

A Graphical Shopping Interface based on Product Attributes

Authors:

Martijn Kagie

Michael van Wezel

Patrick J.F. Groenen

presentation prepared by:

Patrick Lamber

Diana Zverelo

Outline

- **Problem definition**
- Recommender Systems
- Methodology
- Graphical Recommender System
- Graphical Shopping Interface
- Evaluation of the Graphical Shopping Interface
- Conclusions

Problem definition (1/2)

- Recommender systems lose much information about the mutual similarity between two or more products
 - Paradox of choice: more difficult to find an ideal product when there are too many options
- Users normally search by
 - Limited filter criteria search
 - Let the customer describe the "ideal" product
 - Disadvantage
 - Products with same similarity can differ on a completely different set of attributes

Problem definition (2/2)

- New approach: use a 2D visualization to show those differences
 - Show similar products near to each other
- Two prototypes provided
 - Graphical recommender system (GRS)
 - Graphical shopping interface (GSI)
- Inspirations taken from the field of industrial design engineering
 - Explore databases in an interactive way

Outline

- Problem definition
- **Recommender Systems**
- Methodology
- Graphical Recommender System
- Graphical Shopping Interface
- Evaluation of the Graphical Shopping Interface
- Conclusions

Recommender Systems (1/3)

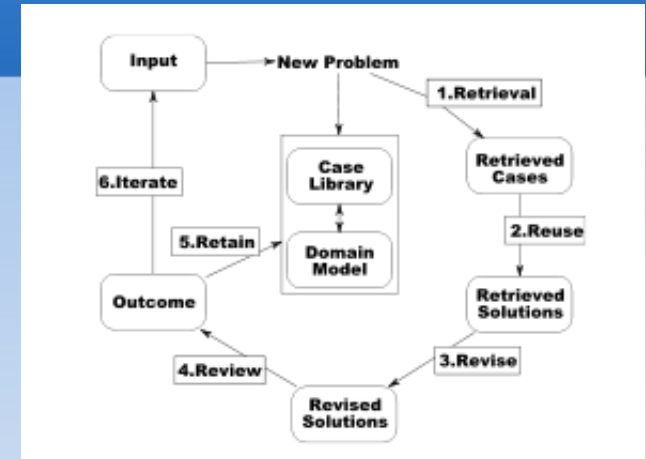
- Systems that are used by E-Commerce sites to suggest products to their customers and to provide consumers with information to help them decide which products to purchase” [Shafer et al., 2001]
- Suggestions
 - the same for all users
 - **dependent on the user's preferences**
 - using past purchases
 - navigation behaviour
 - rating systems
 - asking his preferences directly

Recommender Systems (2/3)

- Types of recommender Systems:
 - **Content-based** (suggests products that are similar to the products the customer liked in the past)
 - Collaborative filtering (suggest products that other people with similar taste bought or liked in the past)
 - Hybrid approaches

Content-based recommender Systems (3/3)

- Case-based reasoning (CBR)
- not all steps have to be implemented by case-based reasoning recommender system (CBR-RS)
- data is stored in *case library*
- *domain model* consists of features describing at least one of sub models (**content model**, user model, session model, evaluation model)
- CBR-RS gives recommendation based on:
 - similarity between cases in the case library
 - problem (input of the customer)



Outline

- Problem definition
- Recommender Systems
- **Methodology**
- Graphical Recommender System
- Graphical Shopping Interface
- Evaluation of the Graphical Shopping Interface
- Conclusions

Methodology – (dis)similarity measure (1/4)

- products $\{x_i\}_i^n$ in data set D
- products have K attributes: $x_i=(x_{i1},x_{i2},\dots,x_{iK})$
- attributes have mixed types: numerical, binary or categorical
- (dis)similarity measures, like Euclidean distance, Person's correlation coefficient, and Jaccard's similarity measure are to handle **one attribute type**
- general coefficient of similarity proposed by Gower can cope with **mixed attribute types**
- Similarity s_{ij} between products i and j is the average of the nonmissing similarity scores s_{ijk} over attributes, where m_{ik} is 0 when the value for attribute K is missing, 1 when not missing

$$s_{ij} = \frac{\sum_{k=1}^K m_{ik}m_{jk}s_{ijk}}{\sum_{k=1}^K m_{ik}m_{jk}}$$

Methodology – (dis)similarity measure (2/4)

- Similarity score s_{ijk} depends upon the type of the attribute
 - for numerical attributes s_{ijk} is based on the absolute distance divided by the range

$$s_{ijk}^N = 1 - \frac{|x_{ik} - x_{jk}|}{\max(\mathbf{x}_k) - \min(\mathbf{x}_k)}$$

where \mathbf{x}_k is a vector containing the values of the k^{th} attribute for all n products

- for binary and categorical attributes s_{ijk} is defined as

$$s_{ijk}^C = 1(x_{ik} = x_{jk})$$

objects having the same category value get similarity score 1, and 0 otherwise

Methodology – (dis)similarity measure (3/4)

- adaptations have to be made:
 - the similarity has to be transformed to a dissimilarity
 - some variables are more important than others
 - influence of categorical/binary attributes on the general coefficient turns out to be too large.
- the following adaptations are made
 - both types of dissimilarity scores are normalized to have an average dissimilarity score of 1 between two different objects
 - since $\delta_{ij} = \delta_{ji}$, dissimilarities having $i \geq j$ are excluded from the sum without loss of generality

Methodology – (dis)similarity measure (4/4)

- the numerical dissimilarity score

$$\delta_{ijk}^N = \frac{|x_{ik} - x_{jk}|}{\left(\sum_{i < j} m_{ik} m_{jk}\right)^{-1} \sum_{i < j} m_{ik} m_{jk} |x_{ik} - x_{jk}|}$$

- the categorical dissimilarity score

$$\delta_{ijk}^C = \frac{1(x_{ik} \neq x_{jk})}{\left(\sum_{i < j} m_{ik} m_{jk}\right)^{-1} \sum_{i < j} m_{ik} m_{jk} 1(x_{ik} \neq x_{jk})}$$

- the combined dissimilarity measure

$$\delta_{ij} = \sqrt{\frac{\sum_{k \in C} w_k m_{ik} m_{jk} \delta_{ijk}^C + \sum_{k \in N} w_k m_{ik} m_{jk} \delta_{ijk}^N}{\sum_{k=1}^K w_k m_{ik} m_{jk}}}$$

vector w is incorporated to emphasize attributes differently

Methodology – Multidimensional Scaling

- dissimilarity scores are used to represent products in 2D space
- low dimensional Euclidean representation can be formalized by minimizing the raw Stress function

$$\sigma_r(\mathbf{Z}) = \sum_{i < j} (\delta_{ij} - d_{ij}(\mathbf{Z}))^2$$

where the matrix \mathbf{Z} is the $n \times 2$ coordinate matrix representing the n products in two dimensions. δ_{ij} is dissimilarity between objects i and j . $d_{ij}(\mathbf{Z})$ is the Euclidean distance between row points i and j

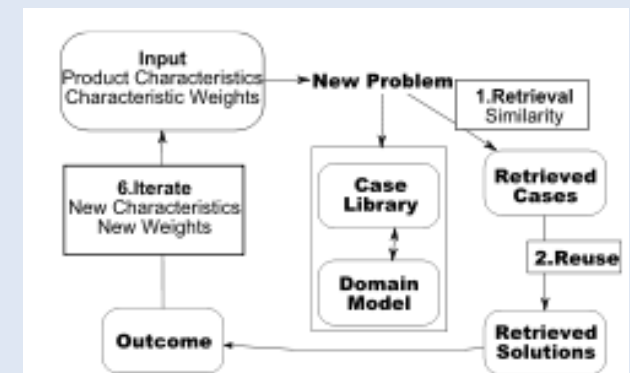
- To minimize $\delta_r(\mathbf{Z})$ SMACOF algorithm based on majorization can be used.

Outline

- Problem definition
- Recommender Systems
- Methodology
- **Graphical Recommender System**
- Graphical Shopping Interface
- Evaluation of the Graphical Shopping Interface
- Conclusions

GRS (1/2)

- *input* is the ideal product described by the customer
- *new problem* is constructed
- in the *retrieval phase*, a set of cases from the *case library* is selected and is *reused* as solutions (*outcome*)
- if results are not satisfying, user adapts his product description or weights of attributes to start the process again in the *iterate* step



GRS (algorithm) (2/2)

- compute weighted dissimilarities δ_{i*} between x^* and all x_i in data set D .
- $p-1$ products are selected that are most similar to x^*
- $p-1$ selected products are combined with x^* in D^* and dissimilarities are computed again
- $p \times p$ matrix Δ^* with dissimilarities between products is constructed and is an input for MDS algorithm
- The algorithm returns the $p \times 2$ coordinate matrix Z

Outline

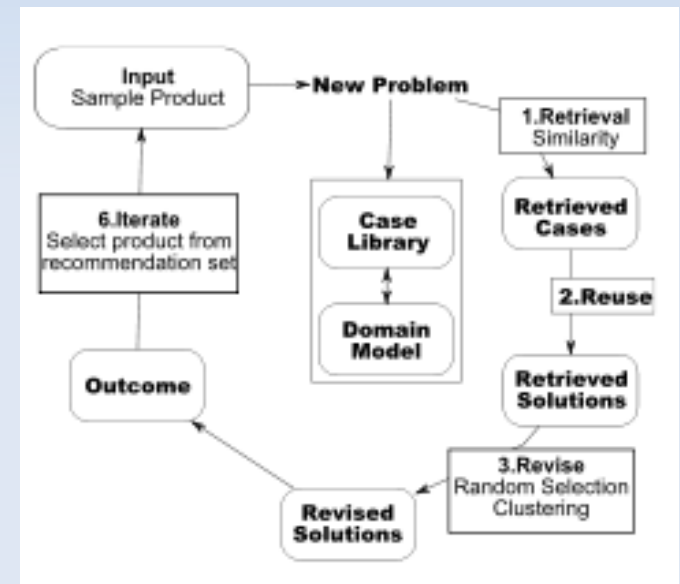
- Problem definition
- Recommender Systems
- Methodology
- Graphical Recommender System
- **Graphical Shopping Interface**
- Evaluation of the Graphical Shopping Interface
- Conclusions

GSI (1/10)

- Customers not always know what exactly they want
- Help them by providing a set of potential products and let them navigate through the product space
- Problem
 - Which products should be shown to the customer at the beginning?

GSI (2/10)

- *input* is a product, selected in the 2D space that was created in the previous iteration. It is a new *problem*
- in the *retrieval* phase, a large set of cases that is most similar to the input is selected
- set is *reused* as solution
- In the *revise* stage, a smaller subset of products is chosen. New set is shown in 2D space
- user selection is new input for next iteration
- Implementation not trivial of *revise* step
 - the random system
 - the clustering system
 - hierarchical system



GSI – random system (3/10)

- First iteration is an initialization iteration
 - D will contain the complete case library
 - Select p products at random (without replacement)
- Next iteration after new user selection
 - Take smaller D, with size $\max(p - 1, a^t n - 1)$
 - Select p random products in D and compute dissimilarity matrix
 - Compute MDS to get 2-dimensional representation.

GSI - random system (4/10)

Algorithm 1 GSI implementation using random selection

procedure RANDOM_GSI(D, p, α)

$D_0 = D$.

Generate random $D_0^* \subset D_0$ with size p .

Compute Δ_0^* given D_0^* using (6).

Compute Z_0 given Δ_0^* using MDS.

$t = 0$.

repeat

$t = t + 1$.

Select a product $\mathbf{x}_t^* \in D_{t-1}^*$.

Get $D_t \subset D$ containing $\max(p - 1, \alpha^t n - 1)$ products most similar to \mathbf{x}_t^* using (6).

Generate random $D_t^* \subset D_t$ with size $p - 1$.

$D_t^* = D_t^* \cup \mathbf{x}_t^*$.

Compute Δ_t^* given D_t^* using (6).

Compute Z_t given Δ_t^* using MDS.

until $D_t^* = D_{t-1}^*$.

end procedure

GSI – cluster system (5/10)

- Random selection replaced by a clustering solution
- Hierarchical clustering method (average linkage algorithm)
 - Calculate dissimilarity matrix
 - Clusters are calculated based on it (tree of clusters)
 - System only uses the p clusters solution in the tree
 - Prototypical product selected in each cluster for p clusters

GSI – cluster system (6/10)

- Prototypical product
 - Smallest total dissimilarity to the other products in the cluster

$$i_c = \arg \min_i \sum_{j=1}^{nc} \delta_{ij}^c$$

- Get all prototypical products and compute the dissimilarity matrix in combination with MDS
- Disadvantage
 - Becomes quite slow as product space gets larger

GSI - clustering system (7/10)

Algorithm 2 GSI implementation using clustering

```
procedure CLUSTERING_GSI( $D, p, \alpha$ )  
   $D_0 = D$ .  
  Compute  $\Delta_0$  given  $D_0$  using (6).  
  Compute  $T_0$  given  $\Delta_0$  using average linkage.  
  Find  $p$  clustering solution in  $T_0$ .  
  Determine prototypical products of clusters using (8).  
  Store prototypical products in  $D_0^*$ .  
  Compute  $\Delta_0^*$  given  $D_0^*$  using (6).  
  Compute  $Z_0$  given  $\Delta_0^*$  using MDS.  
   $t = 0$ .  
  repeat  
     $t = t + 1$ .  
    Select a product  $\mathbf{x}_t^* \in D_{t-1}^*$ .  
    Get  $D_t \subset D$  containing  $\max(p - 1, \alpha^t n - 1)$  products most similar to  $\mathbf{x}_t^*$   
      using (6).  
    Compute  $\Delta_t$  given  $D_t$  using (6).  
    Compute  $T_t$  given  $\Delta_t$  using average linkage.  
    Find  $p$  clustering solution in  $T_t$ .  
    Determine prototypical products of clusters using (8).  
    Store prototypical products in  $D_t^*$ .  
    Compute  $\Delta_t^*$  given  $D_t^*$  using (6).  
    Compute  $Z_t$  given  $\Delta_t^*$  using MDS.  
  until  $D_t^* = D_{t-1}^*$ .  
end procedure
```

GSI – hierarchical system (8/10)

- Don't compute each time clusters
 - Compute one cluster at the beginning and reuse it
- First iteration take the root node of the computed cluster
- Next iterations
 - Go down the tree until we find the p cluster solution and get prototypical products
 - If such a cluster solution does not exist, show the remaining products
 - Compute dissimilarity matrix and use MDS

GSI – hierarchical system (9/10)

- Users selects X from this solution
 - the cluster it represents is the new root node for the next computations
- Procedure terminates when p is higher than the number of clusters

GSI - hierarchical system (10/10)

Algorithm 3 GSI implementation using hierarchical clustering.

procedure HIERARCHICAL_GSI(D, p)

 Compute Δ given D using (6).

 Compute T given Δ using average linkage.

$T_0 = T$.

$t = 0$.

$n_c = \text{size}(D)$.

repeat

 Find $\min(n_c, p)$ clustering solution in T_t .

 Determine prototypical products of clusters using (8).

 Store prototypical products in D_t^* .

 Compute Δ_t^* given D_t^* using (6).

 Compute \mathbf{Z}_t given Δ_t^* using MDS.

 Select $\mathbf{x}_t^* \in D_t^*$ and determine cluster D_t^c it represents.

$n_c = \text{size}(D_t^c)$.

D_t^c is root of T_{t+1} .

$t = t + 1$.

until $n_c \leq p$.

end procedure

Outline

- Problem definition
- Recommender Systems
- Methodology
- Graphical Recommender System
- Graphical Shopping Interface
- **Evaluation of the Graphical Shopping Interface**
- Conclusions

Evaluation of GSI (1/5)

- the quality of the 2D spaces was studied by considering the Stress values
- goal is to evaluate how easily a customer can find the product he wants using GSI
- normalization of Stress value:

$$\sigma_n = \frac{\sum_{i < j} (\delta_{ij} - d_{ij}(\mathbf{Z}))^2}{\sum_{i < j} \delta_{ij}^2}$$

Evaluation of GSI (2/5)

- Estimation of goodness of the representations in the GRS:

- one product from the data set is taken as an ideal product and all other products are case library
- all attributes are used to compute similarities and dissimilarities, weights are set to 1
- $p-1$ most similar products to this ideal product are selected and 2D space is created using MDS
- procedure is repeated, until each product has functioned once as an ideal product description
- procedure is done for $p = 2$ to 10
- results for the average normalized Stress values:

| p | Mean Normalized Stress |
|-----|------------------------|
| 2 | $3.36 \cdot 10^{-32}$ |
| 3 | $3.13 \cdot 10^{-4}$ |
| 4 | $5.77 \cdot 10^{-3}$ |
| 5 | $1.21 \cdot 10^{-2}$ |
| 6 | $1.96 \cdot 10^{-2}$ |
| 7 | $2.63 \cdot 10^{-2}$ |
| 8 | $3.16 \cdot 10^{-2}$ |
| 9 | $3.62 \cdot 10^{-2}$ |
| 10 | $4.05 \cdot 10^{-2}$ |

Evaluation of GSI (3/5)

- Estimation of the navigation in the different implementations of the GSI

- assumptions about the navigation behaviour of the user has to be made
 - customer explicitly or implicitly can specify what his ideal product looks like
 - user compares products using the same dissimilarity measure as the system uses
 - in each step the customer chooses the product that is most similar to the ideal product
- each time, one product is selected as the ideal product and all other products are used as the case library
- procedure is repeated, until every product is left out once
- evaluation is done on the three different implementations with p set to 4,6,8 and 10
- for the random and clustering system parameter α varies to the values 0.2, 0.4, 0.6 and 0.8
- before starting a single experiment, we determine which product in the case library is most similar to the product we left out. During each step in a single experiment we use the assumptions above to compute the product the user will select. We stop when the most similar product is in shown set

Evaluation of GSI (4/5)

Results for different specifications of the random system.

| p | α | Successes | In 5 Steps | Average number of Steps |
|-----|----------|-----------|------------|-------------------------|
| 4 | 0.2 | 16.8% | 16.8% | 5.02 |
| | 0.4 | 29.3% | 19.3% | 6.38 |
| | 0.6 | 41.7% | 10.0% | 9.12 |
| | 0.8 | 56.1% | 5.9% | 15.99 |
| 6 | 0.2 | 29.3% | 28.7% | 4.77 |
| | 0.4 | 42.7% | 34.9% | 5.83 |
| | 0.6 | 52.7% | 17.1% | 8.01 |
| | 0.8 | 70.1% | 10.9% | 13.12 |
| 8 | 0.2 | 43.0% | 42.7% | 4.34 |
| | 0.4 | 47.0% | 43.6% | 5.29 |
| | 0.6 | 66.4% | 30.5% | 6.96 |
| | 0.8 | 80.1% | 16.2% | 11.22 |
| 10 | 0.2 | 46.4% | 46.4% | 4.09 |
| | 0.4 | 58.9% | 56.7% | 4.82 |
| | 0.6 | 70.4% | 37.1% | 6.27 |
| | 0.8 | 83.8% | 19.9% | 9.92 |

Results for different specifications of the clustering system.

| p | α | Successes | In 5 Steps | Average number of Steps |
|-----|----------|-----------|------------|-------------------------|
| 4 | 0.2 | 14.6% | 14.6% | 4.85 |
| | 0.4 | 20.3% | 12.2% | 6.68 |
| | 0.6 | 19.3% | 8.7% | 8.90 |
| | 0.8 | 13.7% | 10.0% | 7.83 |
| 6 | 0.2 | 22.1% | 21.2% | 4.83 |
| | 0.4 | 32.7% | 27.1% | 6.17 |
| | 0.6 | 44.9% | 19.0% | 8.30 |
| | 0.8 | 25.9% | 11.5% | 9.10 |
| 8 | 0.2 | 31.5% | 29.9% | 4.57 |
| | 0.4 | 38.9% | 35.5% | 5.38 |
| | 0.6 | 60.4% | 25.6% | 7.34 |
| | 0.8 | 45.8% | 13.7% | 10.20 |
| 10 | 0.2 | 39.6% | 38.9% | 4.34 |
| | 0.4 | 50.5% | 45.5% | 5.04 |
| | 0.6 | 72.6% | 29.6% | 6.40 |
| | 0.8 | 68.2% | 16.2% | 11.02 |

Results for different specifications of the hierarchical system.

| p | Successes | In 5 Steps | Average number of Steps |
|-----|-----------|------------|-------------------------|
| 4 | 47.4% | 11.5% | 8.03 |
| 6 | 47.7% | 18.4% | 5.69 |
| 8 | 48.9% | 24.6% | 5.02 |
| 10 | 52.3% | 38.0% | 4.10 |

Evaluation of GSI (5/5)

Proportions of cases that the ranking of the recommended product was in the specified ranges for the random system.

| p | α | ranking ranges | | | | | | | |
|-----|----------|----------------|----------|----------|----------|-----------|-----------|-----------|------------|
| | | 1 | ≤ 2 | ≤ 3 | ≤ 5 | ≤ 10 | ≤ 25 | ≤ 50 | ≤ 100 |
| 4 | 0.2 | 16.8% | 24.6% | 29.9% | 36.5% | 47.0% | 66.7% | 84.4% | 95.0% |
| | 0.4 | 29.3% | 36.5% | 42.4% | 49.5% | 60.1% | 78.5% | 91.0% | 99.4% |
| | 0.6 | 41.7% | 50.5% | 58.9% | 64.2% | 75.7% | 87.9% | 94.4% | 100.0% |
| | 0.8 | 56.1% | 67.3% | 73.2% | 80.4% | 89.4% | 96.0% | 98.4% | 99.7% |
| 6 | 0.2 | 29.3% | 40.8% | 45.2% | 49.2% | 60.4% | 75.1% | 86.6% | 96.0% |
| | 0.4 | 42.7% | 50.5% | 57.9% | 64.5% | 74.1% | 88.5% | 96.0% | 97.8% |
| | 0.6 | 52.7% | 64.8% | 69.2% | 75.7% | 84.1% | 91.9% | 96.9% | 97.8% |
| | 0.8 | 70.1% | 80.1% | 82.2% | 86.3% | 94.1% | 97.8% | 99.7% | 100.0% |
| 8 | 0.2 | 43.0% | 54.8% | 59.5% | 64.5% | 75.4% | 89.1% | 95.3% | 98.8% |
| | 0.4 | 47.0% | 58.6% | 63.9% | 71.3% | 77.3% | 90.0% | 94.7% | 98.1% |
| | 0.6 | 66.4% | 76.0% | 79.4% | 84.7% | 91.0% | 96.6% | 97.5% | 98.8% |
| | 0.8 | 80.1% | 87.2% | 91.0% | 93.2% | 97.5% | 99.4% | 99.4% | 99.4% |
| 10 | 0.2 | 46.4% | 56.1% | 60.8% | 67.0% | 79.8% | 92.5% | 96.6% | 99.1% |
| | 0.4 | 58.9% | 70.4% | 74.8% | 80.4% | 91.0% | 95.3% | 98.8% | 100% |
| | 0.6 | 70.4% | 76.0% | 81.6% | 87.5% | 93.2% | 97.2% | 98.4% | 98.8% |
| | 0.8 | 83.8% | 89.7% | 92.2% | 93.8% | 96.0% | 98.8% | 99.1% | 99.1% |

Proportions of cases that the ranking of the recommended product in the specified ranges for the hierarchical system.

| p | ranking ranges | | | | | | | |
|-----|----------------|----------|----------|----------|-----------|-----------|-----------|------------|
| | 1 | ≤ 2 | ≤ 3 | ≤ 5 | ≤ 10 | ≤ 25 | ≤ 50 | ≤ 100 |
| 4 | 47.4% | 56.4% | 64.2% | 71.3% | 77.6% | 85.4% | 89.7% | 95.6% |
| 6 | 47.7% | 55.1% | 62.3% | 70.1% | 76.0% | 85.4% | 90.3% | 95.6% |
| 8 | 48.9% | 56.4% | 62.3% | 70.4% | 77.3% | 86.9% | 94.1% | 97.5% |
| 10 | 52.3% | 58.6% | 67.9% | 74.8% | 81.6% | 90.0% | 93.8% | 98.8% |

Outline

- Problem definition
- Recommender Systems
- Methodology
- Graphical Recommender System
- Graphical Shopping Interface
- Evaluation of the Graphical Shopping Interface
- **Conclusions**

Conclusions (1/3)

- New way to display similarities in a 2D space
 - Two prototypes
 - GRS with user explicit input
 - GSI that can be used to navigate through products
- GRS
 - Representations acceptable up to 2D spaces with 10 products
 - Quality decreases if p increases

Conclusions (2/3)

- GSI
 - Results of the clustering method is not good enough to be applied in practice
 - Random system with high alpha value should be preferred
 - Hierarchical system should be preferred if we need few steps

Conclusions (3/3)

- Improve the system
 - Extend the domain model with other sub models
 - Allow the customer to select any point in the space
 - Computing weights costs time, instead try to learn ideal weights for a population