TDA231
Clustering and Mixture Models

Devdatt Dubhashi
dubhashi@chalmers.se

Dept. of Computer Science and Engg.
Chalmers University

March 2016

Unsupervised learning

- Everything we've seen so far has been supervised
- We were given a set of $\mathbf{x}_n$ **and** associated t_n.

Unsupervised learning

- Everything we've seen so far has been supervised
- We were given a set of $\mathbf{x}_n$ **and** associated t_n.
- What if we just have $\mathbf{x}_n$?
- For example:
  - $\mathbf{x}_n$ is a binary vector indicating products customer n has bought.
  - Can group customers that buy similar products.
  - Can group products bought together.

Unsupervised learning

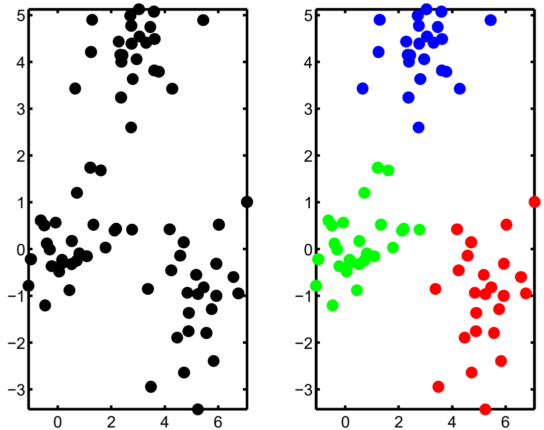
- Everything we've seen so far has been supervised
- We were given a set of $\mathbf{x}_n$ **and** associated t_n.
- What if we just have $\mathbf{x}_n$?
- For example:
  - $\mathbf{x}_n$ is a binary vector indicating products customer n has bought.
  - Can group customers that buy similar products.
  - Can group products bought together.
- Known as Clustering
- And is an example of unsupervised learning.

Unsupervised learning

- Everything we've seen so far has been supervised
- We were given a set of $\mathbf{x}_n$ **and** associated t_n.
- What if we just have $\mathbf{x}_n$?
- For example:
  - $\mathbf{x}_n$ is a binary vector indicating products customer n has bought.
  - Can group customers that buy similar products.
  - Can group products bought together.
- Known as Clustering
- And is an example of unsupervised learning.
- *Supervised Learning is just the icing on the cake which is unsupervised learning.*

  Yann Le CUn, NIPS 2016

Clustering



- In this example each object has two attributes:

$$\mathbf{x}_n = [x_{n1}, x_{n2}]^{\mathrm{T}}$$

- Left: data.
- Right: data after clustering (points coloured according to cluster membership).

What we'll cover

- 2 algorithms:
  - K-means
  - Mixture models
- The two are somewhat related.
- We'll also see how K-means can be kernelised.

What we'll cover

- 2 algorithms:
  - K-means
  - Mixture models
- The two are somewhat related.
- We'll also see how K-means can be kernelised.

K-means

- Assume that there are K clusters.
- Each cluster is defined by a position in the input space:

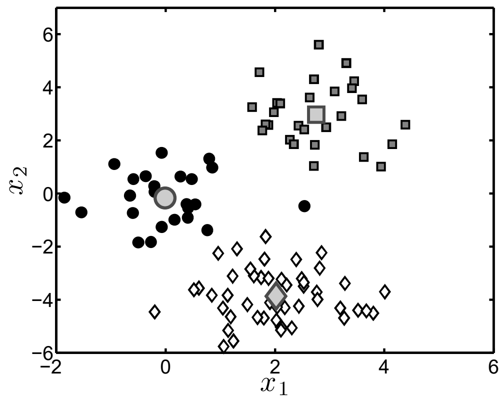
$$\boldsymbol{\mu}_k = [\mu_{k1}, \mu_{k2}]^\mathsf{T}$$

K-means

- Assume that there are K clusters.
- Each cluster is defined by a position in the input space:

$$\boldsymbol{\mu}_k = [\mu_{k1}, \mu_{k2}]^\mathsf{T}$$

- Each $\mathbf{x}_n$ is assigned to its closest cluster:

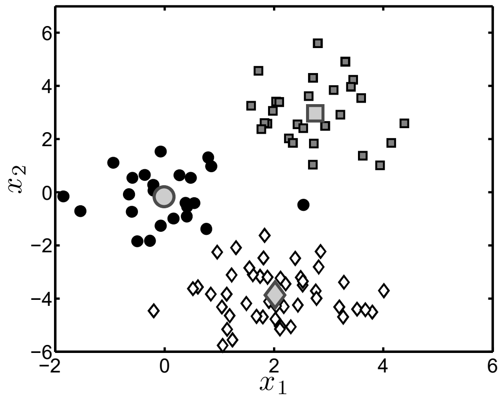


K-means

- Assume that there are K clusters.
- Each cluster is defined by a position in the input space:

$$\boldsymbol{\mu}_k = [\mu_{k1}, \mu_{k2}]^{\mathsf{T}}$$

- Each $\mathbf{x}_n$ is assigned to its closest cluster:



- Distance is normally Euclidean distance:

$$d_{nk} = (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
  2. Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
  2. Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$
  3. $z_{nk} = 1$ if $\mathbf{x}_n$ assigned to $\boldsymbol{\mu}_k$ (0 otherwise)

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
  2. Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$
  3. $z_{nk} = 1$ if $\mathbf{x}_n$ assigned to $\boldsymbol{\mu}_k$ (0 otherwise)
  4. Update $\boldsymbol{\mu}_k$ to average of $\mathbf{x}_n$s assigned to $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk}\mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
  2. Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$
  3. $z_{nk} = 1$ if $\mathbf{x}_n$ assigned to $\boldsymbol{\mu}_k$ (0 otherwise)
  4. Update $\boldsymbol{\mu}_k$ to average of $\mathbf{x}_n$s assigned to $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk}\mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

  5. Return to 2 until assignments do not change.

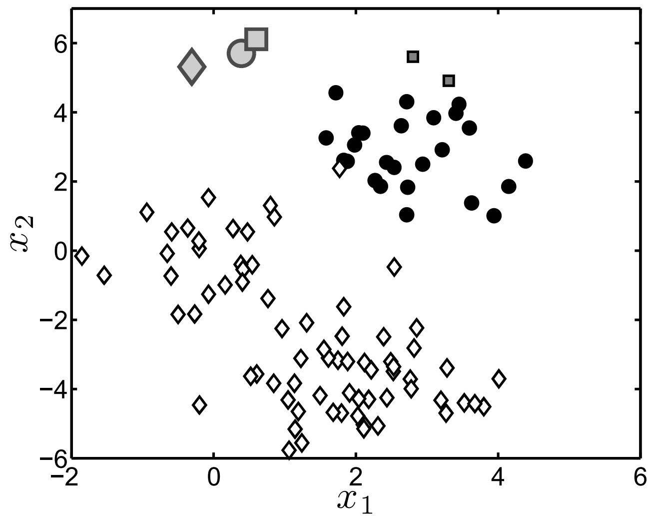
How do we find $\boldsymbol{\mu}_k$?

- No analytical solution – we can't write down $\boldsymbol{\mu}_k$ as a function of $\mathbf{X}$.
- Use an iterative algorithm:
  1. Guess $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
  2. Assign each $\mathbf{x}_n$ to its closest $\boldsymbol{\mu}_k$
  3. $z_{nk} = 1$ if $\mathbf{x}_n$ assigned to $\boldsymbol{\mu}_k$ (0 otherwise)
  4. Update $\boldsymbol{\mu}_k$ to average of $\mathbf{x}_n$s assigned to $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} z_{nk}\mathbf{x}_n}{\sum_{n=1}^{N} z_{nk}}$$

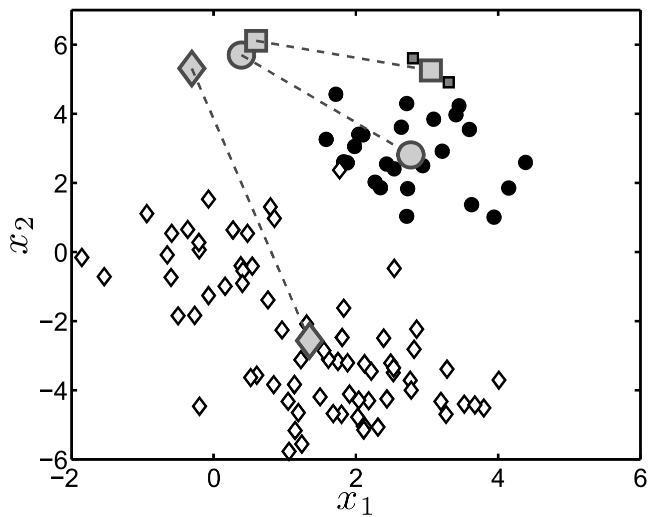
  5. Return to 2 until assignments do not change.
- Algorithm will converge....it will reach a point where the assignments don't change.

K-means – example



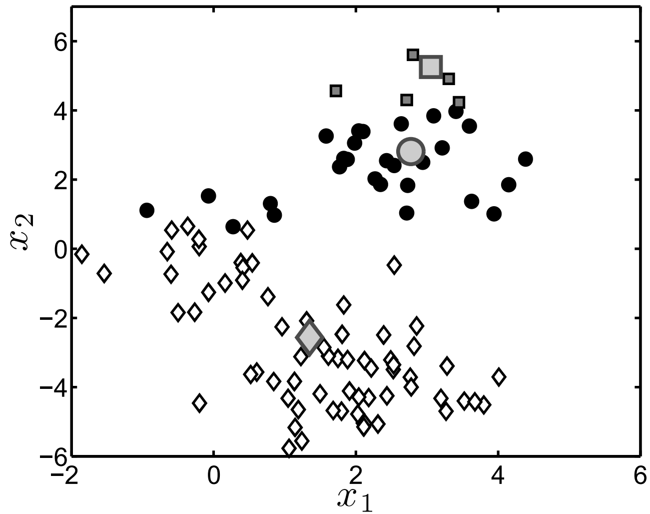
- Cluster means randomly assigned (top left).
- Points assigned to their closest mean.

K-means – example



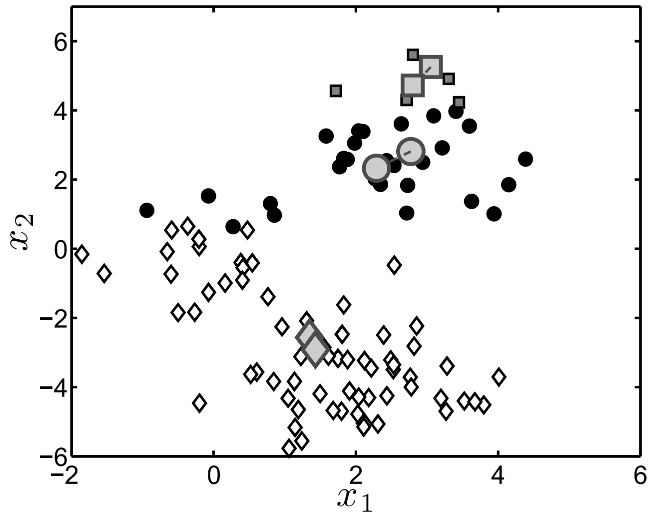
- Cluster means updated to mean of assigned points.

K-means – example



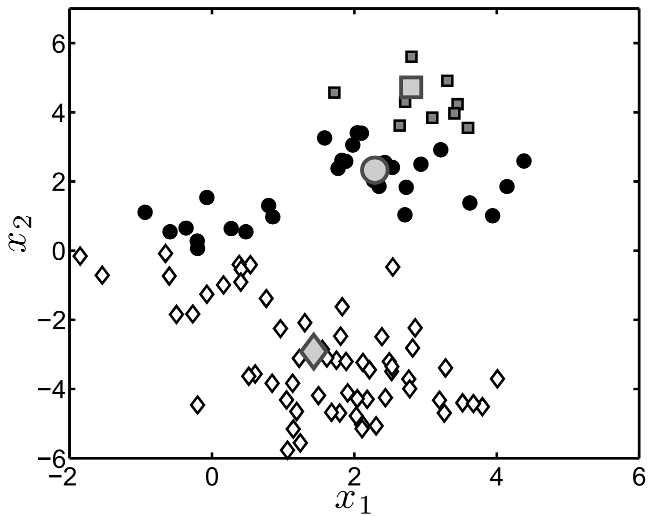
- Points re-assigned to closest mean.

K-means – example



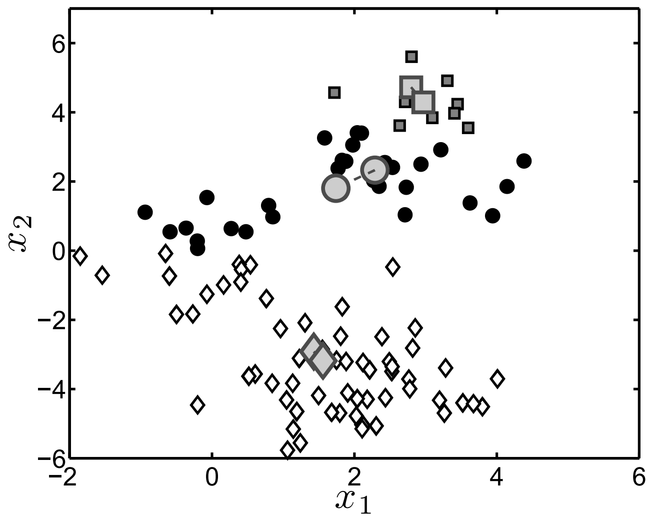
- Cluster means updated to mean of assigned points.

K-means – example



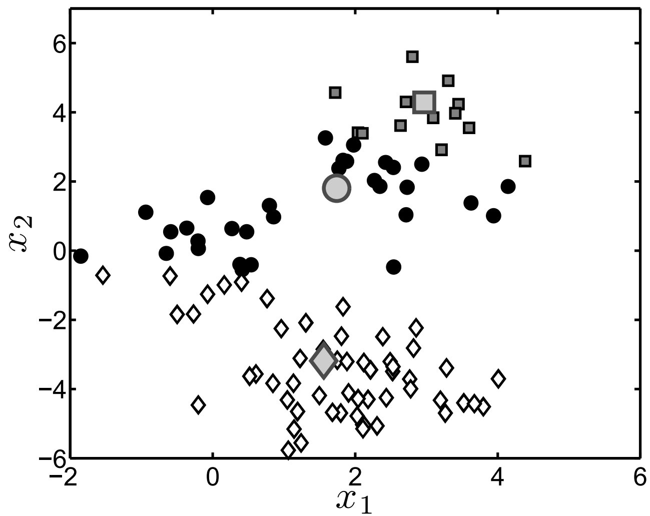
- Assign point to closest mean.

K-means – example



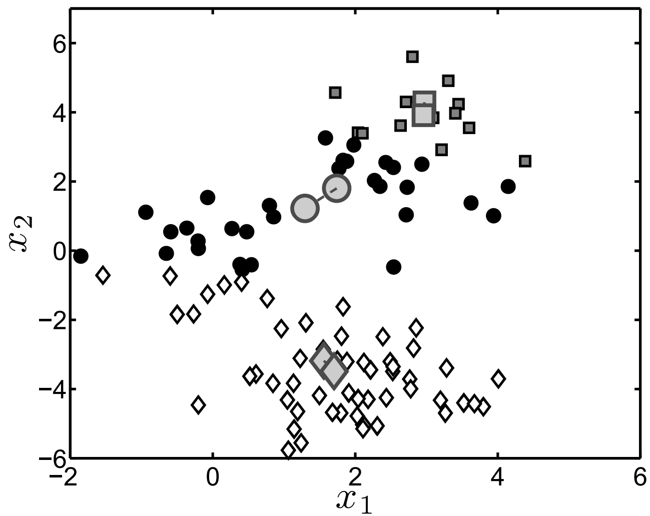
- Update mean.

K-means – example



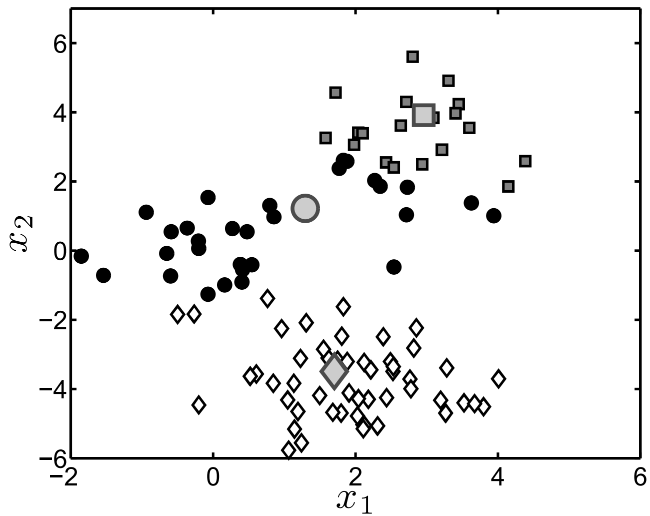
- Assign point to closest mean.

K-means – example



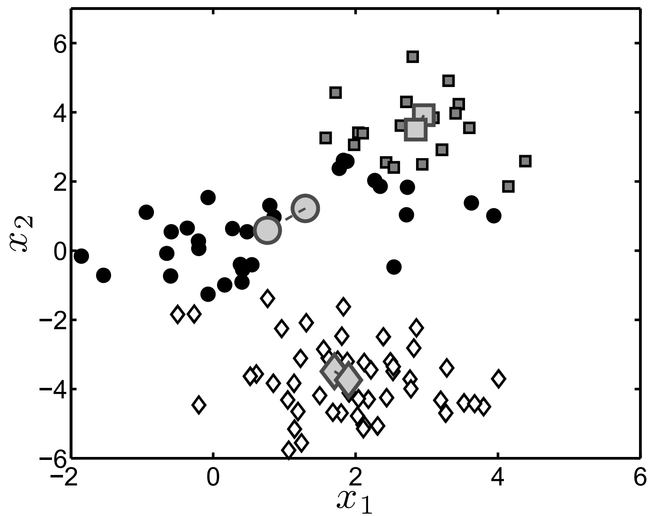
- Update mean.

K-means – example



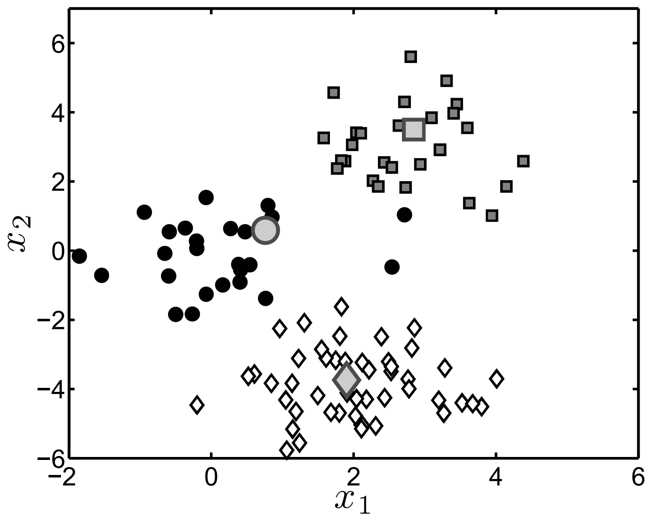
- Assign point to closest mean.

K-means – example



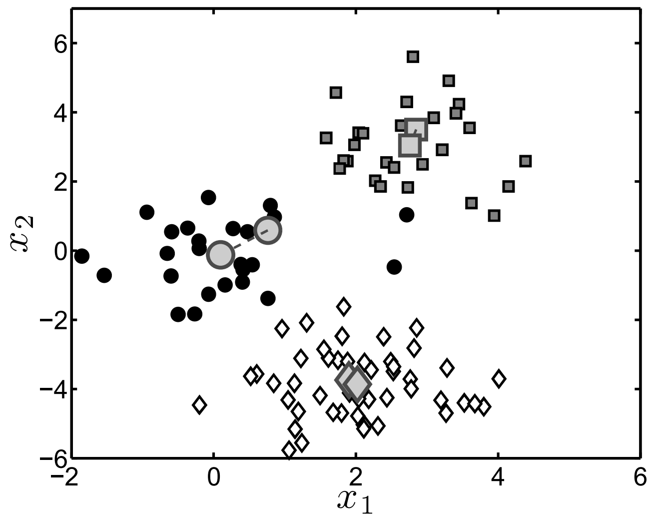
- Update mean.

K-means – example



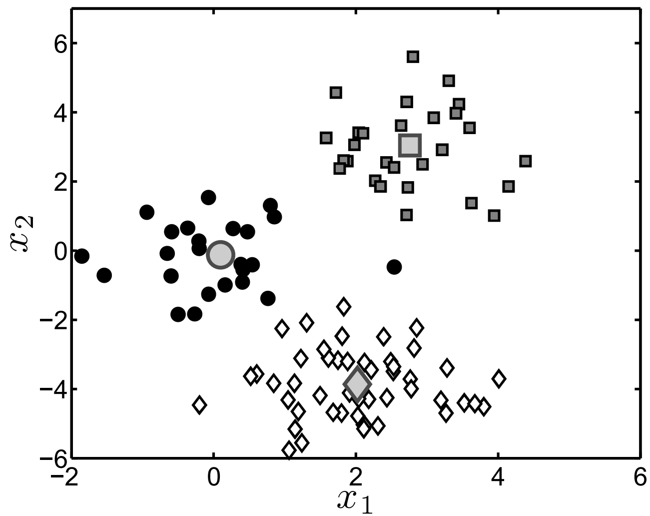
- Assign point to closest mean.

K-means – example



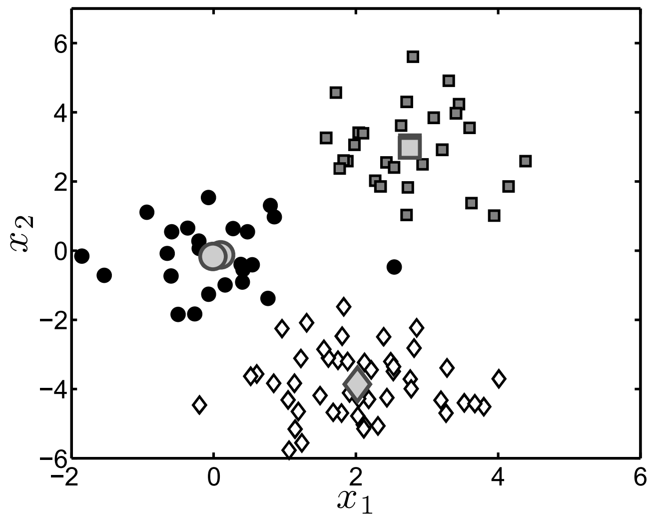
- Update mean.

K-means – example



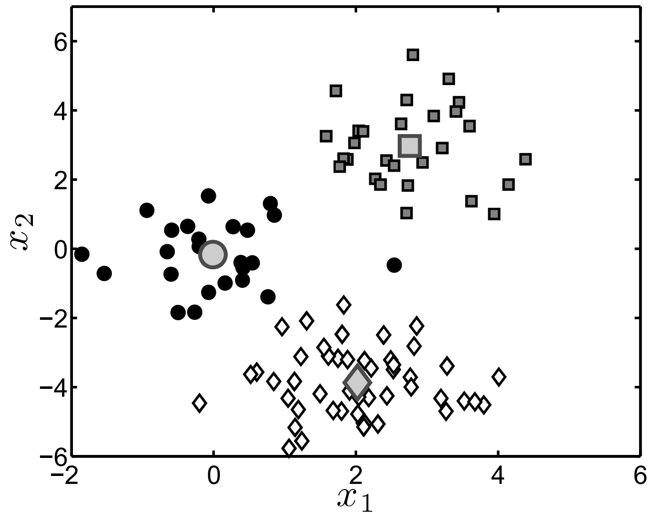
- Assign point to closest mean.

K-means – example



- Update mean.

K-means – example



- Solution at convergence.

Two Issues with K-Means

Two Issues with K-Means

- What value of k should we use?

Two Issues with K-Means

- What value of k should we use?
- How should we pick the initial centers?

Two Issues with K-Means

- What value of k should we use?
- How should we pick the initial centers?
- Both these significantly affect resulting clustering.

Initializing Centers

Initializing Centers

- Pick k random points.

Initializing Centers

- Pick k random points.
- Pick k points at random from input points.

Initializing Centers

- Pick k random points.
- Pick k points at random from input points.
- Assign points at random to k groups and then take centers of these groups.

Initializing Centers

- Pick k random points.
- Pick k points at random from input points.
- Assign points at random to k groups and then take centers of these groups.
- Pick a random input point for first center, next center at a point as far away from this as possible, next as far away from first two ...

k–Means++ (D. Arthur and S. Vassilvitskii (2007)

- Start with $C_1 := \{\mathbf{x}\}$ where $\mathbf{x}$ is chosen at random from input points.
- For $i \geq 2$, pick a point $\mathbf{x}$ according to a probability distribution ν_i:

$$\nu_i(x) = \frac{d^2(x, C_{i-1})}{\sum_y d^2(y, C_{i-1})}$$

  and set $C_i := C_{i-1} \cup \{\mathbf{x}\}$.

Gives a provably good $O(\log n)$ approximation to optimal clustering.

Choosing k

- Intra-cluster variance:

$$W_k := \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \boldsymbol{\mu}_k)^2.$$

- $W := \sum_k W_k$.
- Pick k to minimize W_k
- Elbow heuristic, Gap Statistic ...

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.
- If you take only second term ...

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.
- If you take only second term ...
- ... $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ for all i, j.

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.
- If you take only second term ...
- ... $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ for all i, j.
- By varying λ, we steer between these two extremes.

Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.
- If you take only second term ...
- ... $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ for all i, j.
- By varying λ, we steer between these two extremes.
- Do not need to know k in advance and do not need to do careful intialization.

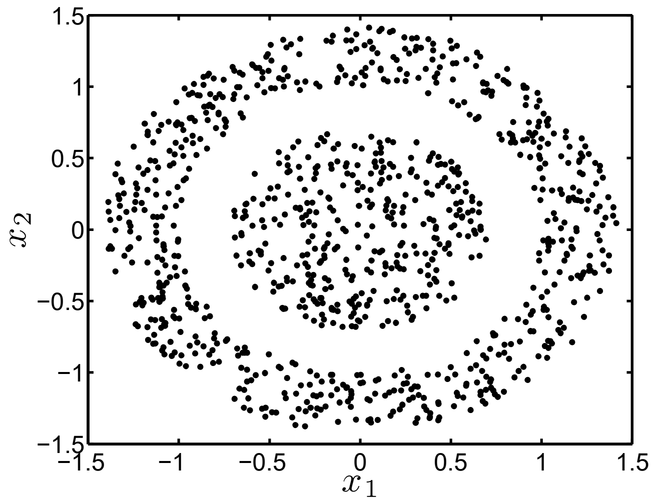
Sum of Norms (SON) Convex Relaxation

SON Relaxation (Lindsten et al 2011)

$$\min_{\boldsymbol{\mu}} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|^2 + \lambda \sum_{i<j} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

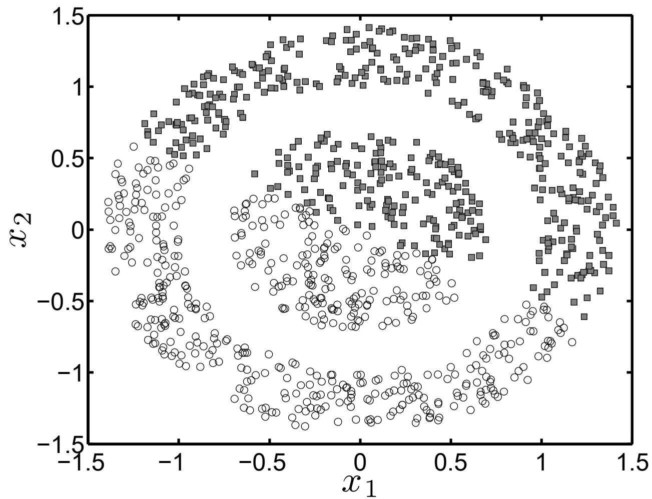
- If you take only first term ...
- ... $\boldsymbol{\mu}_i = \mathbf{x}_i$ for all i.
- If you take only second term ...
- ... $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$ for all i, j.
- By varying λ, we steer between these two extremes.
- Do not need to know k in advance and do not need to do careful intialization.
- Fast scalable algorithm with guarantees under submission later today to ICML ...

When does K-means break?



- Data has clear cluster structure.
- Outer cluster can not be represented as a single point.

When does K-means break?



- Data has clear cluster structure.
- Outer cluster can not be represented as a single point.

Kernelising K-means

- Maybe we can kernelise K-means?
- Distances:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

Kernelising K-means

- Maybe we can kernelise K-means?
- Distances:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- Cluster means:

$$\boldsymbol{\mu}_k = \frac{\sum_{m=1}^{N} z_{mk}\mathbf{x}_m}{\sum_{m=1}^{N} z_{mk}}$$

Kernelising K-means

- Maybe we can kernelise K-means?
- Distances:

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- Cluster means:

$$\boldsymbol{\mu}_k = \frac{\sum_{m=1}^N z_{mk}\mathbf{x}_m}{\sum_{m=1}^N z_{mk}}$$

- Distances can be written as (defining $N_k = \sum_n z_{nk}$):

$$(\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T}(\mathbf{x}_n - \boldsymbol{\mu}_k) = \left(\mathbf{x}_n - N_k^{-1}\sum_{m=1}^N z_{mk}\mathbf{x}_m\right)^\mathsf{T}\left(\mathbf{x}_n - N_k^{-1}\sum_{m=1}^N z_{mk}\mathbf{x}_m\right)$$

Kernelising K-means

- Multiply out:

$$\mathbf{x}_n^{\mathsf{T}}\mathbf{x}_n - 2N_k^{-1}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m^{\mathsf{T}}\mathbf{x}_n + N_k^{-2}\sum_{m,l} z_{mk}z_{lk}\mathbf{x}_m^{\mathsf{T}}\mathbf{x}_l$$

Kernelising K-means

- Multiply out:

$$\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n - 2N_k^{-1}\sum_{m=1}^{N} z_{mk}\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_n + N_k^{-2}\sum_{m,l} z_{mk}z_{lk}\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_l$$

- Kernel substitution:

$$k(\mathbf{x}_n,\mathbf{x}_n) - 2N_k^{-1}\sum_{m=1}^{N} z_{mk}k(\mathbf{x}_n,\mathbf{x}_m) + N_k^{-2}\sum_{m,l=1}^{N} z_{mk}z_{lk}k(\mathbf{x}_m,\mathbf{x}_l)$$

Kernel K-means

- Algorithm:
  1. Choose a kernel and any necessary parameters.

Kernel K-means

- Algorithm:
  1. Choose a kernel and any necessary parameters.
  2. Start with random assignments z_{nk}.

Kernel K-means

- Algorithm:
  1. Choose a kernel and any necessary parameters.
  2. Start with random assignments z_{nk}.
  3. For each $\mathbf{x}_n$ assign it to the nearest 'center' where distance is defined as:

$$k(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1} \sum_{m=1}^{N} z_{mk} k(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2} \sum_{m,l=1}^{N} z_{mk} z_{lk} k(\mathbf{x}_m, \mathbf{x}_l)$$

N_k = number of points assigned to k

k(x_n, x_m) = ||x_n - x_m || / 2sigma^2

Kernel K-means

- Algorithm:
  1. Choose a kernel and any necessary parameters.
  2. Start with random assignments z_{nk}.
  3. For each $\mathbf{x}_n$ assign it to the nearest 'center' where distance is defined as:

$$k(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1} \sum_{m=1}^{N} z_{mk} k(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2} \sum_{m,l=1}^{N} z_{mk} z_{lk} k(\mathbf{x}_m, \mathbf{x}_l)$$

  4. If assignments have changed, return to 3.

Kernel K-means

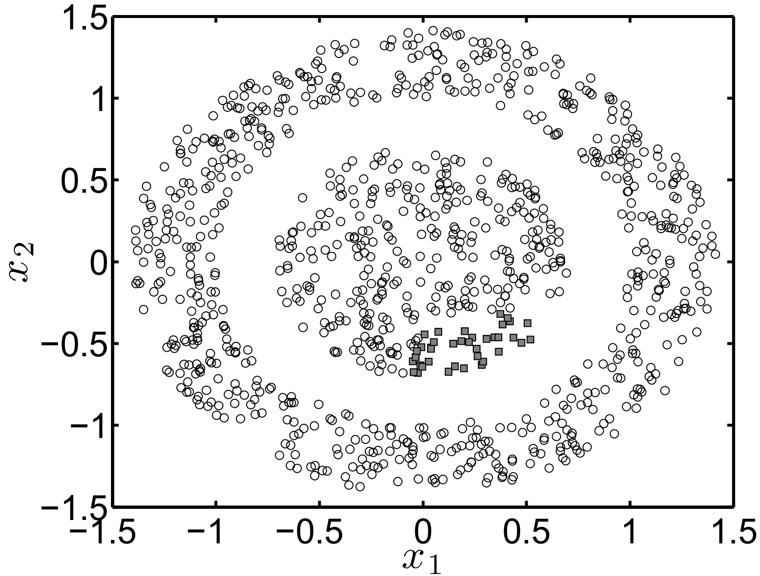
- Algorithm:
  1. Choose a kernel and any necessary parameters.
  2. Start with random assignments z_{nk}.
  3. For each $\mathbf{x}_n$ assign it to the nearest 'center' where distance is defined as:

$$k(\mathbf{x}_n, \mathbf{x}_n) - 2N_k^{-1} \sum_{m=1}^{N} z_{mk} k(\mathbf{x}_n, \mathbf{x}_m) + N_k^{-2} \sum_{m,l=1}^{N} z_{mk} z_{lk} k(\mathbf{x}_m, \mathbf{x}_l)$$

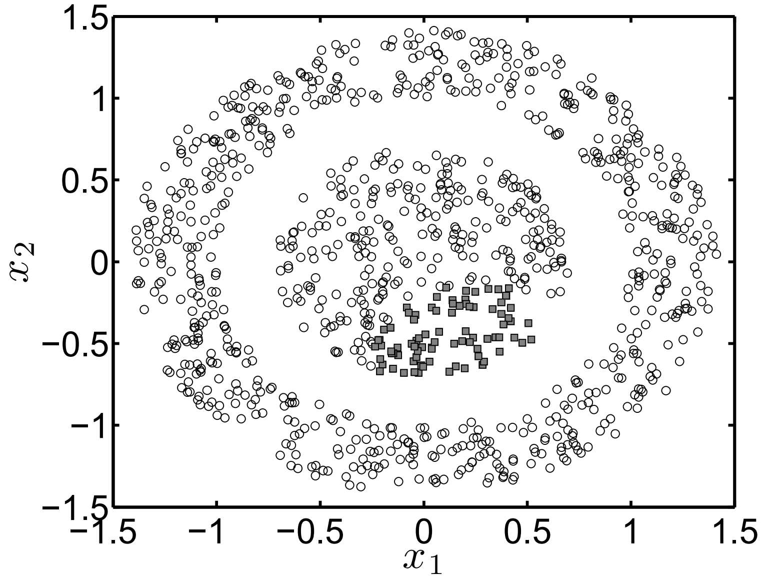
  4. If assignments have changed, return to 3.

- Note – no $\boldsymbol{\mu}_k$. This would be $N_k^{-1} \sum_n z_{nk} \phi(\mathbf{x}_n)$ but we don't know $\phi(\mathbf{x}_n)$ for kernels. We only know $\phi(\mathbf{x}_n)^\mathsf{T} \phi(\mathbf{x}_m)$ (last week)...

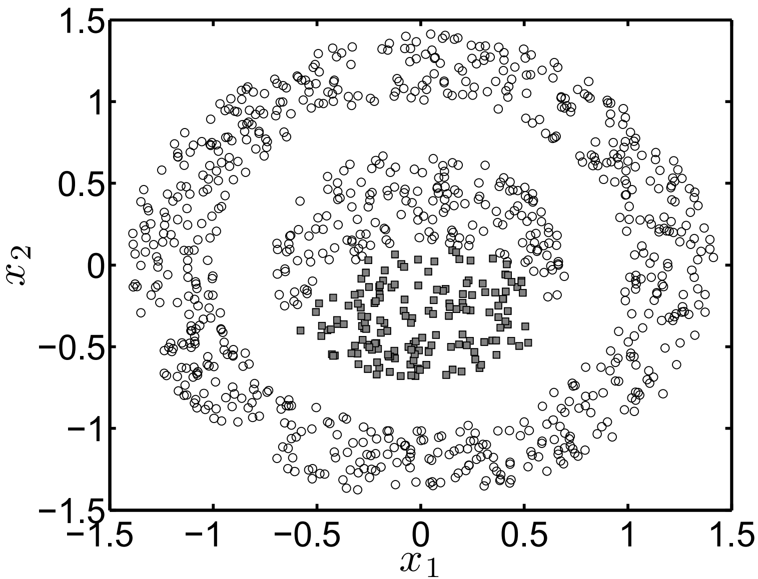
Kernel K-means – example



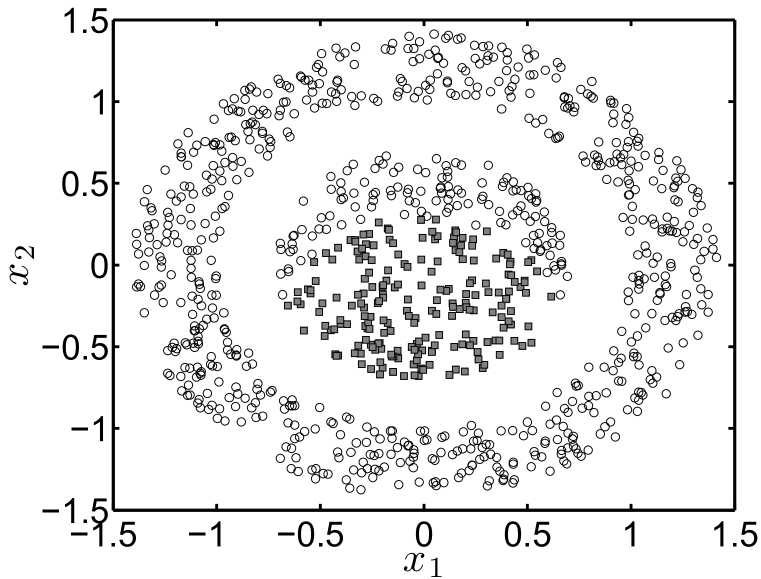
Kernel K-means – example



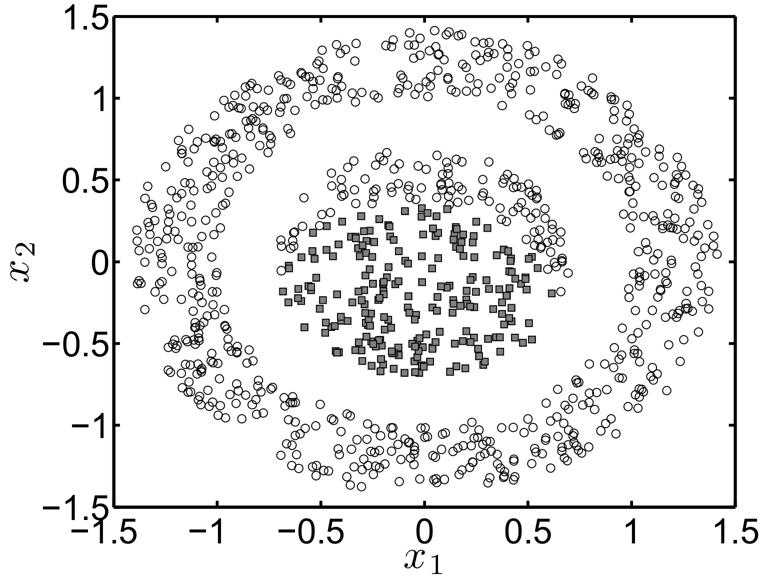
Kernel K-means – example



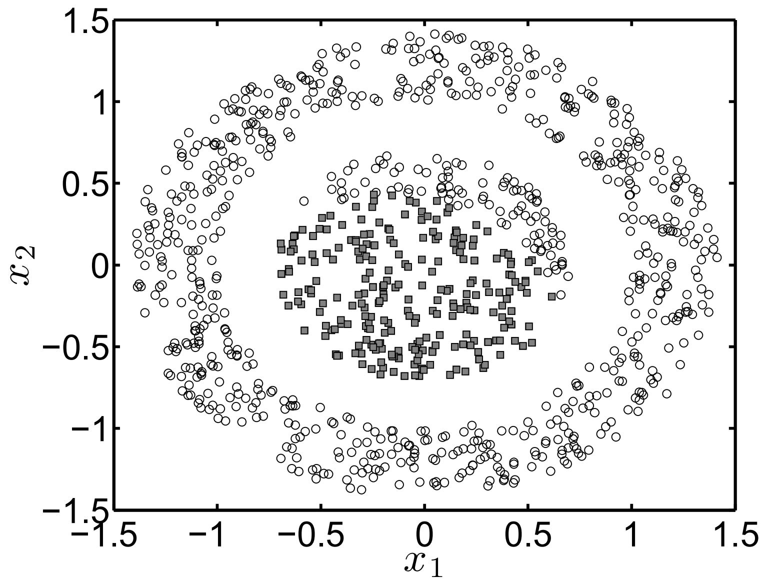
Kernel K-means – example



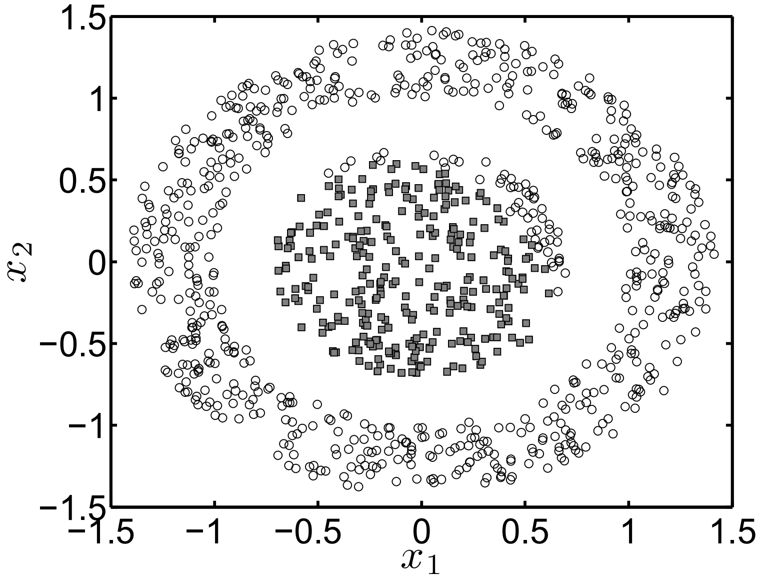
Kernel K-means – example



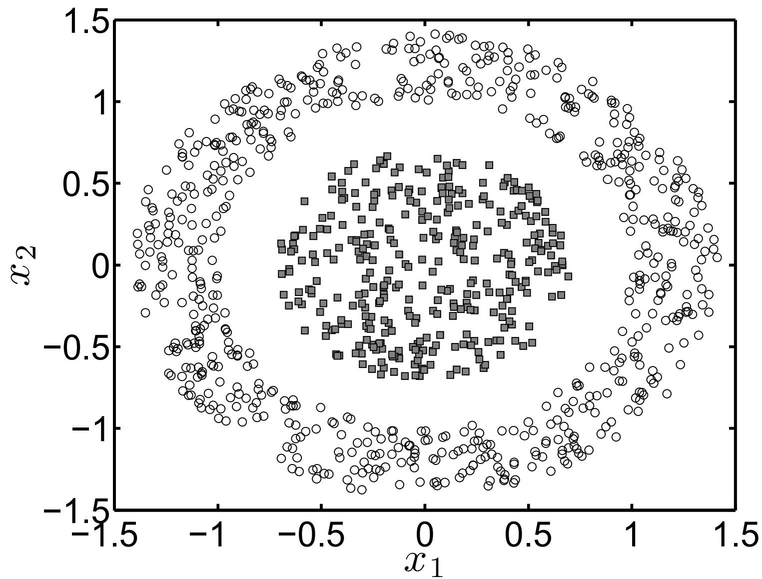
Kernel K-means – example



Kernel K-means – example



Kernel K-means – example



- Solution at convergence.

Kernel K-means

- Makes simple K-means algorithm more flexible.
- But, have to now set additional parameters.
- Very sensitive to initial conditions – lots of local optima.

K-means – summary

- Simple (and effective) clustering strategy.

K-means – summary

- Simple (and effective) clustering strategy.
- Converges to (local) minima of:

$$\sum_n \sum_k z_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

K-means – summary

- Simple (and effective) clustering strategy.
- Converges to (local) minima of:

$$\sum_n \sum_k z_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- Sensitive to initialisation.

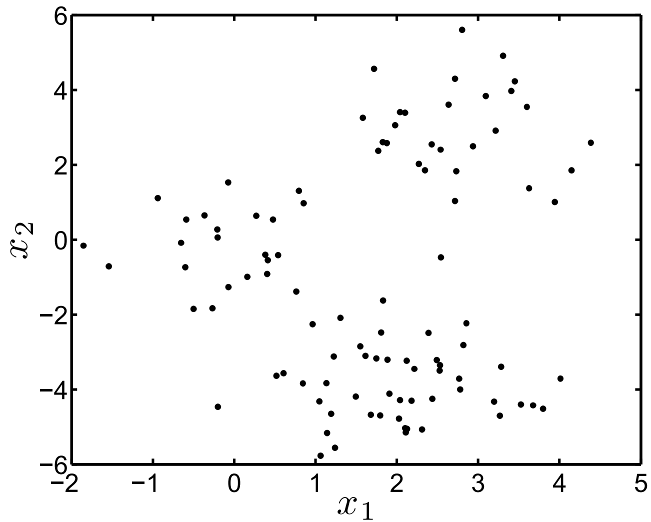
K-means – summary

- Simple (and effective) clustering strategy.
- Converges to (local) minima of:

$$\sum_n \sum_k z_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

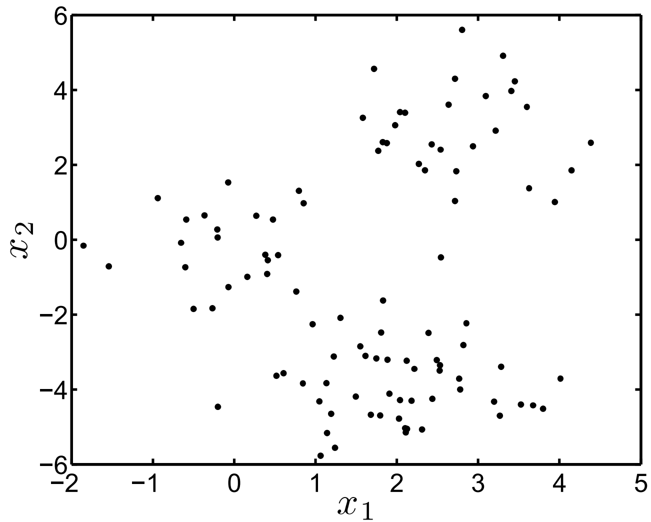
- Sensitive to initialisation.
- How do we choose K?
  - Tricky: Quantity above always decreases as K increases.
  - Can use CV if we have a measure of 'goodness'.
  - For clustering these will be application specific.

Mixture models – thinking generatively



- Could we hypothesis a model that could have created this data?

Mixture models – thinking generatively



- Could we hypothesis a model that could have created this data?
- Each $\mathbf{x}_n$ seems to have come from one of three distributions.