Clustering algorithms

Konstantinos Koutroumbas

Unit 6

– Fuzzy CFO clustering algorithms
– Optimization theory: basic concepts

koutroum@noa.gr

Fuzzy CFO clustering algorithms

Fuzzy clustering algorithms:

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ be a set of data points.

Each vector $\boldsymbol{x}_i$ **belongs** to all clusters up to a certain degree, $u_{ij}, j = 1, \dots, m$,
Subject to the constraints

- $u_{ij} \in [0,1], \; i = 1, \dots, N, j = 1, \dots, m$
- $\sum_{j=1}^{m} u_{ij} = 1 \,, \; i = 1, \dots, N$
- $0 < \sum_{i=1}^{N} u_{ij} < N, \; j = 1, \dots, m$

Each cluster is **represented** by a representative $\boldsymbol{\theta}_j$ (point repr., hyperplane…).
Let $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m\}$

Define the cost function

$$J_q(U, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^q \, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right), \qquad (q > 1)$$

When $J_q(U, \Theta)$ is **minimized**?

When large u_{ij}'s are **multiplied** with small $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$'s.

Fuzzy CFO clustering algorithms

Minimizing the cost function

$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q \, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed for **both** of them.

For **fixed $\boldsymbol{\theta}_j$'s:** Define the Lagrangian function

$$\mathcal{L}_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q \, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right) - \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{m} u_{ij} - 1 \right)$$

Equating the partial derivative of $\mathcal{L}_q(U,\Theta)$ wrt u_{rs} to 0, it turns out that

$$\frac{\partial \mathcal{L}_q(U,\Theta)}{\partial u_{rs}} = 0 \Leftrightarrow u_{rs} = \frac{1}{\sum_{j=1}^{m} \left(\frac{d(\boldsymbol{x}_r, \boldsymbol{\theta}_s)}{d(\boldsymbol{x}_r, \boldsymbol{\theta}_j)} \right)^{\frac{1}{q-1}}}$$

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$$

Fuzzy CFO clustering algorithms

Generalized Fuzzy Algorithmic Scheme (GFAS)

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \dots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of* $u_{ij}'s$
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^{q}(t-1) d(\boldsymbol{x}_i, \boldsymbol{\theta}_j), j = 1, \dots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Remarks:

- A candidate termination condition is
$$||\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)|| < \varepsilon,$$
where $||\cdot||$ is any vector norm and ε a user-defined constant.

- GFAS may also be initialized from $U(0)$ instead of $\boldsymbol{\theta}_j(0), j = 1, \ldots, m$ and start iterations with computing $\boldsymbol{\theta}_j$ first.

- If a point $\boldsymbol{x}_i$ **coincides** with one or more representatives, then it is shared arbitrarily among the clusters whose representatives coincide with $\boldsymbol{x}_i$, s.t. the constraint that the summation of all u_{ij}'s sum to 1.

- The degree of membership of $\boldsymbol{x}_i$ in C_j cluster is related to the grade of membership of $\boldsymbol{x}_i$ in rest $m-1$ clusters.

- If $q = 1$, **no** fuzzy clustering is better than the best hard clustering in terms of $J_q(\Theta, U)$.
- If $q > 1$**, there are** fuzzy clusterings with lower values of $J_q(\Theta, U)$ than the best hard clustering.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The point representatives case

- Point representatives are used in the case of compact clusters.
- Each $\boldsymbol{\theta}_j$ consists of l parameters.
- Any dissimilarity measure $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ between two points can be used.
- Common choices for $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ are

$$d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = (\boldsymbol{x}_i - \boldsymbol{\theta}_j)^T A(\boldsymbol{x}_i - \boldsymbol{\theta}_j),$$

where A is symmetric and positive definite matrix.

It is:

$$\frac{\partial d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 2A(\boldsymbol{\theta}_j - \boldsymbol{x}_i)$$

In this case the problem

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

is solved as

$$\frac{\partial}{\partial \boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = 0 \Leftrightarrow 2A \sum_{i=1}^{N} u_{ij}{}^q (\boldsymbol{\theta}_j - \boldsymbol{x}_i) = 0 \Leftrightarrow$$

$$\boldsymbol{\theta}_j = \frac{\sum_{i=1}^{N} u_{ij}{}^q \boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q}$$

Fuzzy CFO clustering algorithms

GFAS – The point representative with squared Mahalanobis distance

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \ldots, m$.
- $t = 0$
- **Repeat**

  – For $i = 1$ to N *% Determination of $u_{ij}'s$*

  o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

  o End {For-j}

  – End {For-i}

  $-t = t + 1$

  – For $j = 1$ to m *% Parameter updating*

  o Set

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}, j = 1, \ldots, m$$

  – End {For-j}

- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The point representatives case

Remarks:

- GFAS with the Euclidean distance ($A = I$) is also known as Fuzzy c-Means (FCM) or Fuzzy k-Means algorithm.

- FCM converges to a stationary point of the cost function or it has at least one subsequence that converges to a stationary point. This point may be a local (or global) minimum or a saddle point.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The point representatives case

Example:

Generate and plot the data set X7, which consists of N=216 2-dim. vectors. The first 100 stem from the normal distribution with mean $\boldsymbol{m}_1=[0, 0]^T$, the next 100 stem from the normal distribution with mean $\boldsymbol{m}_2=[13, 13]^T$. The other two groups of eight points each stem from the normal distribution with means $\boldsymbol{m}_3=[0, -40]^T$ and $\boldsymbol{m}_4=[-30, -30]^T$, respectively. The covariance matrices for all distributions are all equal to the 2x2 identity matrix. Obviously, the last two groups of points may be considered as outliers.

Apply the FCM on the data set X7 with m=2 clusters, plot the results and comment on the grade of memberships of the vectors to the two obtained clusters.

Apply also the k-means and the PAM on X7 and compare the results obtained from the three algorithms. (SEE attached code)

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- Here the representatives are quadric surfaces (hyperellipsoids, hyperparaboloids, etc.)
- **First issue:** How to represent them?
- General forms of an equation describing a quadric surface Q:

1. $\boldsymbol{x}^T A \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c = 0$,

where A is an $l \times l$ symmetric matrix, $\boldsymbol{b}$ is an $l \times 1$ vector, c is a scalar and $\boldsymbol{x} = [x_1, \ldots, x_l]^T$.

For various choices of A, $\boldsymbol{b}$ and c we obtain hyperellipses, hyperparabolas and so on.

NOTE: The above representations of Q are equivalent.

2. $\boldsymbol{q}^T \boldsymbol{p} = 0$,

where

$$\boldsymbol{q} = [x_1{}^2, x_2{}^2, \ldots, x_l{}^2, x_1 x_2, \ldots, x_{l-1} x_l, x_1, x_2, \ldots, x_l, 1]^T$$

and

$$\boldsymbol{p} = [p_1, p_2, \ldots, p_l, p_{l+1}, \ldots, p_r, p_{r+1}, \ldots, p_s]^T$$

with $r = \frac{l(l+1)}{2}$ and $s = r + l + 1$.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Second issue:** *"Definition of the distance of a point* $\boldsymbol{x}$ *to a quadric surface* Q*"*

Types of distances

– Perpendicular distance:

$$d_p{}^2(\boldsymbol{x}, Q) = min_{\boldsymbol{z}} \|\boldsymbol{x} - \boldsymbol{z}\|^2,$$

subject to the constraint

$$\boldsymbol{z}^T A\boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c = 0$$

In words, $d_p{}^2(\boldsymbol{x}, Q)$ is the distance between $\boldsymbol{x}$ and the closest to $\boldsymbol{x}$ point that lies in Q.

Prove it for the $l = 2$ case.

– (Squared) Algebraic distance:

$$d_p{}^2(\boldsymbol{x}, Q) = (\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c)^2 \equiv \boldsymbol{p}^T M\boldsymbol{p}$$

where $M = \boldsymbol{q}\boldsymbol{q}^T$.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

– Radial distance (**only** when Q is a hyperellipsoidal):

For Q hyperellipsoidal, the representative equation can be written as

$$(\boldsymbol{x}-\boldsymbol{c})^T A(\boldsymbol{x}-\boldsymbol{c}) = 1$$

where $\boldsymbol{c}$ is the center of the ellipse and A a **positive definite symmetric** matrix defining major axis, minor axis and orientation.

Then the radial distance is defined as

$$d_r{}^2(\boldsymbol{x}, Q) = \|\boldsymbol{x}-\boldsymbol{z}\|^2$$

subject to the constraints

$$(\boldsymbol{z}-\boldsymbol{c})^T A(\boldsymbol{z}-\boldsymbol{c}) = 1$$

and

$$(\boldsymbol{z}-\boldsymbol{c}) = a(\boldsymbol{x}-\boldsymbol{c}).$$

In words,

–the intersection point $\boldsymbol{z}$ between the line segment $\boldsymbol{x}-\boldsymbol{c}$ and Q is determined

–the $d_r{}^2(\boldsymbol{x}, Q)$ is defined as the squared Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{z}$.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

– (Squared) Normalized radial distance (only when Q is a hyperellipsoidal):

$$d_{nr}{}^2(\boldsymbol{x}, Q) = \left(\left((\boldsymbol{x}-\boldsymbol{c})^T A(\boldsymbol{x}-\boldsymbol{c})\right)^{1/2} - 1\right)^2$$

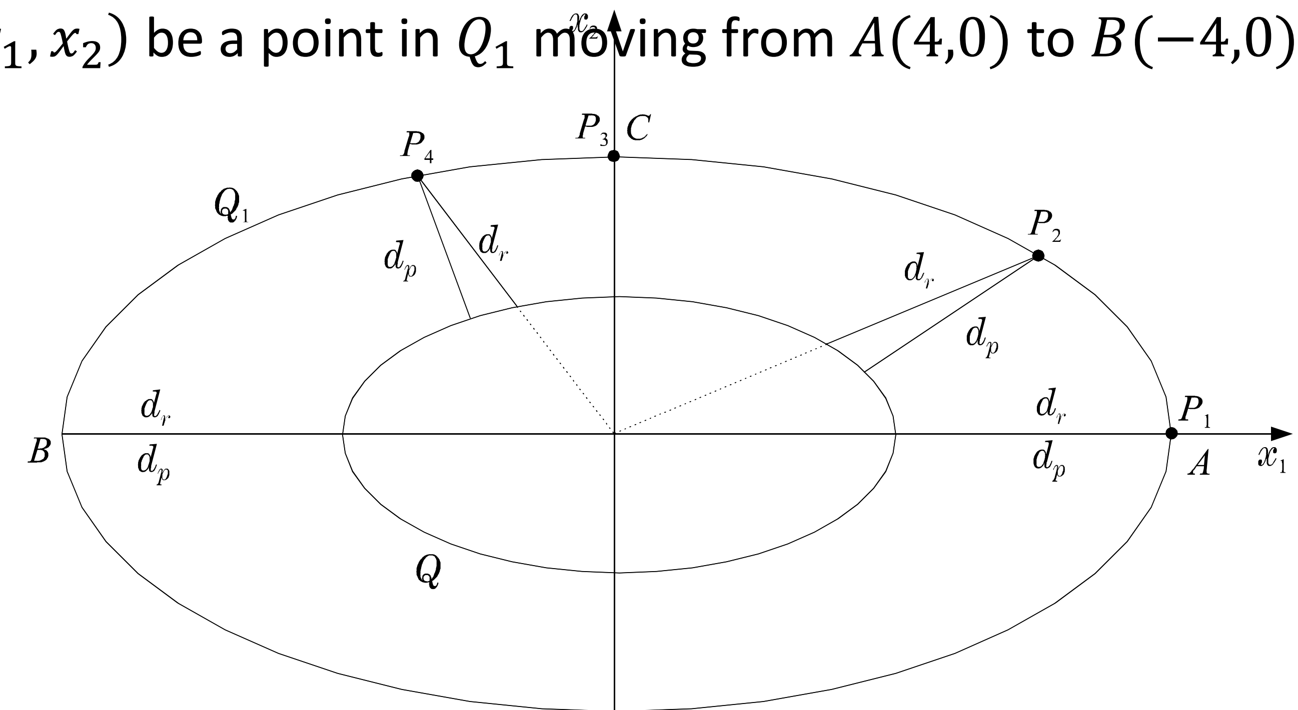
$d_r{}^2(\boldsymbol{x}, Q) = d_{nr}{}^2(\boldsymbol{x}, Q)\|\boldsymbol{x}-\boldsymbol{z}\|^2$
$\boldsymbol{z}$: intersection of $\boldsymbol{x}-\boldsymbol{c}$ and Q.

– Example 3:

- Consider two ellipses Q and Q_1, centered at $\boldsymbol{c} = [0,0]^T$, with

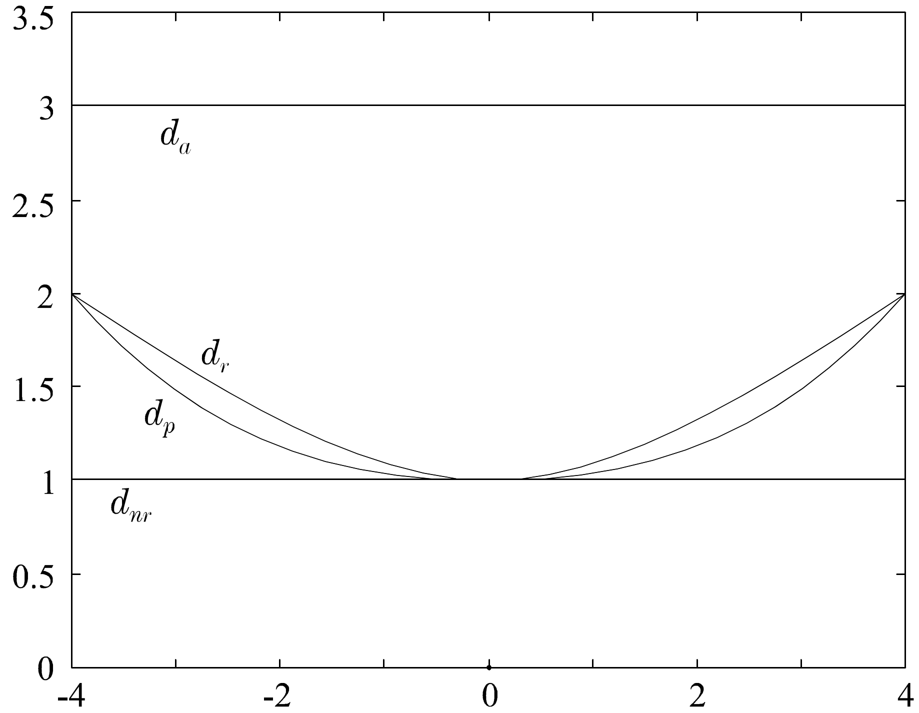
$A = diag(\frac{1}{4}, 1)$ and $A_1 = diag\left(\frac{1}{16}, \frac{1}{4}\right)$, respectively.

- Let $P(x_1, x_2)$ be a point in Q_1 moving from $A(4{,}0)$ to $B(-4{,}0)$, with $x_2 > 0$



Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case



Remarks:

- d_a and d_{nr} **do not vary** as P moves.
- d_r can be used as an **approximation** of d_p, when Q is a hyperellipsoid.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

Recall that

$$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} {u_{ij}}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \text{ s.t. } \sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$$

- The algorithms in this case fall under the umbrella of GFAS.
- They all **share** the same rule for updating the matrix U.
- They **differ** on the choice of the distance between a point and the representative of a quadric surface.

  $\Rightarrow$ they **differ** in the representatives updating part.
- *At each iteration*, the updating of the representatives is carried out by **setting** the gradient of J_q wrt them equal to $\mathbf{0}$ (for **fixed** u_{ij}'s) and **solving** (usually using *iterative schemes*) for the involved parameters.

Fuzzy CFO clustering algorithms

Generalized Fuzzy Algorithmic Scheme (GFAS)

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \ldots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u_{ij}'s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} {u_{ij}}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j), j = 1, \ldots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^q \, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Algorithms:

- *Fuzzy C Ellipsoidal Shells* (FCES) Algorithm:

- It **adopts** the radial distance between a vector and the surface representative

- It **recovers** only ellipsoidal clusters.

- *Fuzzy C Quadric Shells* (FCQS) Algorithm:

- It **adopts** the algebraic distance between a vector and the surf. repr. in the form ${d_a}^2(\boldsymbol{x}, Q) = \boldsymbol{p}^T M \boldsymbol{p}$, **imposing** constraints on vector $\boldsymbol{p}$.

- It **recovers** quadric clusters of any kind (ellipsoidal, hyperbolical, paraboloidal, pairs of lines).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Algorithms:

- *Modified Fuzzy C Quadric Shells* (MFCQS) Algorithm:

- It **adopts** :
  - the perpendicular distance between a vector and the surface representative for the updating of matrix U
  - The algebraic distance between a vector and the surface representative for the updating of the cluster representatives.
- It **recovers** quadric clusters of any kind (ellipsoidal, hyperbolical, paraboloidal, pairs of lines).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- Here the representatives are hyperplanes (lines in the 2-D space, planes in the 3-D space etc.)
- **First issue:** How to represent them?

1. Via the equation of a hyperplane H:

$$H\colon \boldsymbol{\theta}^T \boldsymbol{x} + \theta_0 = 0,$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_l]^{\boldsymbol{T}}$, $\boldsymbol{x} = [x_1, x_2, \dots, x_l]^{\boldsymbol{T}}$.

2. Via a center $\boldsymbol{c}_j$ and a covariance matrix Σ_j, that is, $\boldsymbol{\theta}_j = (\boldsymbol{c}_j, \Sigma_j)$.

NOTE: Another choice for **representing** such clusters is by using line segments. (only for the *2-D case*).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- **Second issue**: *"Definition of the distance of a point* $\boldsymbol{x}$ *to a cluster"*

Types of distances

– Distance of a point from a hyperplane:

$$d(\boldsymbol{x}, H) = \frac{|\boldsymbol{\theta}^T \boldsymbol{x} + \theta_0|}{||\boldsymbol{\theta}||}$$

– GK distance:

$$d_{GK}{}^2(\boldsymbol{x}, \boldsymbol{\theta}_j) = \left|\Sigma_j\right|^{1/l} \left(\boldsymbol{x} - \boldsymbol{c}_j\right)^T \Sigma_j{}^{-1} \left(\boldsymbol{x} - \boldsymbol{c}_j\right)$$

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- **Third issue:** Choice of algorithm.

Recall that

$$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \text{ s.t. } \sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$$

- The algorithms in this case fall under the umbrella of GFAS.
- They all **share** the same rule for updating the matrix U.
- They **differ** on the choice of the distance between a point and the representative of a plane cluster.

  $\Rightarrow$ they **differ** in the representatives updating part.
- *At each iteration*, the updating of the representatives is carried out by **setting** the gradient of J_q wrt them equal to $\mathbf{0}$ (for **fixed** $\boldsymbol{u_{ij}}$**'s**) and **solving** (usually using *iterative schemes*) for the involved parameters.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^{q}\, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Algorithms:

- *Fuzzy C varieties* (FCV) Algorithm:

- It **adopts** the classical distance between a point and a hyperplane.

- *Disadvantages*:
  - It tends to **recover** very long clusters and, thus, collinear distinct clusters may be **merged** to a single one.
  - If, at a certain iteration, a hyperplane representative crosses two distinct clusters, there is no way to recover from this situation.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^q\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Algorithms:

- *Gustafson-Kessel* (GK) algorithm:

- It **adopts** the GK distance between a point and a cluster.
- The parameter updating takes place via the following two equations

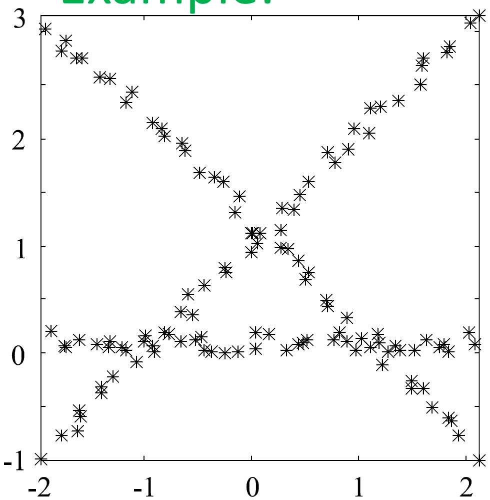
$$\boldsymbol{c}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)\,\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}$$

$$\Sigma_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)(\boldsymbol{x}_i - \boldsymbol{c}_j(t))(\boldsymbol{x}_i - \boldsymbol{c}_j(t))^T}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}$$

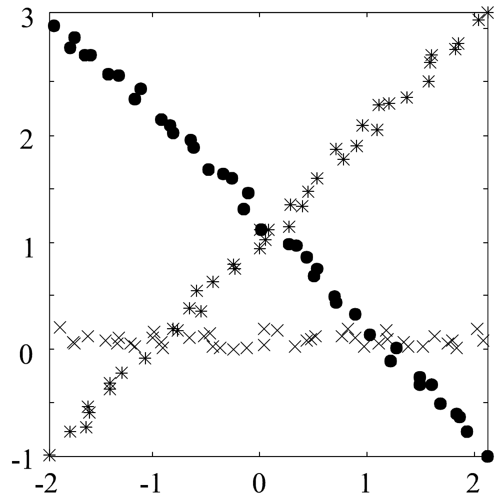
Fuzzy CFO clustering algorithms

<u>Fuzzy Clustering – The quadric surfaces representatives case</u>

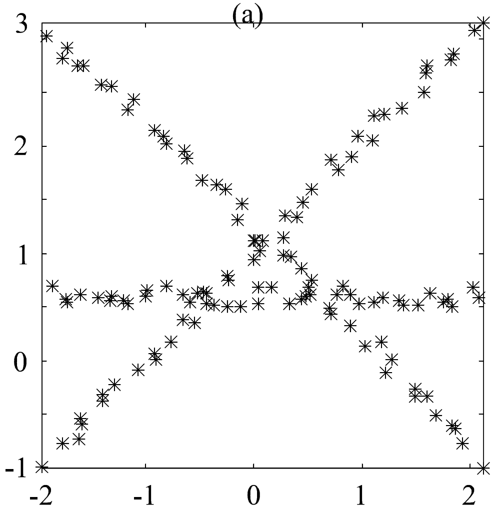
- *<u>Gustafson-Kessel</u>* (GK) algorithm (cont.):
  - Example:



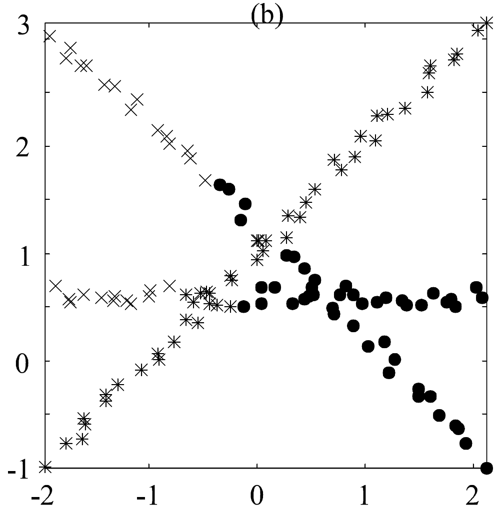
(a)



(b)



(a)



(b)

Comments:

In the **first case**, the clusters are well discriminated and the GK-algorithm **recovers** them correctly.

In the **second case**, the clusters are not well discriminated and the GK-algorithm **fails** to recover them correctly.

Optimization theory – Basic concepts

Let $J(\boldsymbol{w})$ be a continuous function of $\boldsymbol{w}$.

Problem (P1): Determine the position $\boldsymbol{w}^*$ where the function $J(\boldsymbol{w})$ achieves its minimum value.

A simple method for solving **(P1)** is that of gradient descent.

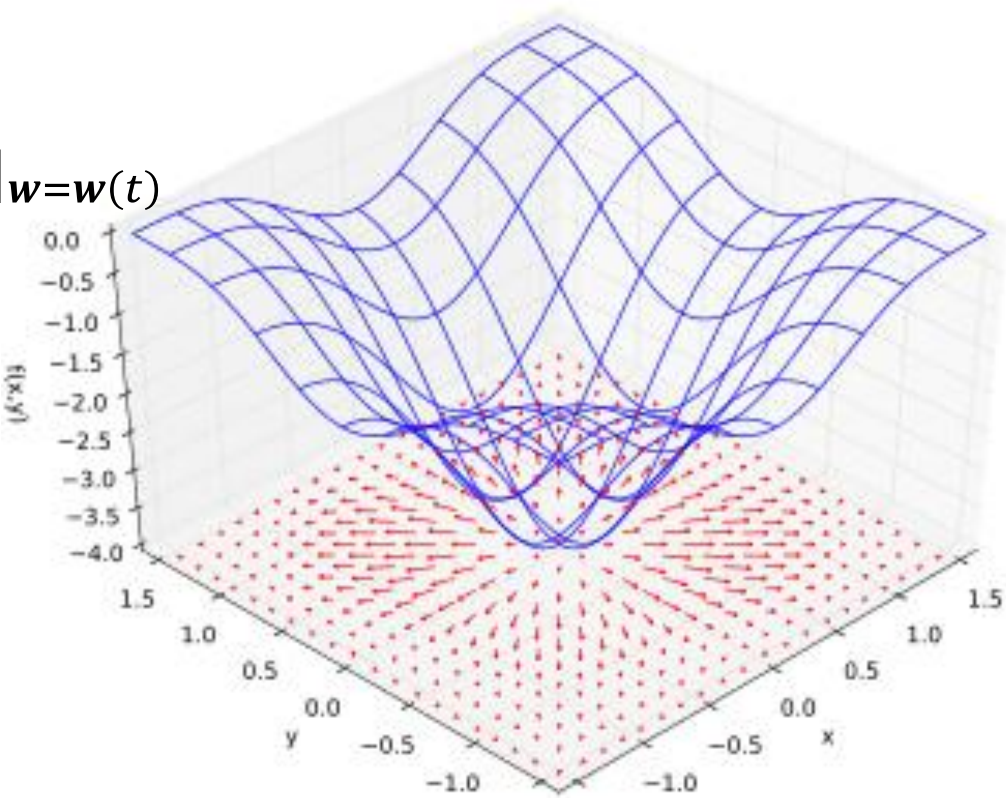
-**Initialize** $\boldsymbol{w} = \boldsymbol{w}(0)$

-$t = 0$

-**Repeat**

- $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \mu \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} |_{\boldsymbol{w}=\boldsymbol{w}(t)}$
- $t = t + 1$

-**Until** convergence



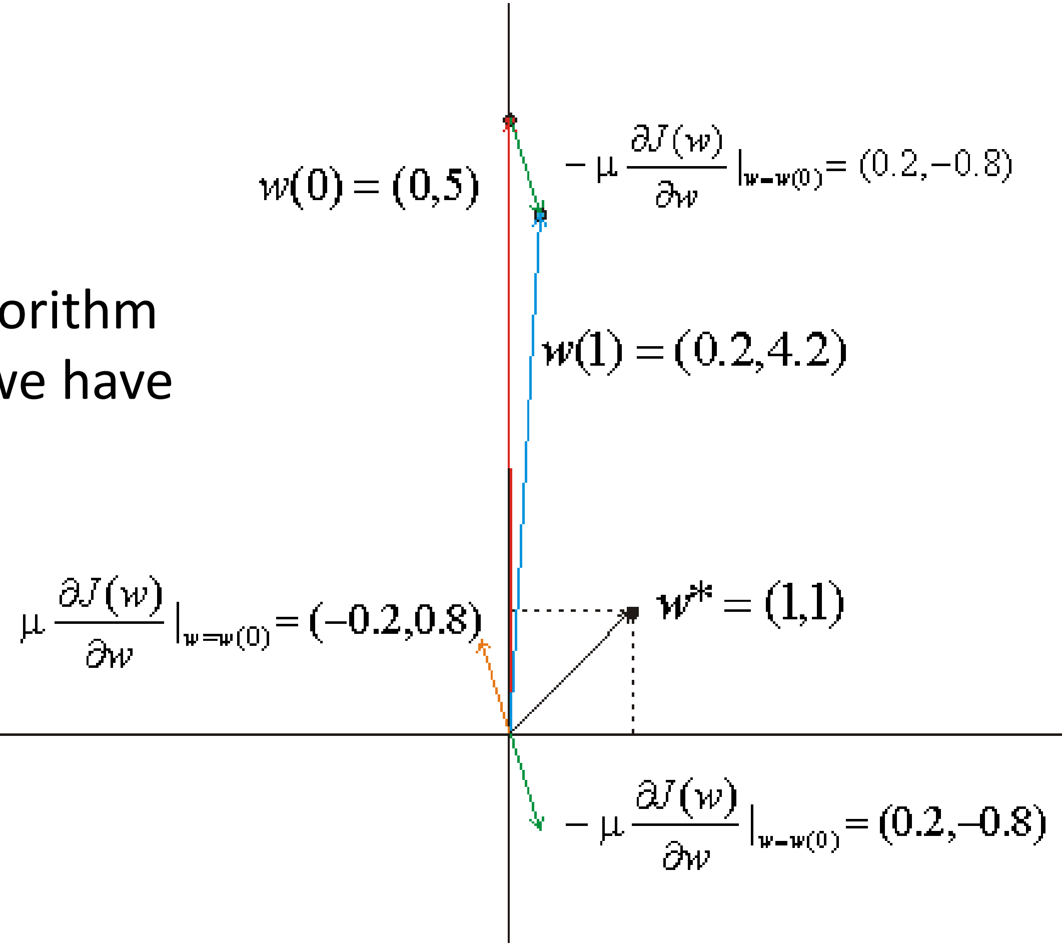
-**An example:** Let $\boldsymbol{w} = [w_1, w_2]^T$ and $J(\boldsymbol{w}) = (w_1 - 1)^2 + (w_2 - 1)^2$. Clearly, the minimum value of $J(\boldsymbol{w})$ is met at $\boldsymbol{w}^* = [1, 1]^T$.

-It is $\dfrac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \begin{bmatrix} 2w_1 - 2 \\ 2w_2 - 2 \end{bmatrix}$

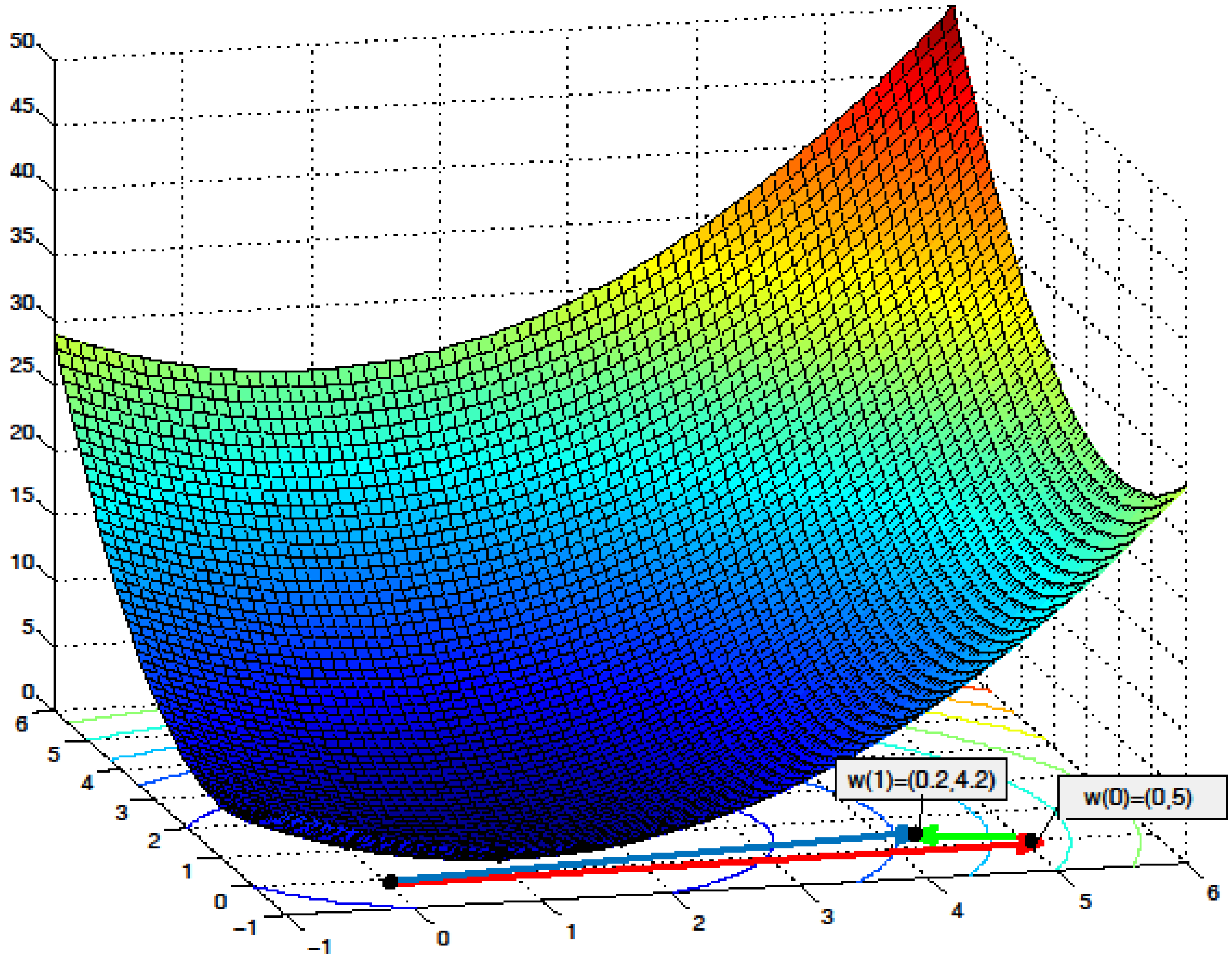
-Applying the gradient descent algorithm for $\boldsymbol{w}(0) = [0, 5]^T$, and $\mu = 0.1$, we have

$$\boldsymbol{w}(1) = \begin{bmatrix} 0 \\ 5 \end{bmatrix} - 0.1 \begin{bmatrix} -2 \\ 8 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 4.2 \end{bmatrix}$$

-Thus, $\boldsymbol{w}(1)$ comes closer to $\boldsymbol{w}^*$.



Optimization theory – Basic concepts



Optimization theory – Basic concepts

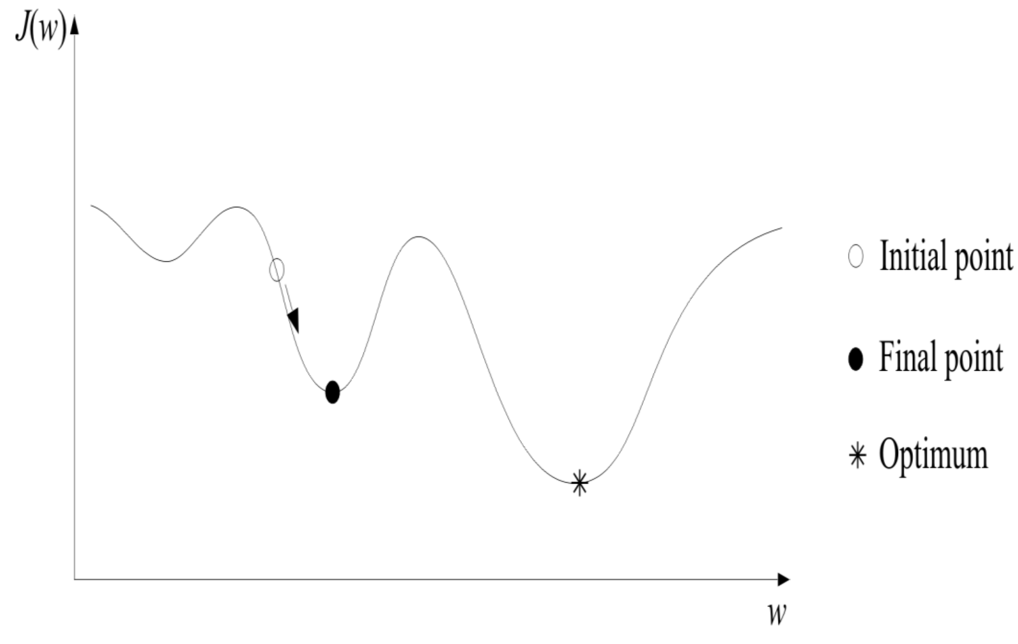
Remarks for gradient descent:

-The value of μ should be chosen not too large, in order to avoid oscillations around the minimum and not too small in order to avoid unnecessary delays in the convergence

-If $J(\boldsymbol{w})$ has more than one local minima, the gradient descent will converge (in general) to the one that is closest to $\boldsymbol{w}(0)$.

-If the algorithm is trapped to a local minimum *that correspond to a poor solution*, the only way to escape from it is to re-initialize the algorithm from another initial position.

-It can be proved that, under certain conditions, the algorithm converges asymptotically to a local minimum of $J(\boldsymbol{w})$.

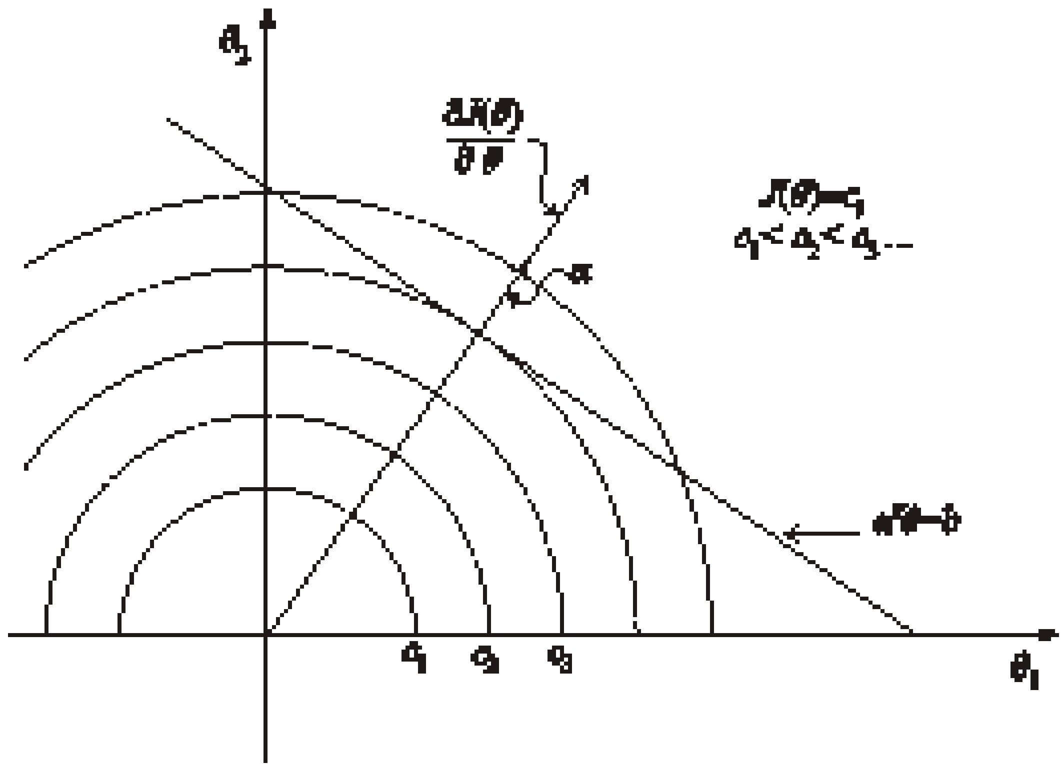


Optimization theory – Basic concepts

Let $J(\boldsymbol{w})$ be a continuous function of $\boldsymbol{w}$.
Problem (P2): Determine the position $\boldsymbol{w}^*$ where the function $J(\boldsymbol{w})$ achieves its minimum value, under the constraint that $\boldsymbol{w}$ satisfies some equality constraints.

For linear equality constraints, the problem is stated as follows

- Minimize $J(\boldsymbol{w})$
- Subject to the constraints $A\boldsymbol{w} = \boldsymbol{b}$, where A an mxl matrix and $\boldsymbol{b}$ an m-dim. Vector.



Solution: Lagrange multipliers
Minimize
- $L(\boldsymbol{w}) = J(\boldsymbol{w}) + \boldsymbol{\lambda}^{\mathrm{T}}(A\boldsymbol{w} - \boldsymbol{b})$
- $\boldsymbol{\lambda}$ is an m-dim vector that is estimated through the constraints $A\boldsymbol{w} = \boldsymbol{b}$

Optimization theory – Basic concepts

Let $J(\boldsymbol{w})$ be a continuous function of $\boldsymbol{w}$.

Problem (P3): Determine the position $\boldsymbol{w}^*$ where the function $J(\boldsymbol{w})$ achieves its minimum value, under the constraint that $\boldsymbol{w}$ satisfies some inequality constraints.

For linear inequality constraints, the problem is stated as follows

- Minimize $J(\boldsymbol{w})$
- Subject to the constraints
  $A\boldsymbol{w} \geq \boldsymbol{b}$, where A an mxl matrix and $\boldsymbol{b}$ an m-dim. Vector.