn nonlinear decision boundaries for SVMs simply by **replacing all dot products in the SVM computation with our kernel, if it satisfies certain properties !**

Examples


Input data
Nearest Neighbors
Linear SVM
RBF SVM
Gaussian Process
Decision Tree
Random Forest
Neural Net
AdaBoost
Naive Bayes
QDA
.97
.88
.97
.97
.95
.95
.90
.93
.88
.85
.93
.40
.88
.90
.78
.82
.75
.82
.70
.72
.93
.93
.95
.93
.95
.90
.95
.95
.95
.93

What SVM special? Linear Regression

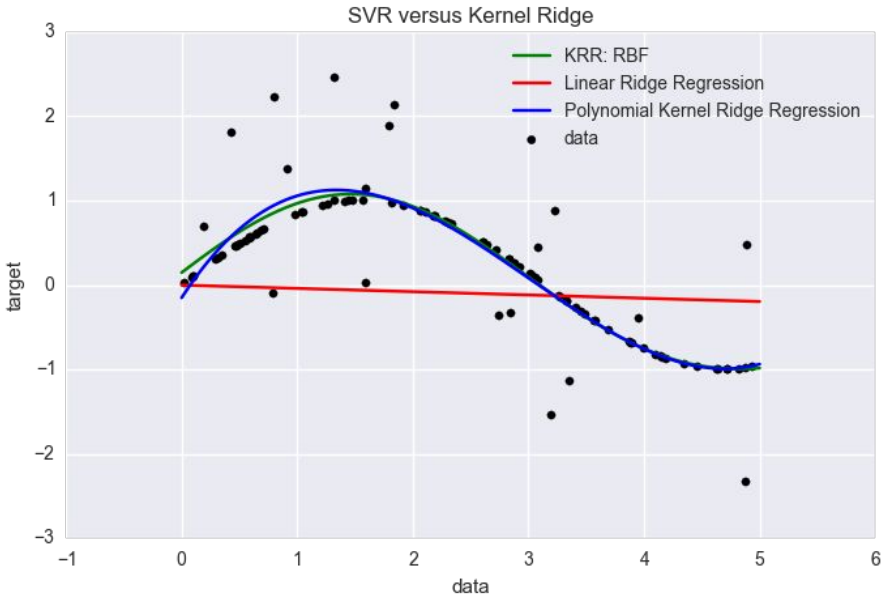
Original OLS:

Minimize: $\frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{w}\cdot\mathbf{x_i})^2 \longrightarrow \mathbf{w} = (X^TX)^{-1}X^T\mathbf{y}$

$$K_{ij} = K(\mathbf{x_i}, \mathbf{x_j})$$

Minimize: $\sum_{i=1}^{N}(y_i - K(\mathbf{w}, \mathbf{x_i}))^2 \longrightarrow \mathbf{w} = (K^TK)^{-1}K^T\mathbf{y}$

Performance Comparison of Kernels



- Linear

$$\mathbf{X} \cdot \mathbf{V}$$

- Polynomial

$$(r + \mathbf{x} \cdot \mathbf{v})^d \text{ for } r \geq 0, d \geq 0$$

- Radial Basis Function

$$\exp(-\gamma \|\mathbf{x} - \mathbf{v}\|^2), \text{ for } \gamma > 0$$

- Gaussian

$$\exp(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{v}\|^2)$$

Conclusion - Representer Theorem

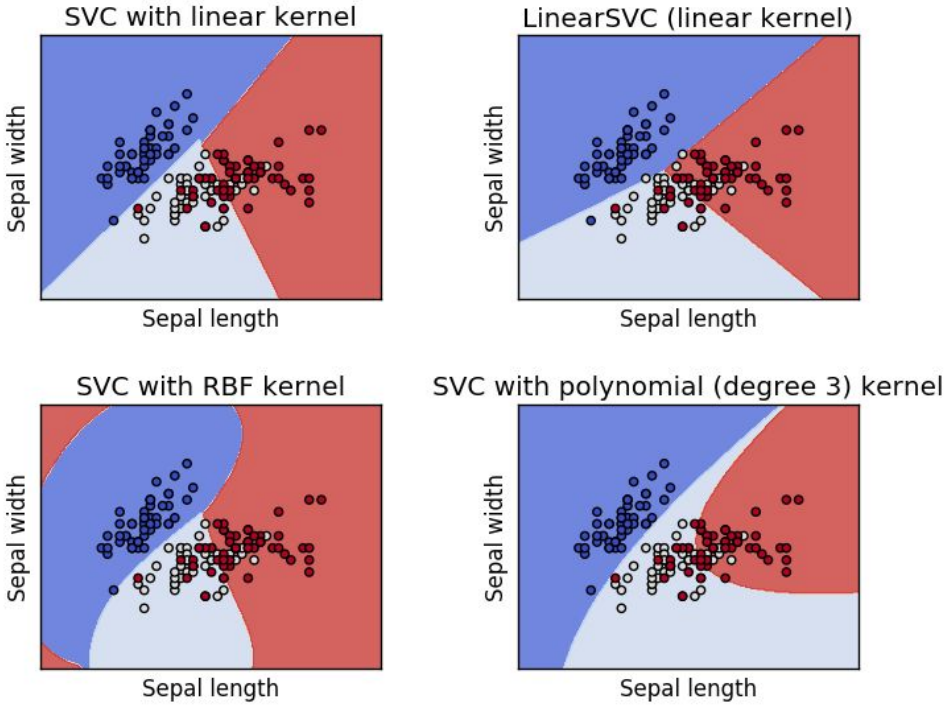
It turns out that almost all problems can be written as dot products, and therefore can be replaced with kernels! There is a powerful theorem the Representor Theorem which says:

Theorem 1 (The Representer Theorem). *Let k be a kernel on $\mathcal{X}$ and let $\mathcal{F}$ be its associated RKHS. Fix $x_1, \ldots, x_n \in \mathcal{X}$, and consider the optimization problem*

$$\min_{f \in \mathcal{F}} \; D(f(x_1), \ldots, f(x_n)) + P(\|f\|_{\mathcal{F}}^2), \tag{2}$$

where P is nondecreasing and D depends on f only though $f(x_1), \ldots, f(x_n)$. If (2) has a minimizer, then it has a minimizer of the form $f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$ where $\alpha_i \in \mathbb{R}$. Furthermore, if P is strictly increasing, then every solution of (2) has this form.

Multiclass classification with different kernels



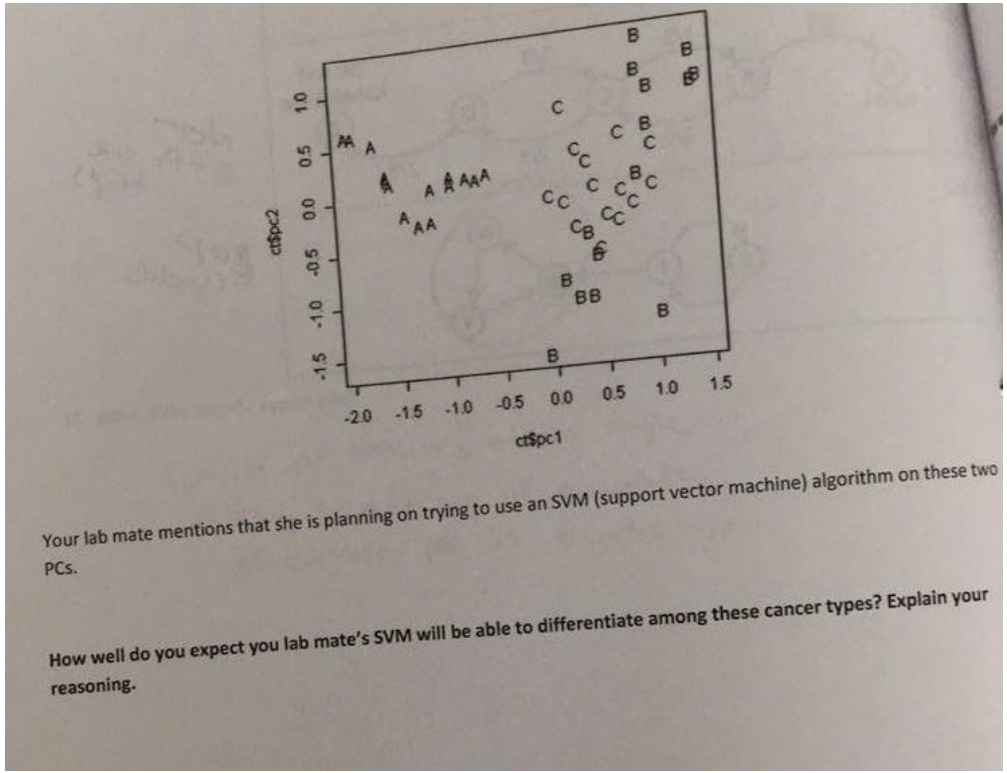
- For multiple classes, we use the *‘one vs all approach’.*
- Take one of the classes, split with respect to your desired kernel.
- Repeat this with the remaining sub-classes.

When to use which kernel?

Key Points (from Andrew Ng, Professor of Machine Learning at Stanford)

- Use linear kernel when number of features is larger than number of observations.
- Use gaussian kernel when number of observations is larger than number of features.
- If number of observations is larger than 50,000 speed could be an issue when using gaussian kernel; hence, one might want to use linear kernel.

Should this work for linear SVM?



Work through this example, thinking carefully about how to use the one vs all approach, and which kernels would work vs which would not.

Bayes Theorem

A digression about conditional probabilities

Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

Bayes' theorem is stated mathematically as the following equation:[2]

where *A* and *B* are events and P(B) ≠ 0.

- *P*(*A*) and *P*(*B*) are the probabilities of observing *A* and *B* without regard to each other.
- *P*(*A* | *B*), a conditional probability, is the probability of observing event *A* given that *B* is true.
- P(A) is known as the *prior - our initial assumption on how the data is distributed.*

P(*B* | *A*) is the probability of observing event *B* given that *A* is true.

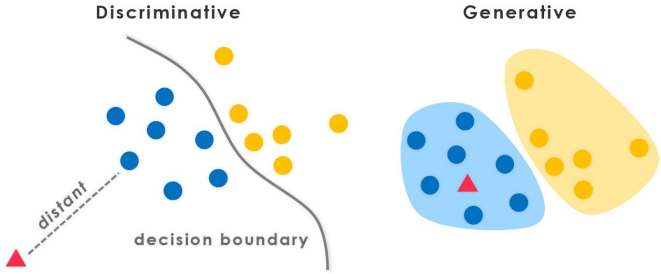
Bayesian interpretation

For proposition *A* and evidence *B*,

- *P* (*A*), the *prior*, is the initial degree of belief in *A*.
- *P* (*A* | *B*), the “posterior,” is the degree of belief having accounted for *B*.
- the quotient *P*(*B* |*A*)/*P*(*B*) represents the ***support*** *B* provides for *A*.

Bayes Theorem - Why do we use it?

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$



- We usually want to know P(Y|X) in discriminative models. This tells us the probability of an outcome Y given information about the population X.
- But what if we want to know more about the distribution of X given outcomes?
- For instance, can we identify if someone is male or female based on weight, height and shoe size?

Who stole the cookie?

Suppose there are two full bowls of cookies. Bowl #1 has 10 chocolate chip and 30 plain cookies, while bowl #2 has 20 of each. Our friend Fred picks a bowl at random, and then picks a cookie at random. We may assume there is no reason to believe Fred treats one bowl differently from another, likewise for the cookies. The cookie turns out to be a plain one. How probable is it that Fred picked it out of bowl #1?

Who stole the cookie?

Suppose there are two full bowls of cookies. Bowl #1 has 10 chocolate chip and 30 plain cookies, while bowl #2 has 20 of each. Our friend Fred picks a bowl at random, and then picks a cookie at random. We may assume there is no reason to believe Fred treats one bowl differently from another, likewise for the cookies. The cookie turns out to be a plain one. How probable is it that Fred picked it out of bowl #1?

$$P(E \mid H_1) = 30/40 = 0.75$$

$$P(E \mid H_2) = 20/40 = 0.5.$$

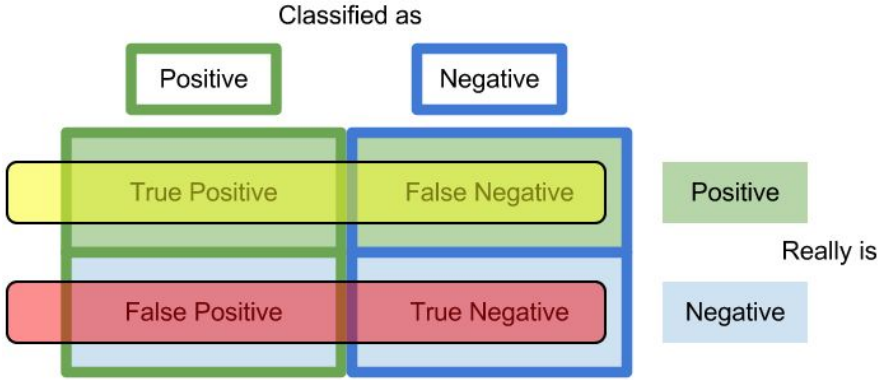
$$P(H_1 \mid E) = \frac{P(E \mid H_1)\,P(H_1)}{P(E \mid H_1)\,P(H_1) \;+\; P(E \mid H_2)\,P(H_2)}$$

$$= \frac{0.75 \times 0.5}{0.75 \times 0.5 + 0.5 \times 0.5}$$

$$= 0.6$$

Drug Test

Suppose a drug test is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. If a randomly selected individual tests positive, what is the probability that he is a user?

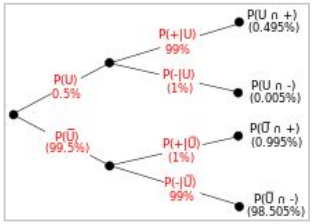


Computing the conditional probability

$$P(\text{User} \mid +) = \frac{P(+ \mid \text{User})P(\text{User})}{P(+ \mid \text{User})P(\text{User}) + P(+ \mid \text{Non-user})P(\text{Non-user})}$$

$$= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995}$$

$$\approx 33.2\%$$



Predicting bugs

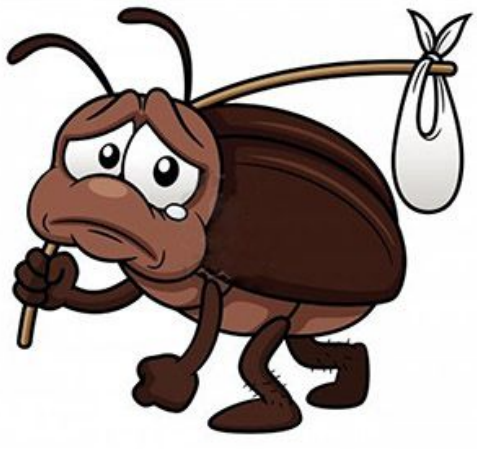
An entomologist spots what might be a rare subspecies of beetle, due to the pattern on its back. In the rare subspecies, 98% have the pattern, or $P(\text{Pattern} \mid \text{Rare}) = 98\%$. In the common subspecies, 5% have the pattern. The rare subspecies accounts for only 0.1% of the population. How likely is the beetle having the pattern to be rare, or what is $P(\text{Rare} \mid \text{Pattern})$?

Computing the conditional probability

$$P(\text{Rare} \mid \text{Pattern}) = \frac{P(\text{Pattern} \mid \text{Rare})P(\text{Rare})}{P(\text{Pattern} \mid \text{Rare})P(\text{Rare}) + P(\text{Pattern} \mid \text{Common})P(\text{Common})}$$

$$= \frac{0.98 \times 0.001}{0.98 \times 0.001 + 0.05 \times 0.999}$$

$$\approx 1.9\%$$



Exercise: Sad Lamps

Factory	% of total production	Probability of defective lamps
A	$0.35 = P(A)$	$0.015 = P(D \mid A)$
B	$0.35 = P(B)$	$0.010 = P(D \mid B)$
C	$0.30 = P(C)$	$0.020 = P(D \mid C)$

Question: Given that a lamp is defective, what is the probability that it was of type A?

Sex Classification

Problem: classify whether a given person is a male or a female based on the measured features. The features include height, weight, and foot size.

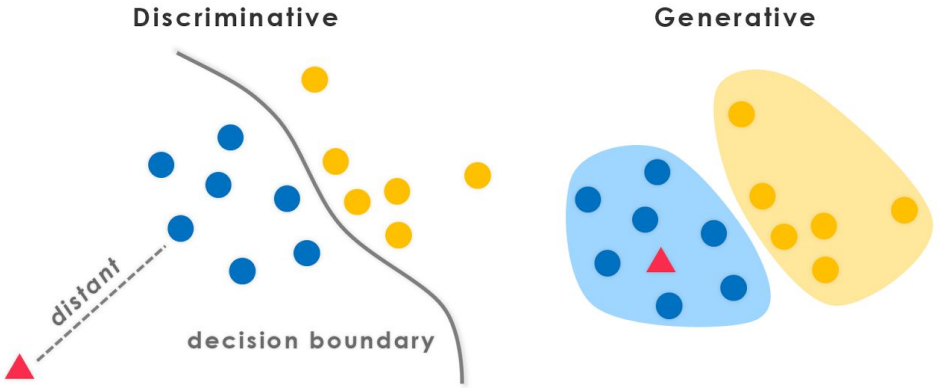
Sample data

Sex	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

But wait? Isn’t this just classification

Yes! But so far we have studied only discriminative models, ie. trying to learn p(y|x) from the data. This is our first example (Naives Bayes) where we want to actually learn the entire distribution of the data, ie the joint distribution p(y,x).

Let’s see this in action!



Computing the parameters

Sex	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033e-02	176.25	1.2292e+02	11.25	9.1667e-01
female	5.4175	9.7225e-02	132.5	5.5833e+02	7.5	1.6667e+00

$$\text{posterior (male)} = \frac{P(\text{male})\, p(\text{height} \mid \text{male})\, p(\text{weight} \mid \text{male})\, p(\text{foot size} \mid \text{male})}{evidence}$$

$$\text{posterior (female)} = \frac{P(\text{female})\, p(\text{height} \mid \text{female})\, p(\text{weight} \mid \text{female})\, p(\text{foot size} \mid \text{female})}{evidence}$$

$$\begin{aligned}\text{evidence} = {} & P(\text{male})\, p(\text{height} \mid \text{male})\, p(\text{weight} \mid \text{male})\, p(\text{foot size} \mid \text{male}) \\ & + P(\text{female})\, p(\text{height} \mid \text{female})\, p(\text{weight} \mid \text{female})\, p(\text{foot size} \mid \text{female})\end{aligned}$$

Computing the parameters

Sex	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033e-02	176.25	1.2292e+02	11.25	9.1667e-01
female	5.4175	9.7225e-02	132.5	5.5833e+02	7.5	1.6667e+00

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789$$

$$\sigma^2 = 3.5033 \cdot 10^{-2} \qquad \mu = 5.855$$

The parameters of normal distribution which have been previously determined from the training set. Note that a value greater than 1 is OK here – it is a probability density rather than a probability, because *height* is a continuous variable.

Posterior calculations

$$P(\text{female}) = 0.5$$

$$p(\text{height} \mid \text{female}) = 2.2346 \cdot 10^{-1}$$

$$p(\text{weight} \mid \text{female}) = 1.6789 \cdot 10^{-2}$$

$$p(\text{foot size} \mid \text{female}) = 2.8669 \cdot 10^{-1}$$

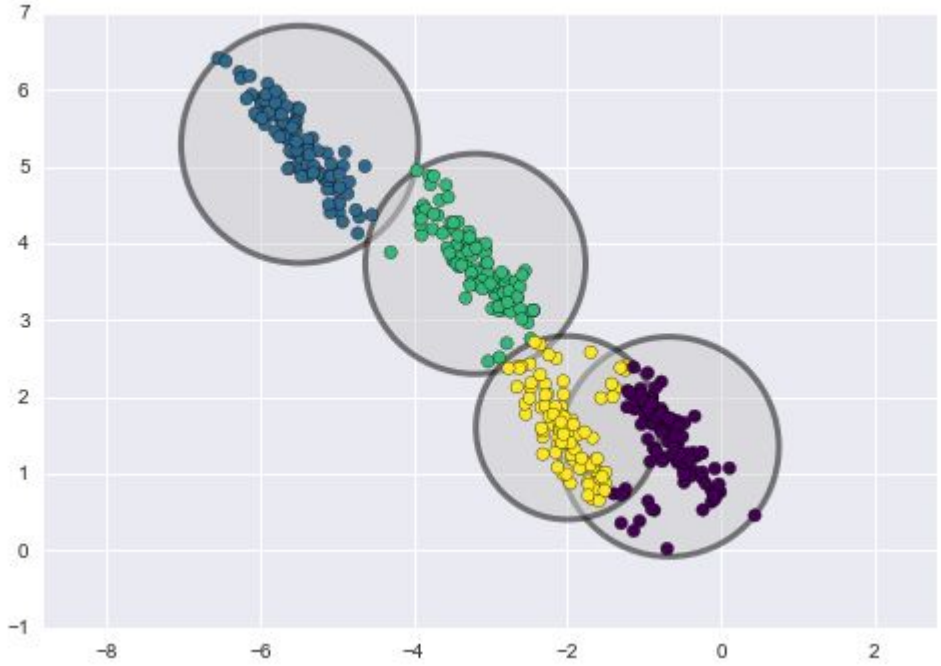
$$\text{posterior numerator (female)} = \text{their product} = 5.3778 \cdot 10^{-4}$$
$$\text{posterior numerator (male)} = \text{their product} = 6.1984 \cdot 10^{-9}$$

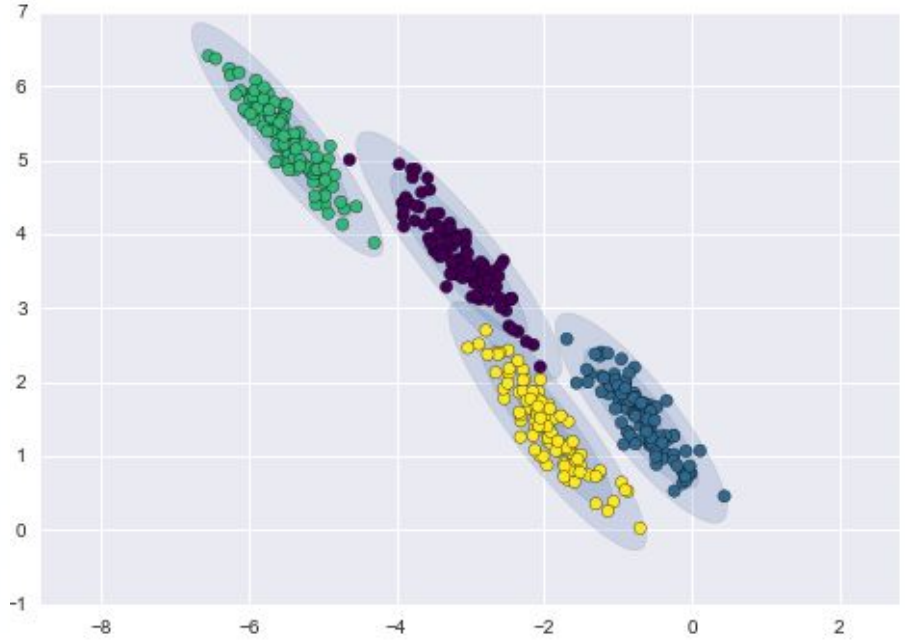
So it’s a female probably!

Finding the bias of a coin

Problem: If I flip a coin N times and I get k heads, what's the probability that the coin is fair, not fair? What's the bias?

K means vs GMM

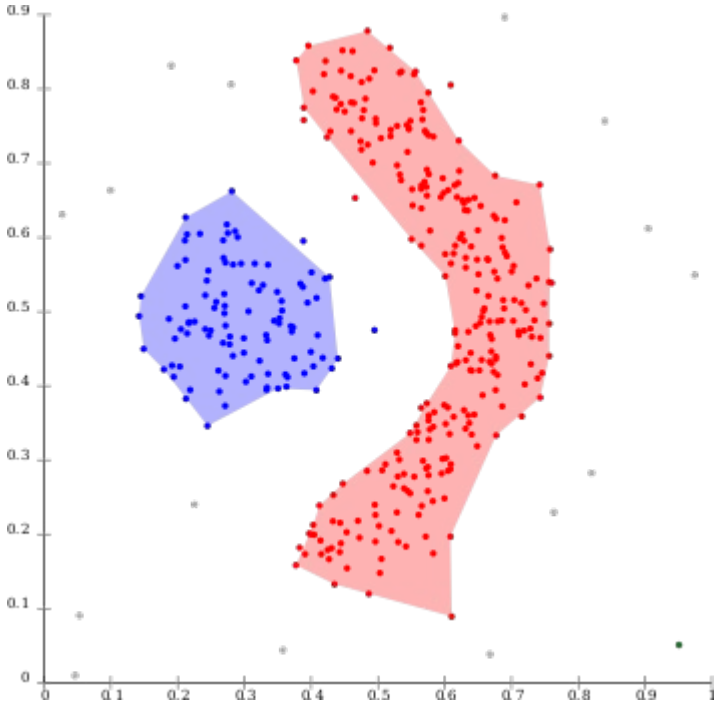




Density Based Clustering

Introduction to topic models

What to do here?



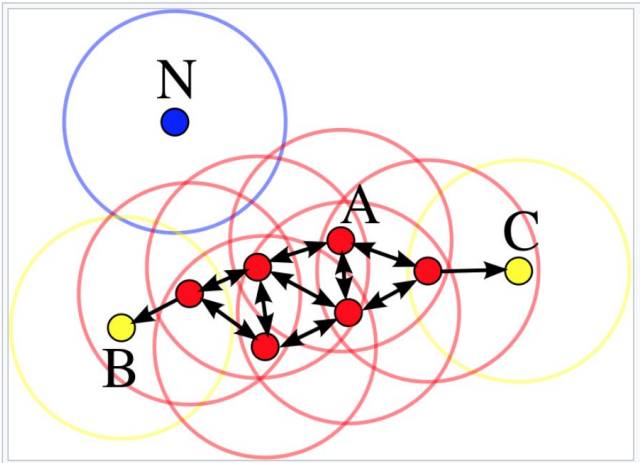
DBSCAN can find non-linearly separable clusters. This dataset cannot be adequately clustered with k-means or Gaussian Mixture EM clustering.

DBScan

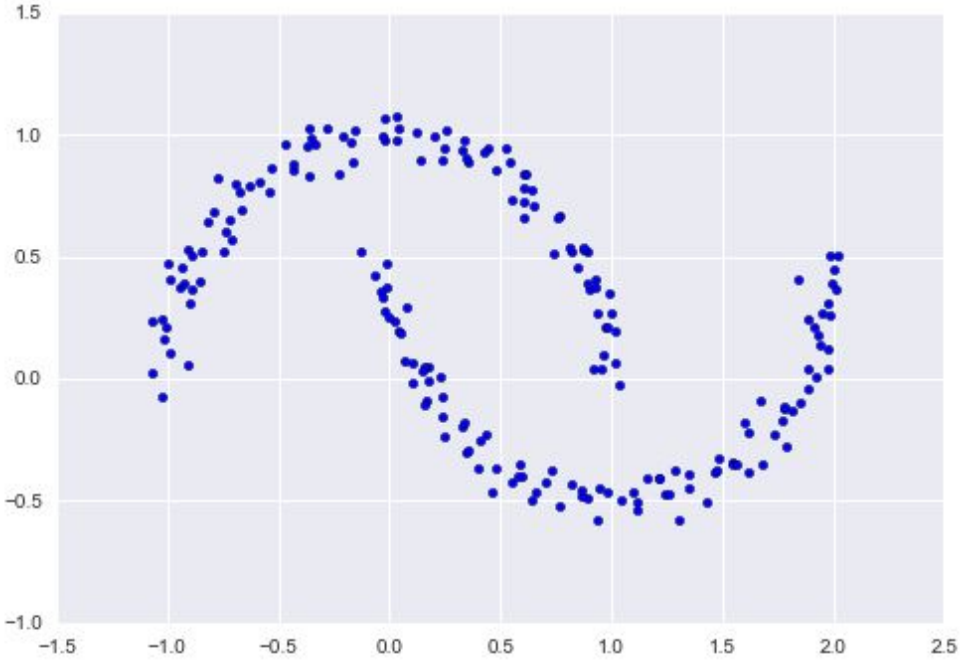
- A point p is a **core point** if at least minPts points are within distance ε (ε is the maximum radius of the neighborhood from p) of it (including p). Those points are said to be ***directly reachable*** from p.
- A point q **is directly reachable from** p if point q is within distance ε from point p and p must be a core point.
- A point q **is reachable from** p if there is a path $p_1, ..., p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i (all the points on the path must be core points, with the possible exception of q).
- All points not reachable from any other point are outliers.

- In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster.

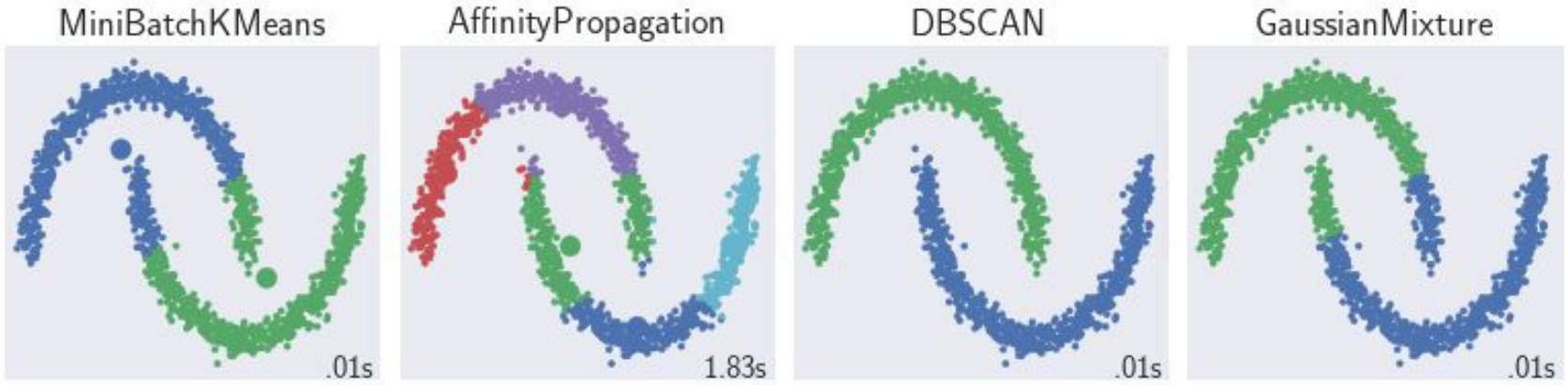
- Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.



Two Moons Problem



Comparison


MiniBatchKMeans
.01s
AffinityPropagation
1.83s
DBSCAN
.01s
GaussianMixture
.01s

Complexity Comparisons

Training

SVM: Complexity O(n^2) - O(n^3)

Expectation Maximization: O(n^2)

K-means: O(n^2)

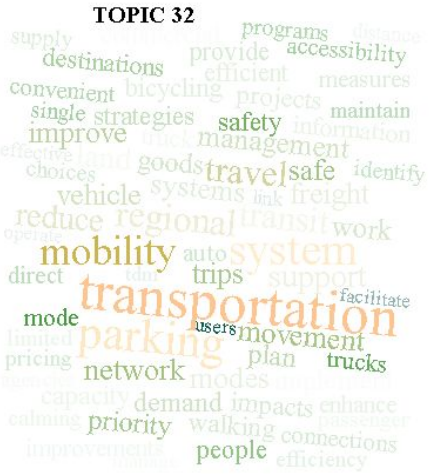
DBScan: O(n^2)

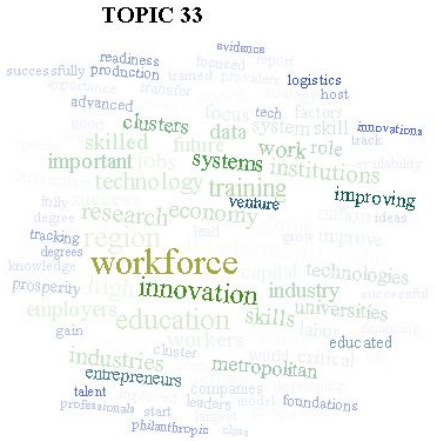
Latent Dirichlet Allocation (LDA)

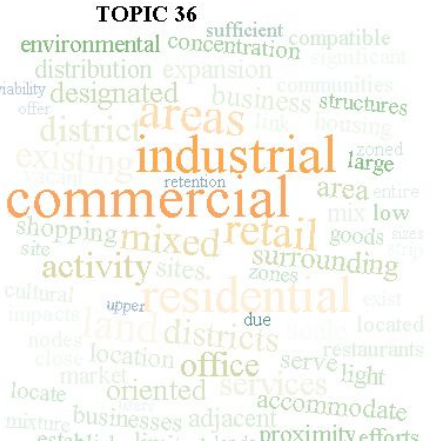
Introduction to topic models

Topic Modeling

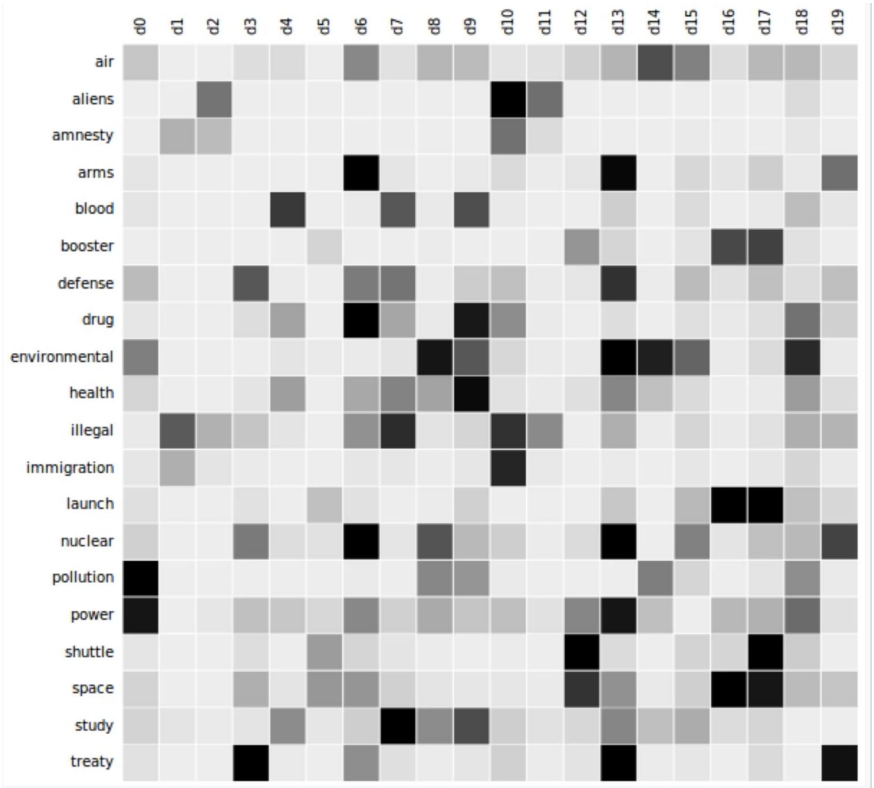
- Another example of a generative /clustering model.
- We wish to discover the distribution of words, and cluster them into some finite collection. How do we proceed? Not much different than before!



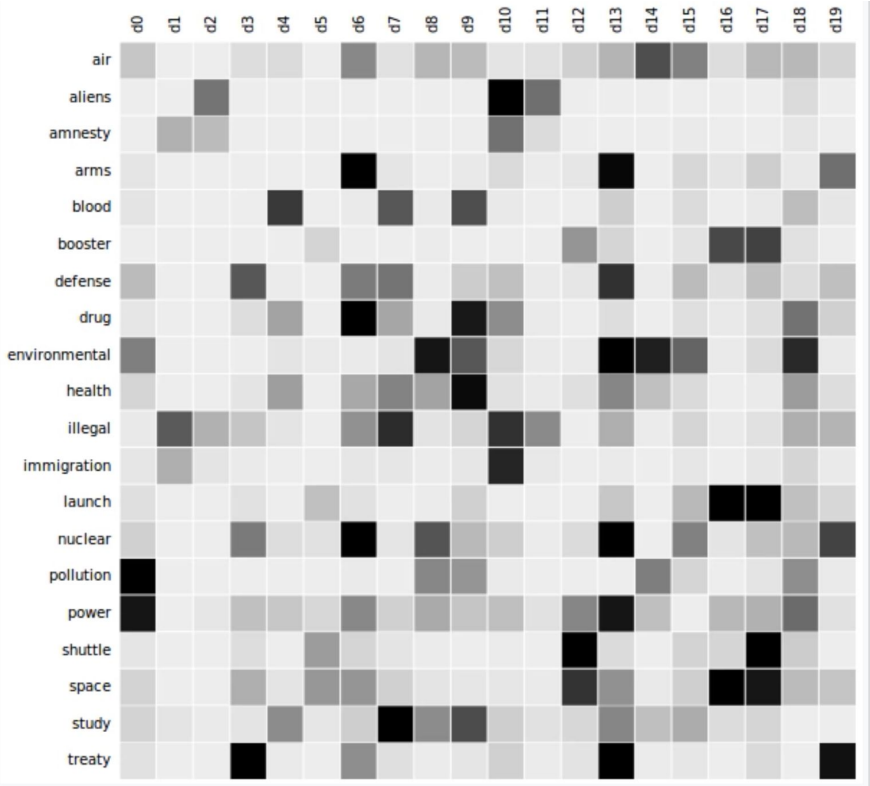


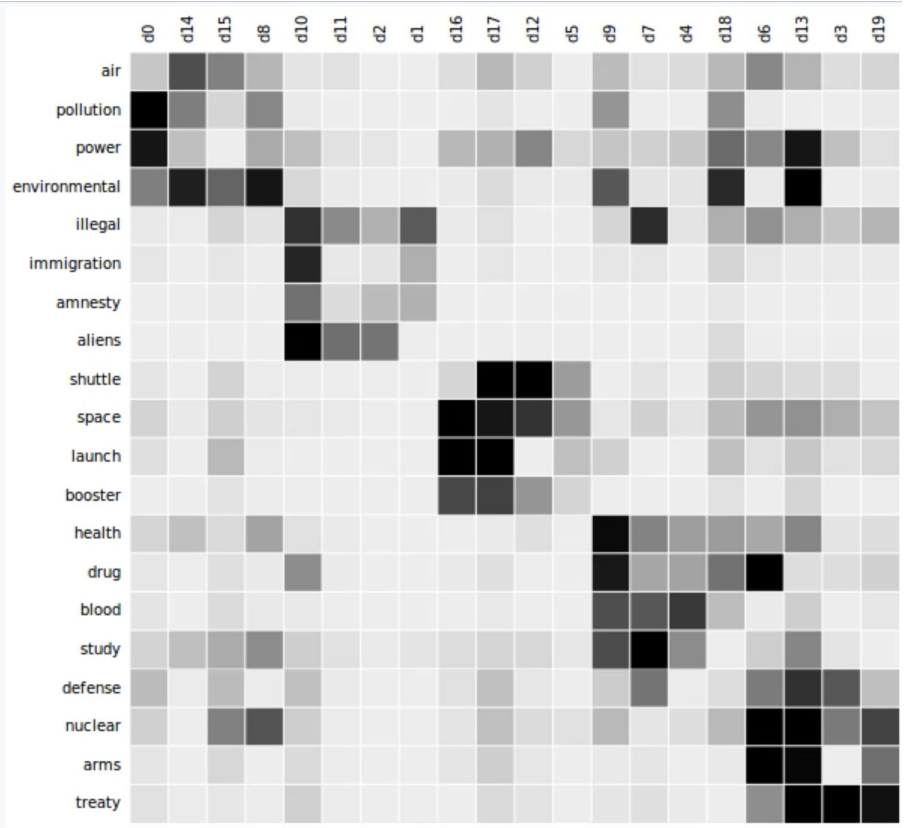


Distribution of words in documents



Distribution of words in documents





Latent Dirichlet Allocation (LDA)

- α is the parameter of the Dirichlet prior on the per-document topic distributions,
- β is the parameter of the Dirichlet prior on the per-topic word distribution,
- θ_m is the topic distribution for document *m*.
- φ_k is the word distribution for topic *k*.
- z_{mn} is the topic for the *n*-th word in document *m*, and
- w_{mn} is the specific word.