Multi-model fitting as clustering in Preference Space

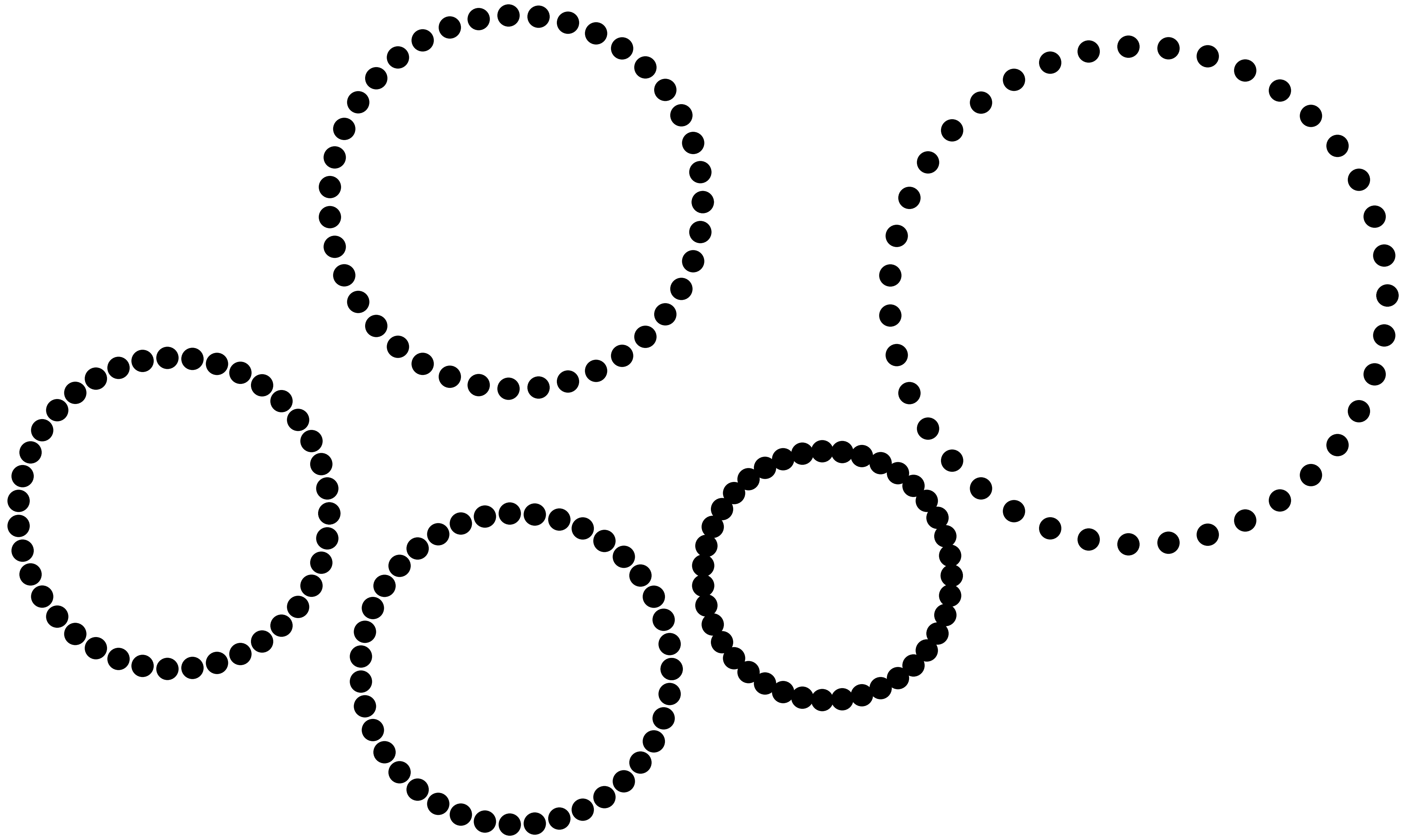
Luca Magri

DEIB - 9 DECEMBER 2019

POLITECNICO
MILANO 1863

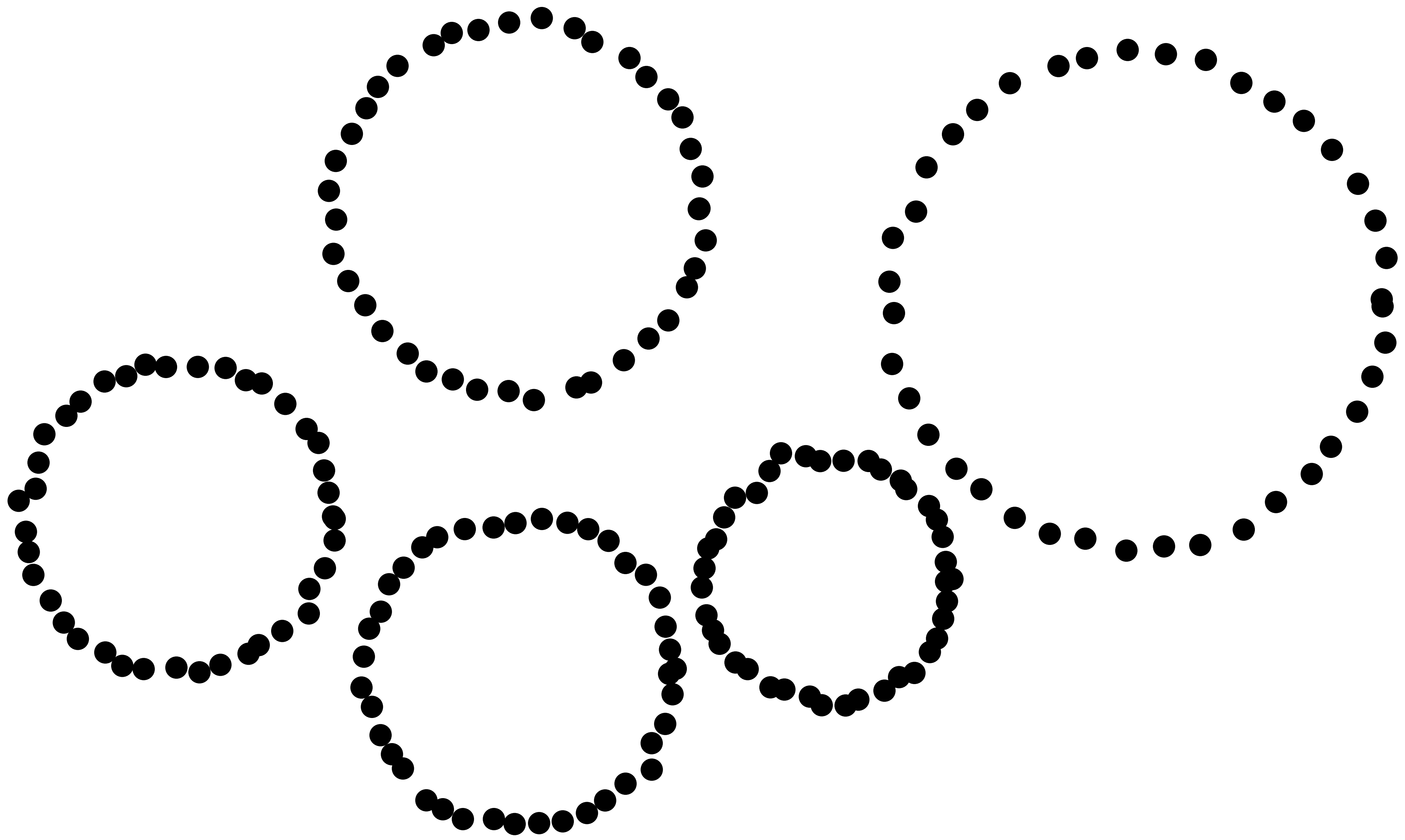
The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$
possibly corrupted by noise and outlier,
and a family of geometric models Θ,



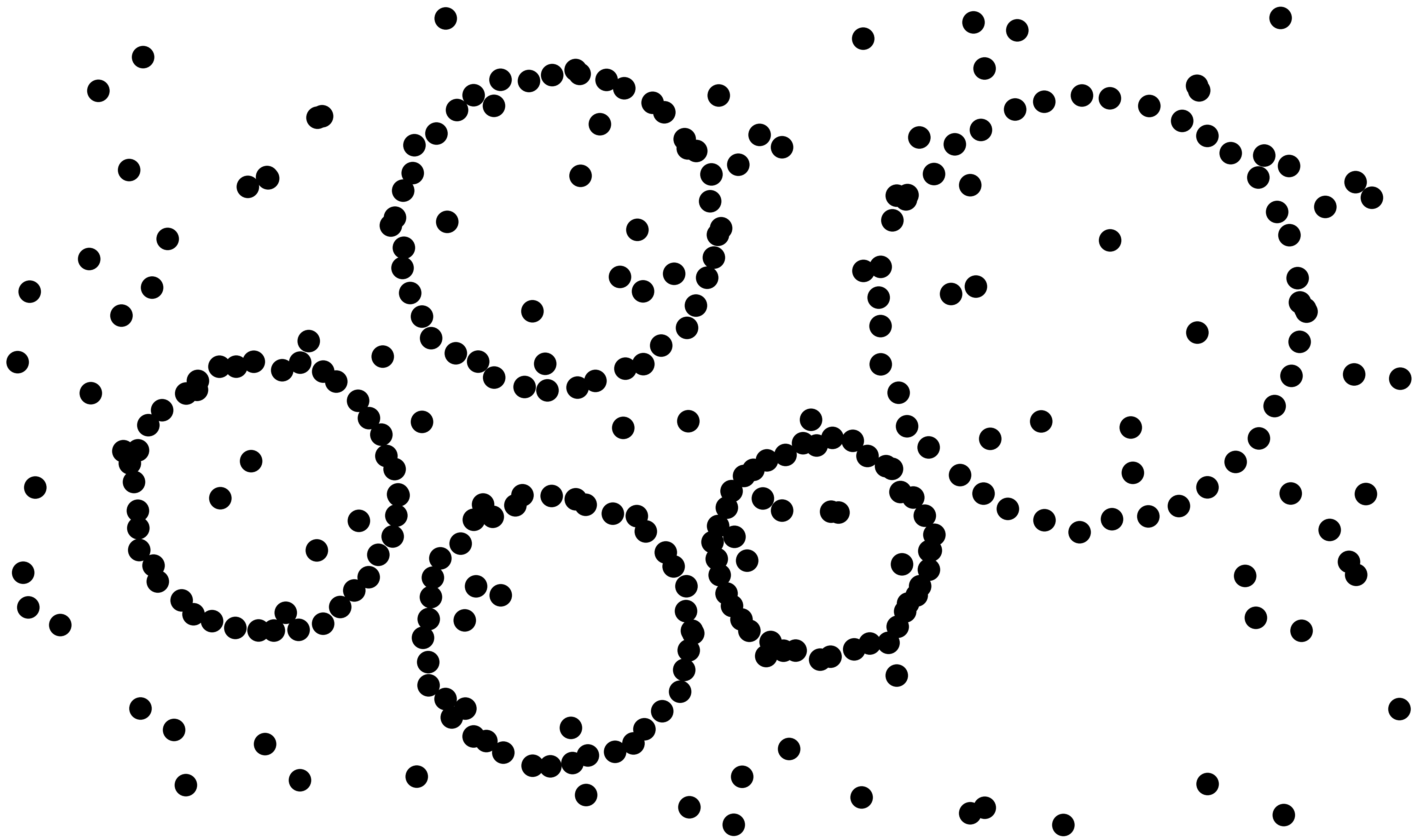
The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
possibly corrupted by noise and outlier,
and a family of geometric models Θ,



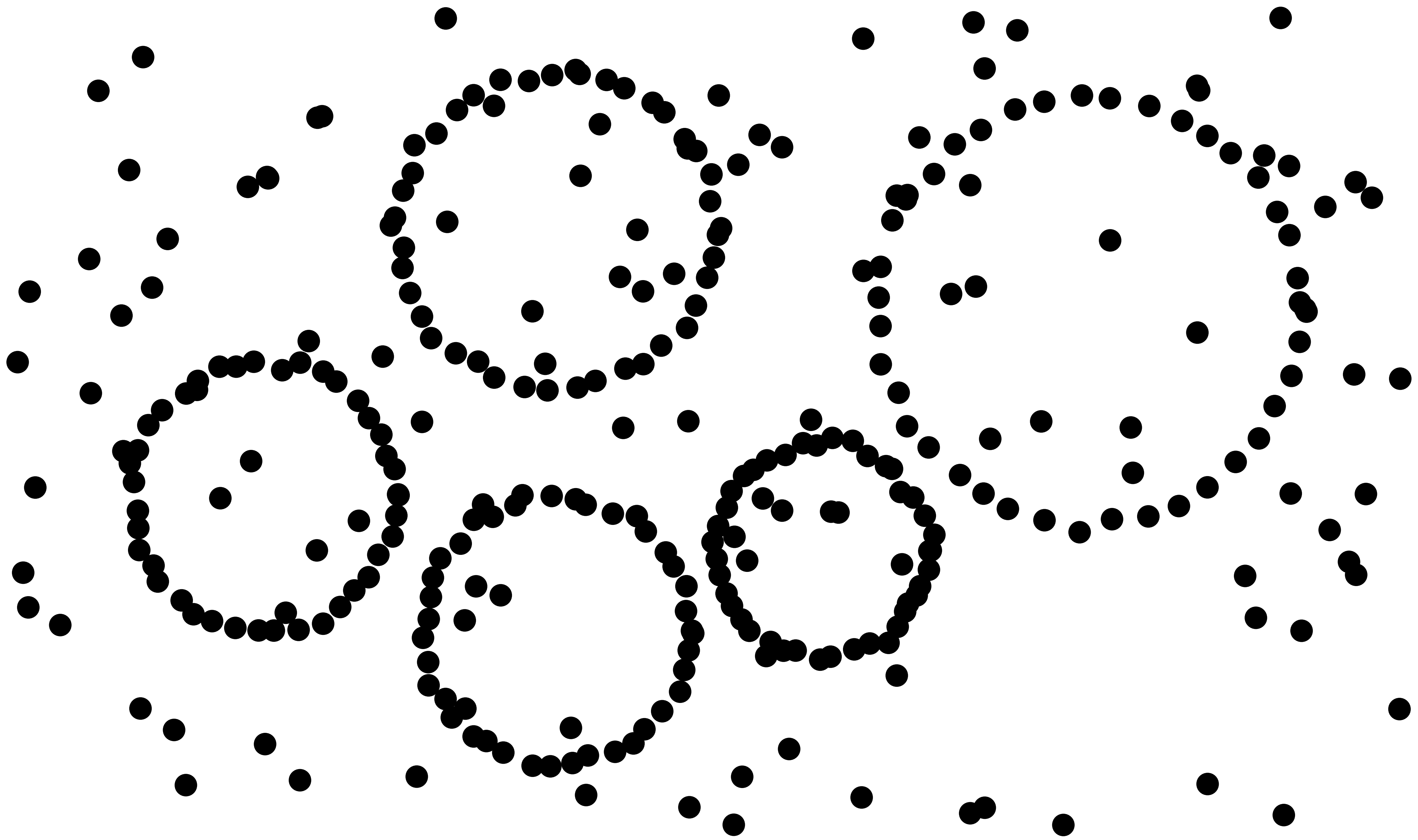
The problem of multi-model fitting (or structure recovery)

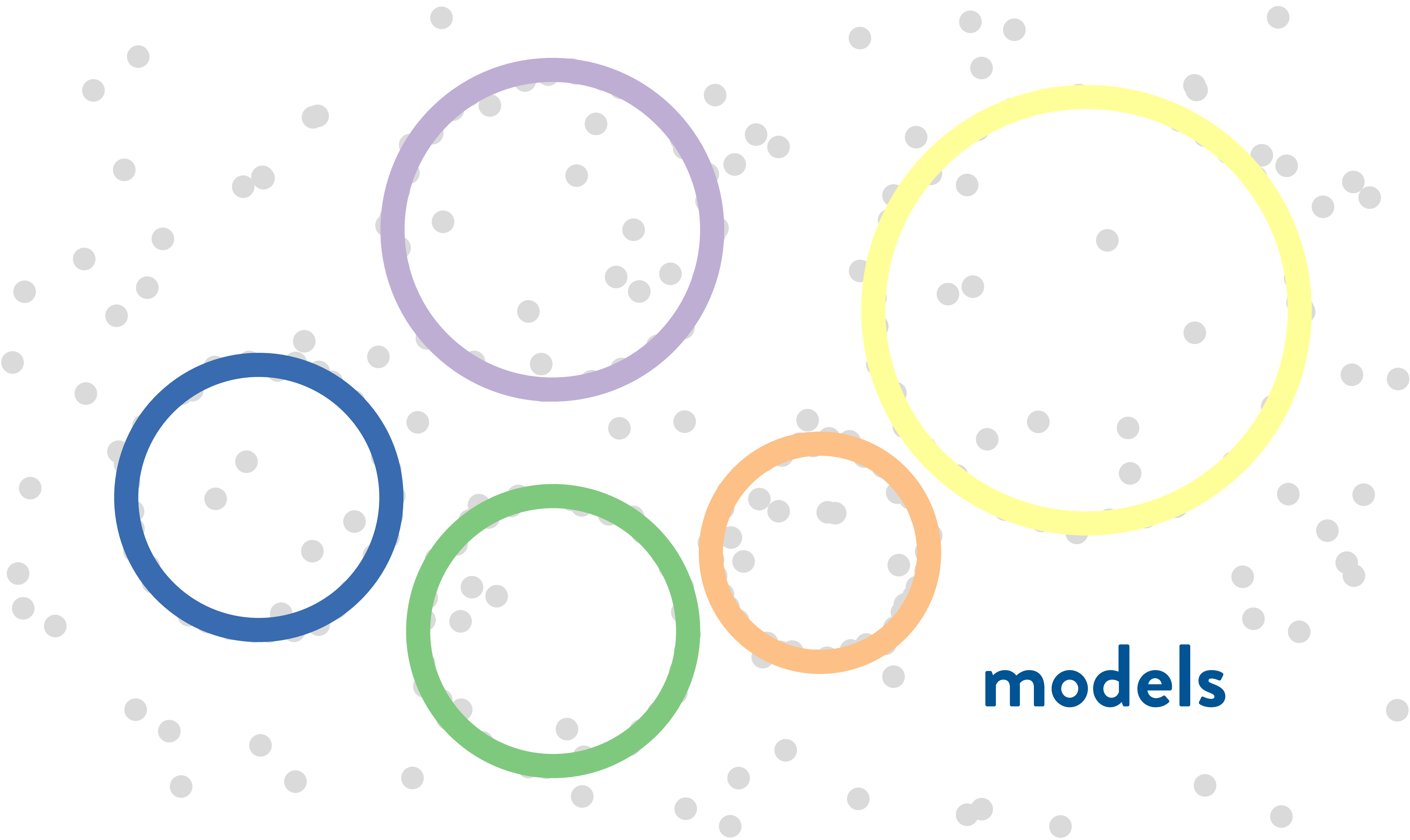
Given a set of data $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
possibly corrupted by noise and outlier,
and a family of geometric models Θ,

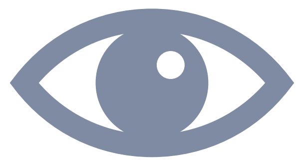


The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ possibly corrupted by noise and outlier, and a family of geometric models Θ, automatically estimate the models that best explain the data/discover the structures hidden in the data



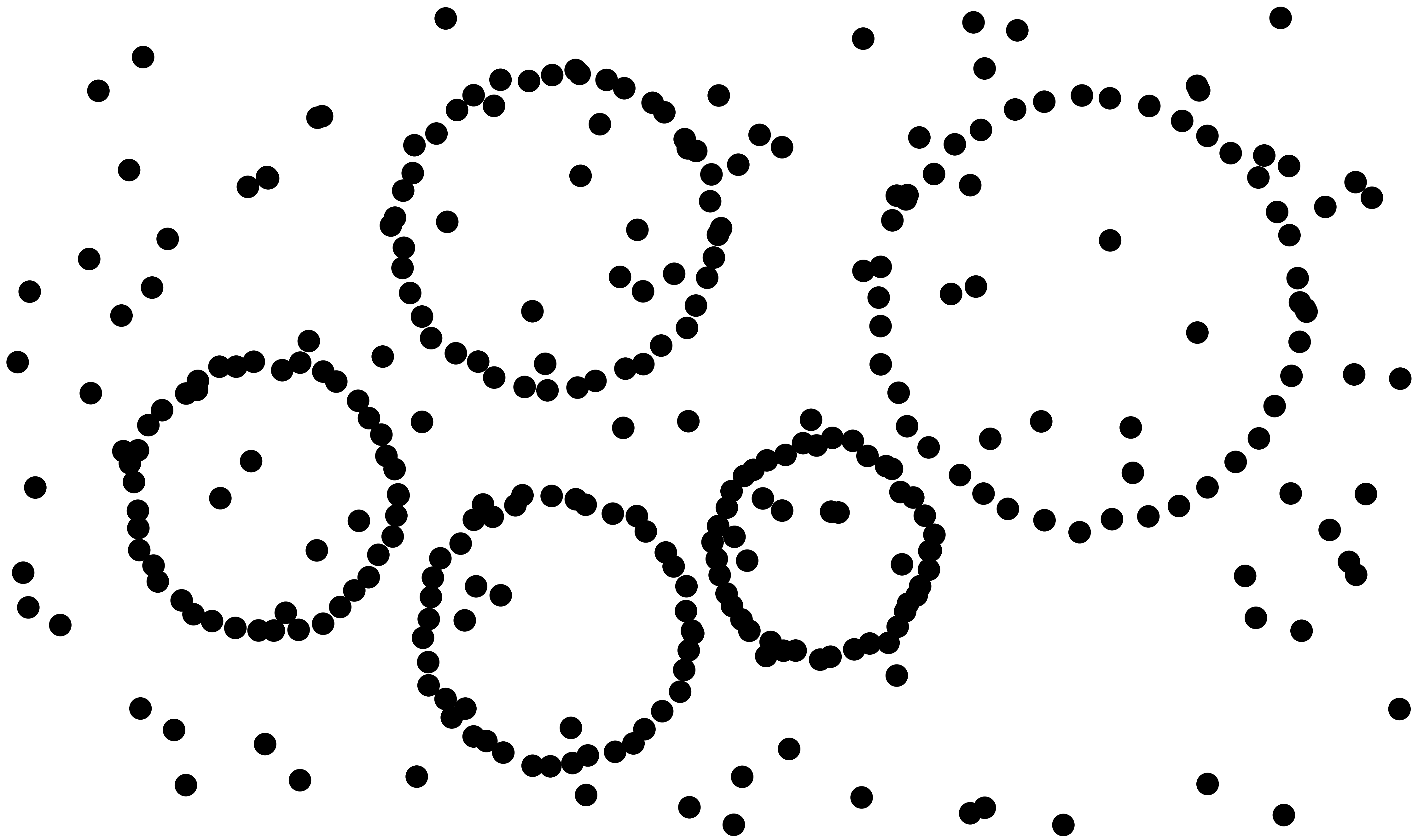


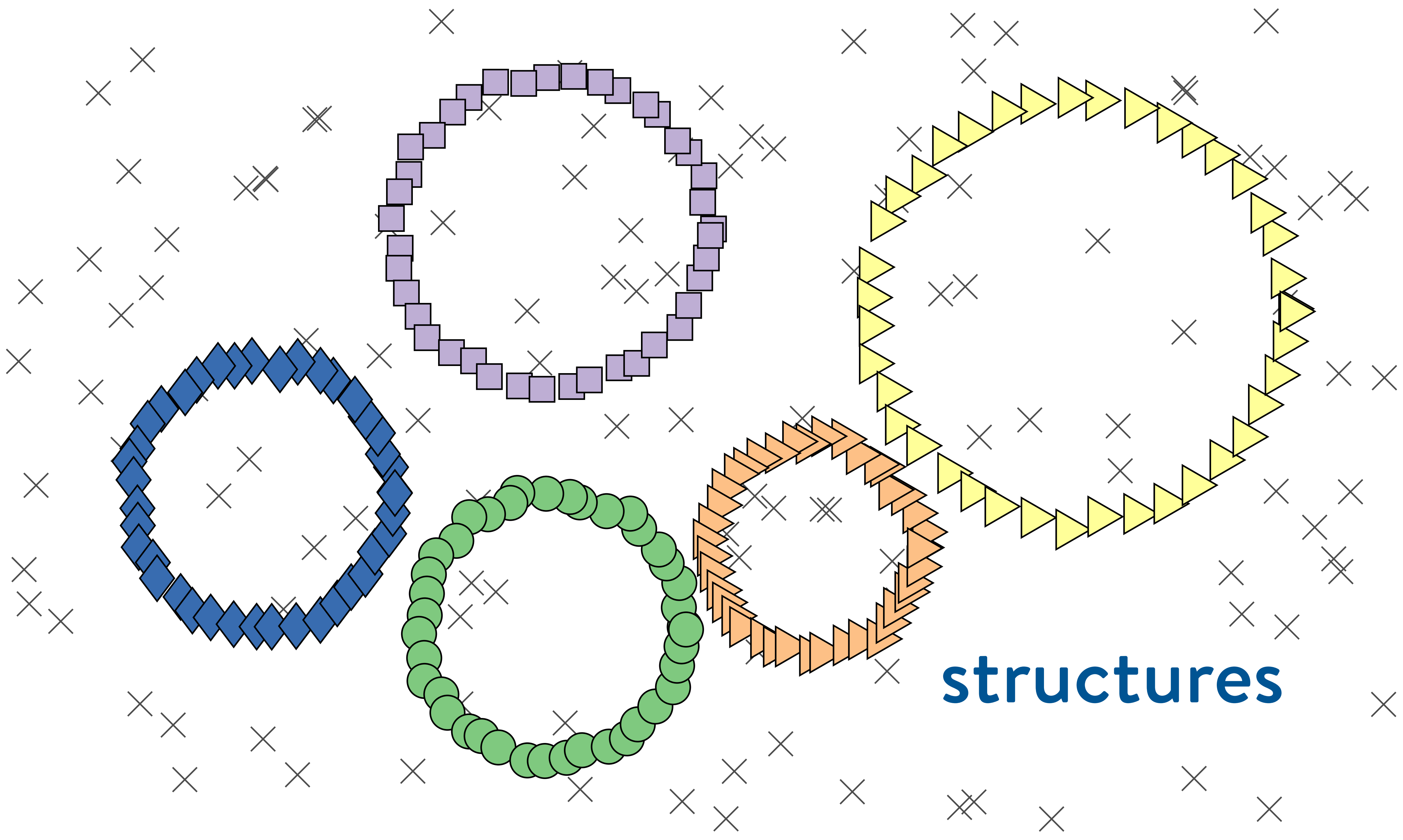
 "in the eye of the beholder", mathematical descriptions of the data that an observer fits

The problem of multi-model fitting (or structure recovery)

Given a set of data $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ possibly corrupted by noise and outlier, and a family of geometric models Θ,

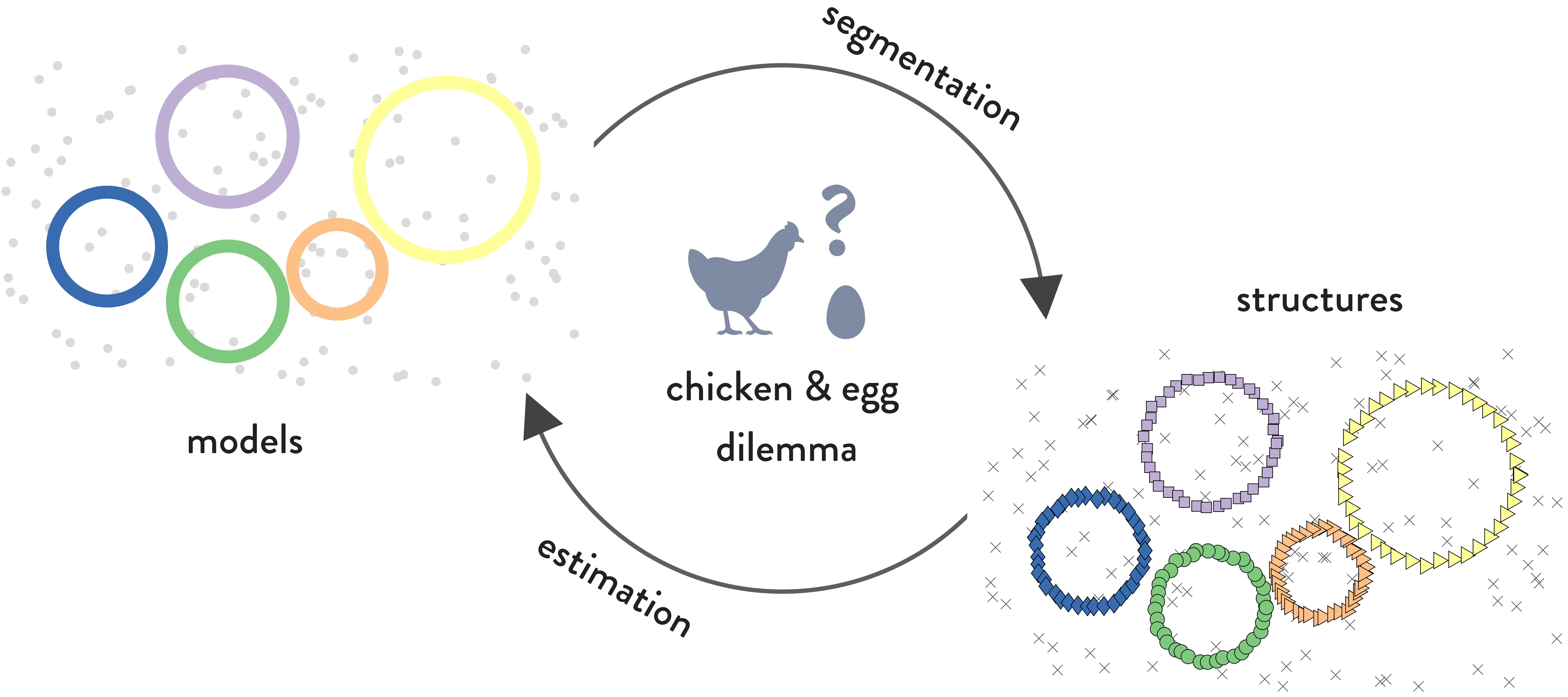
automatically estimate the models that best explain the data/discover the structures hidden in the data





relations among the data, intrinsic to data

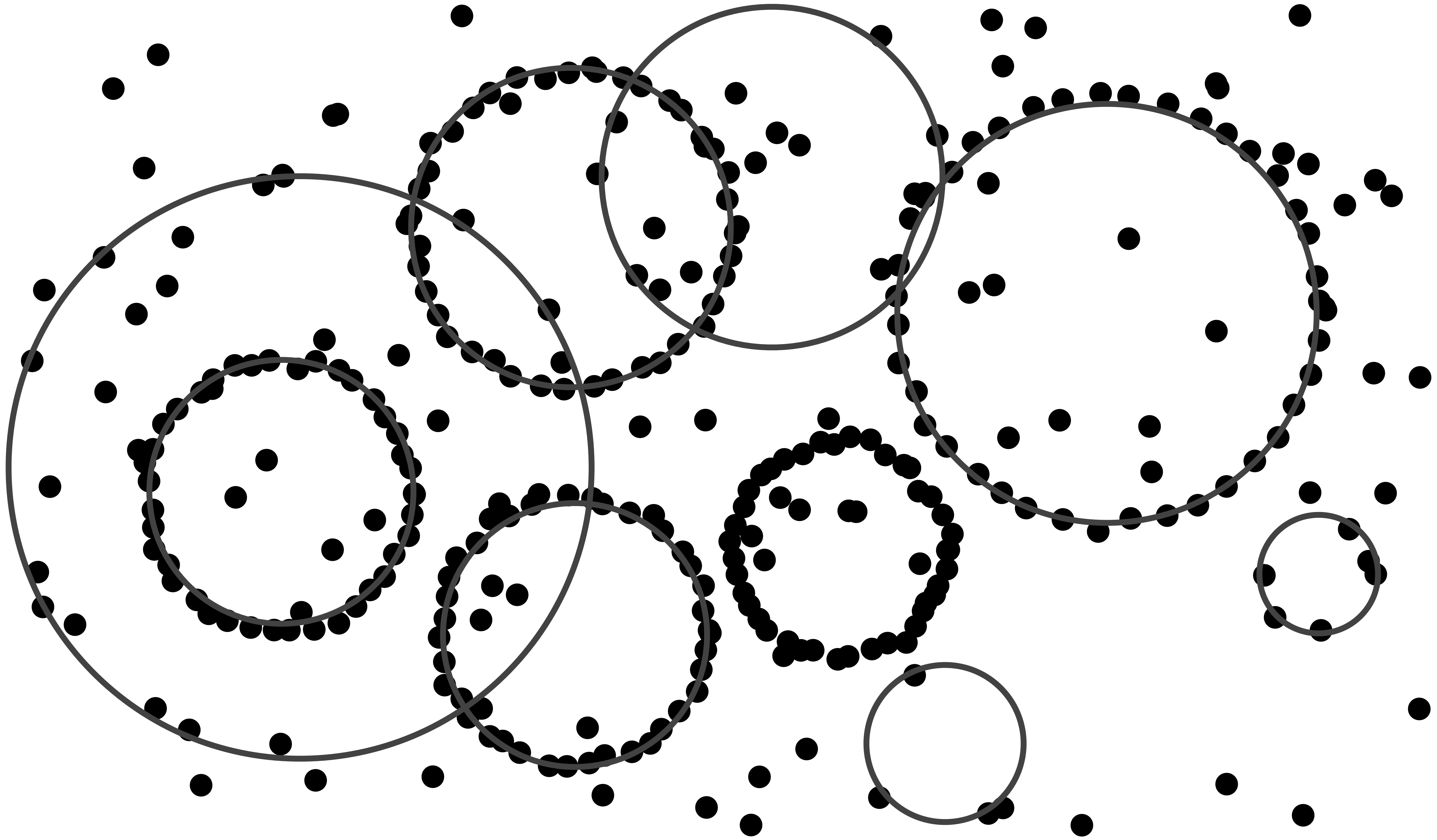
The challenges of multi model fitting



The challenges of multi model fitting

Number of models?

Inliers/outliers?

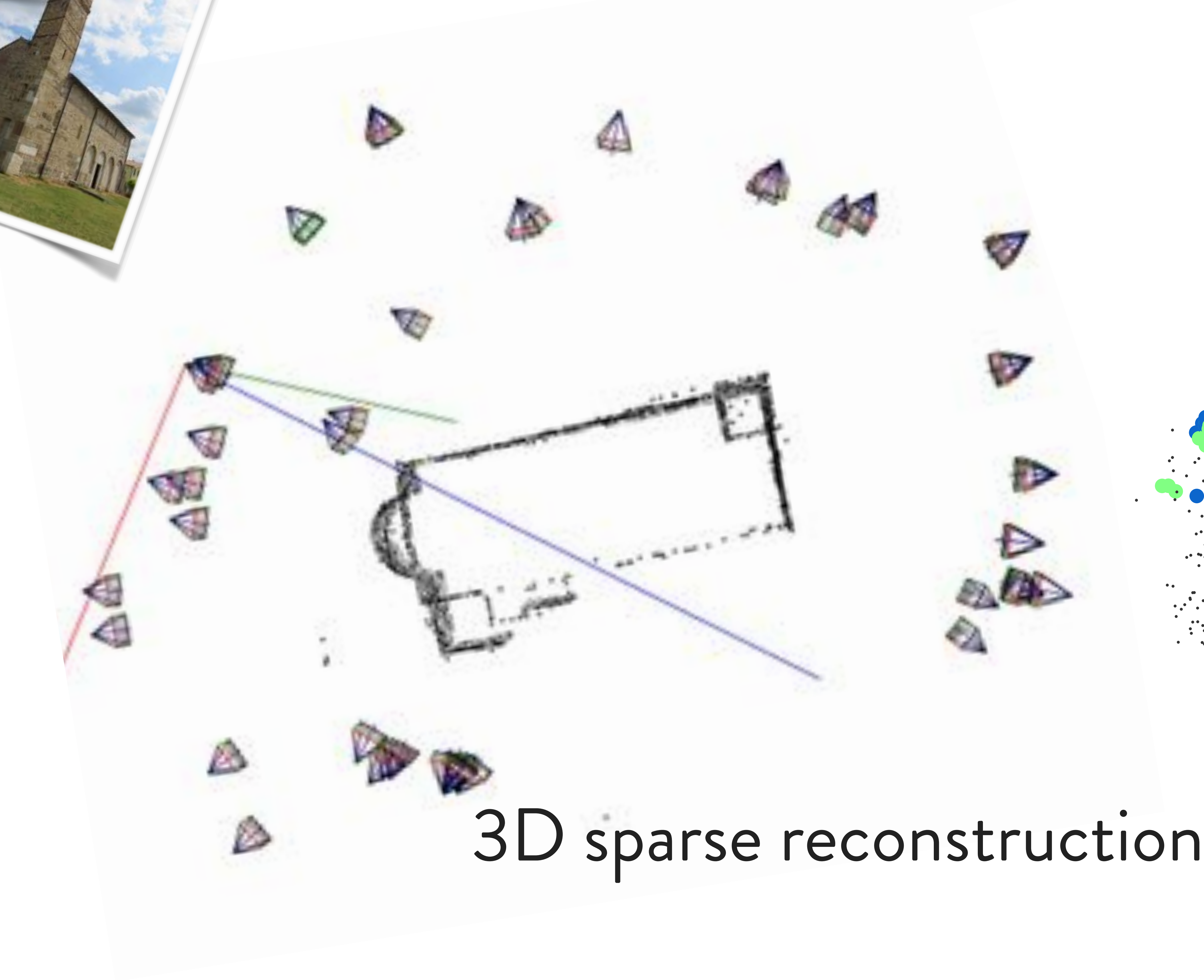


ill posed

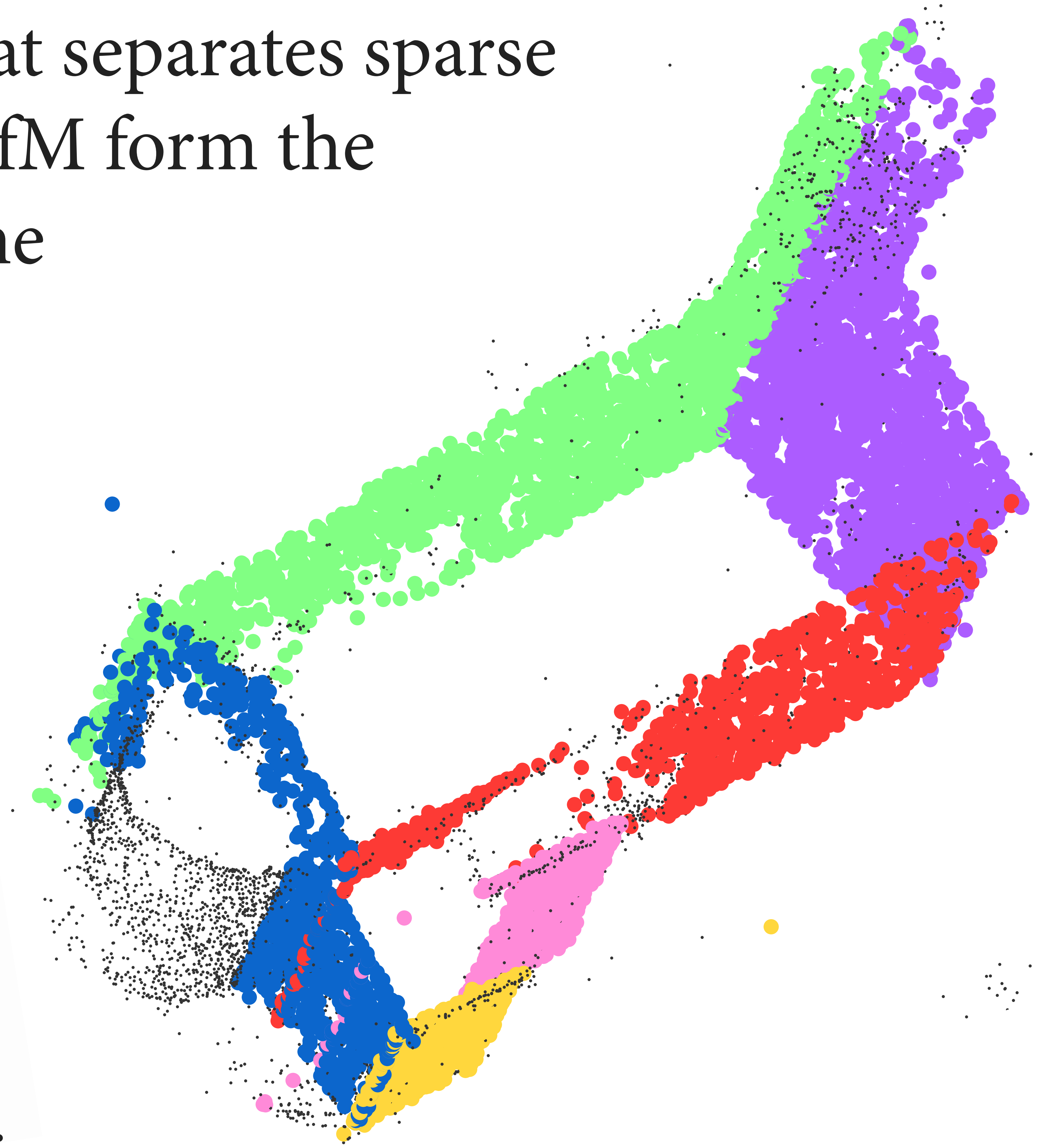
Multi model fitting applications: primitive fitting

Bridge the semantic gap that separates sparse point cloud coming from SfM form the understanding of a 3D scene

input images



3D sparse reconstruction

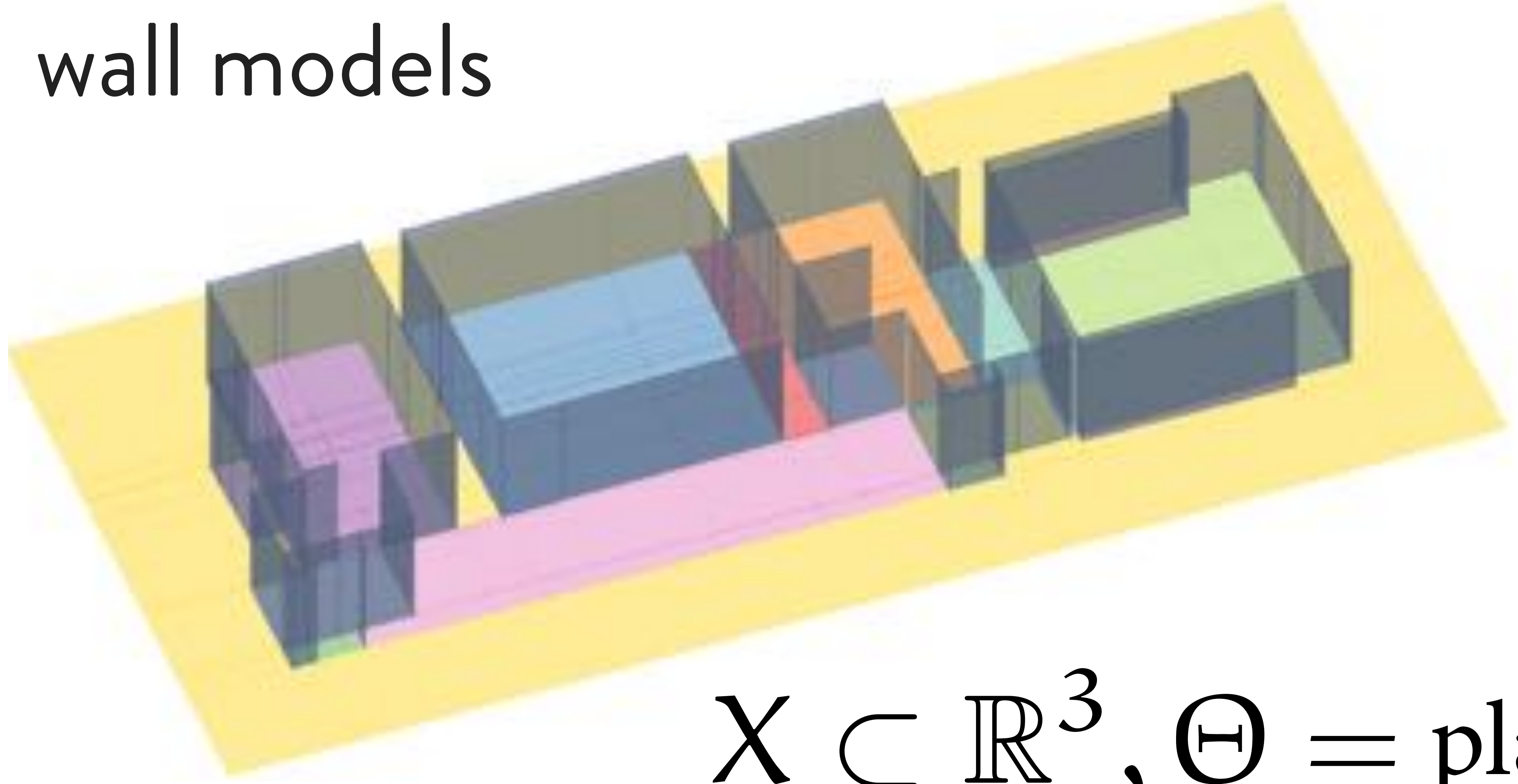


$X \subset \mathbb{R}^3, \Theta = \text{planes}$

Multi model fitting applications: scan2bim

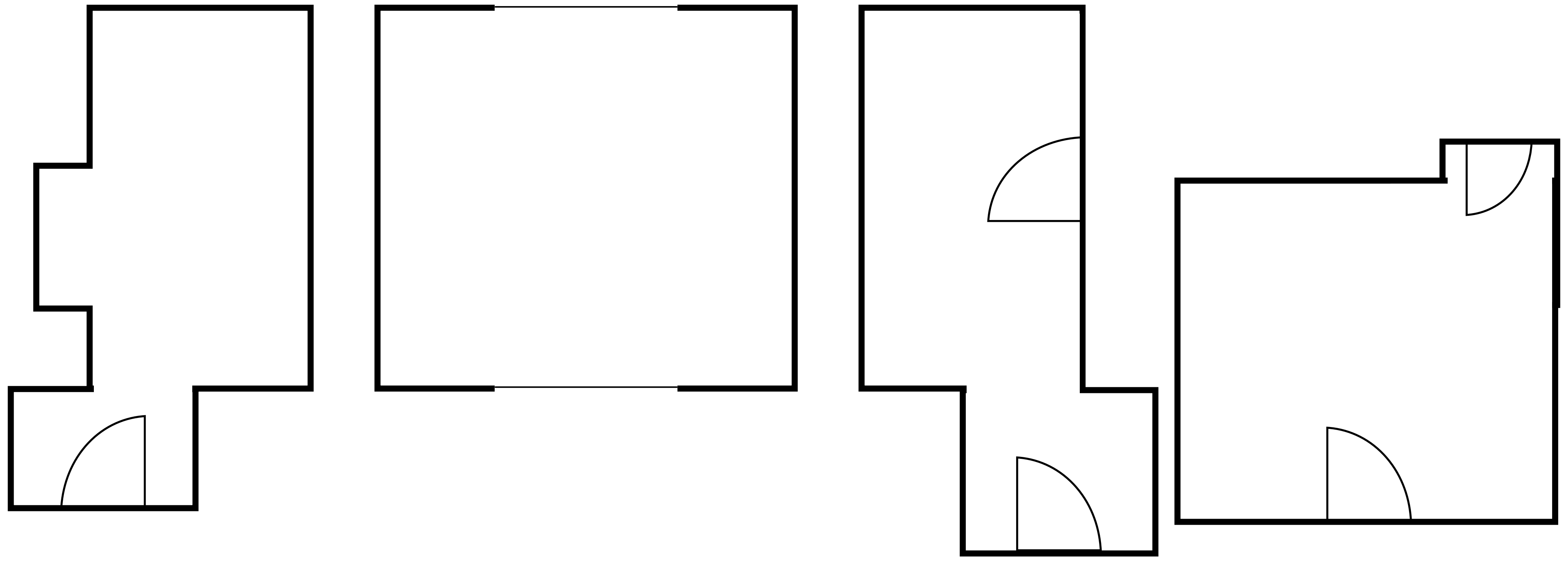
scanned point cloud

wall models



$X \subset \mathbb{R}^3, \Theta = \text{planes}$

floor-plan



$X \subset \mathbb{R}^2, \Theta = \text{lines}$

Given a scanned point cloud of an interior environment, detect its primary facility surfaces – such as floors, walls, and ceilings.

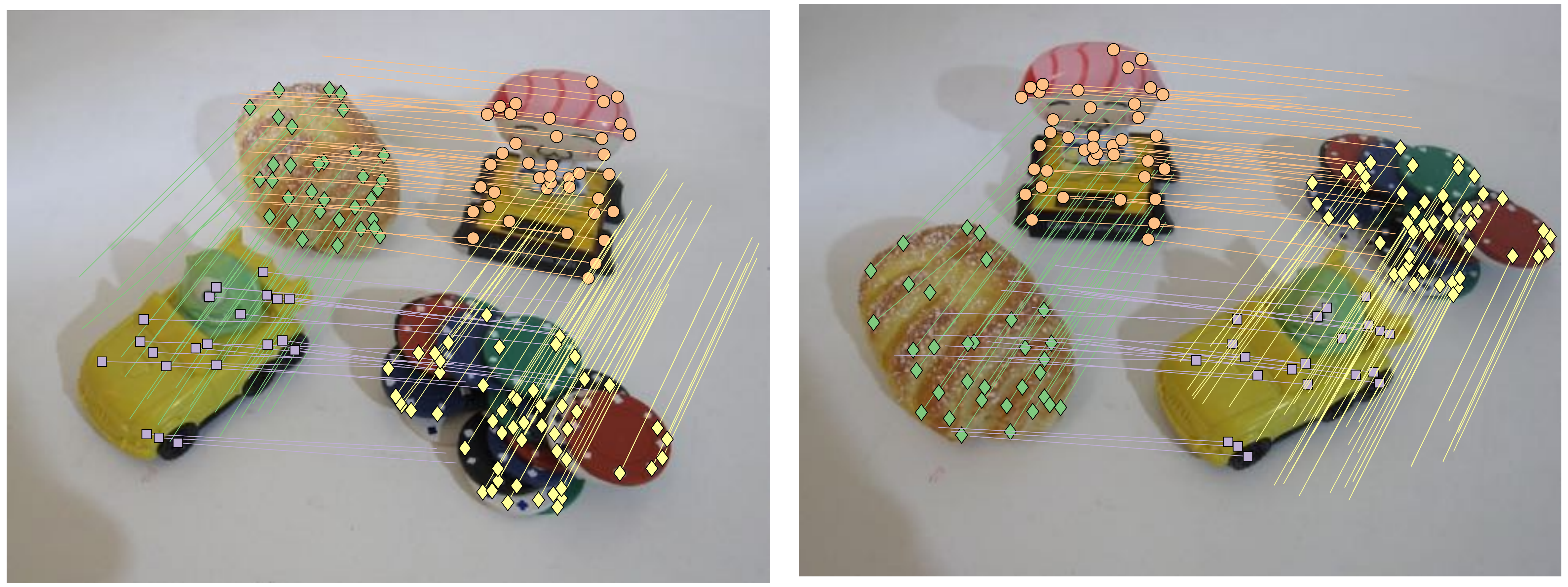
Multi model fitting applications: two view geometry

Geometric fit on corresponding matches across two images

plane detection

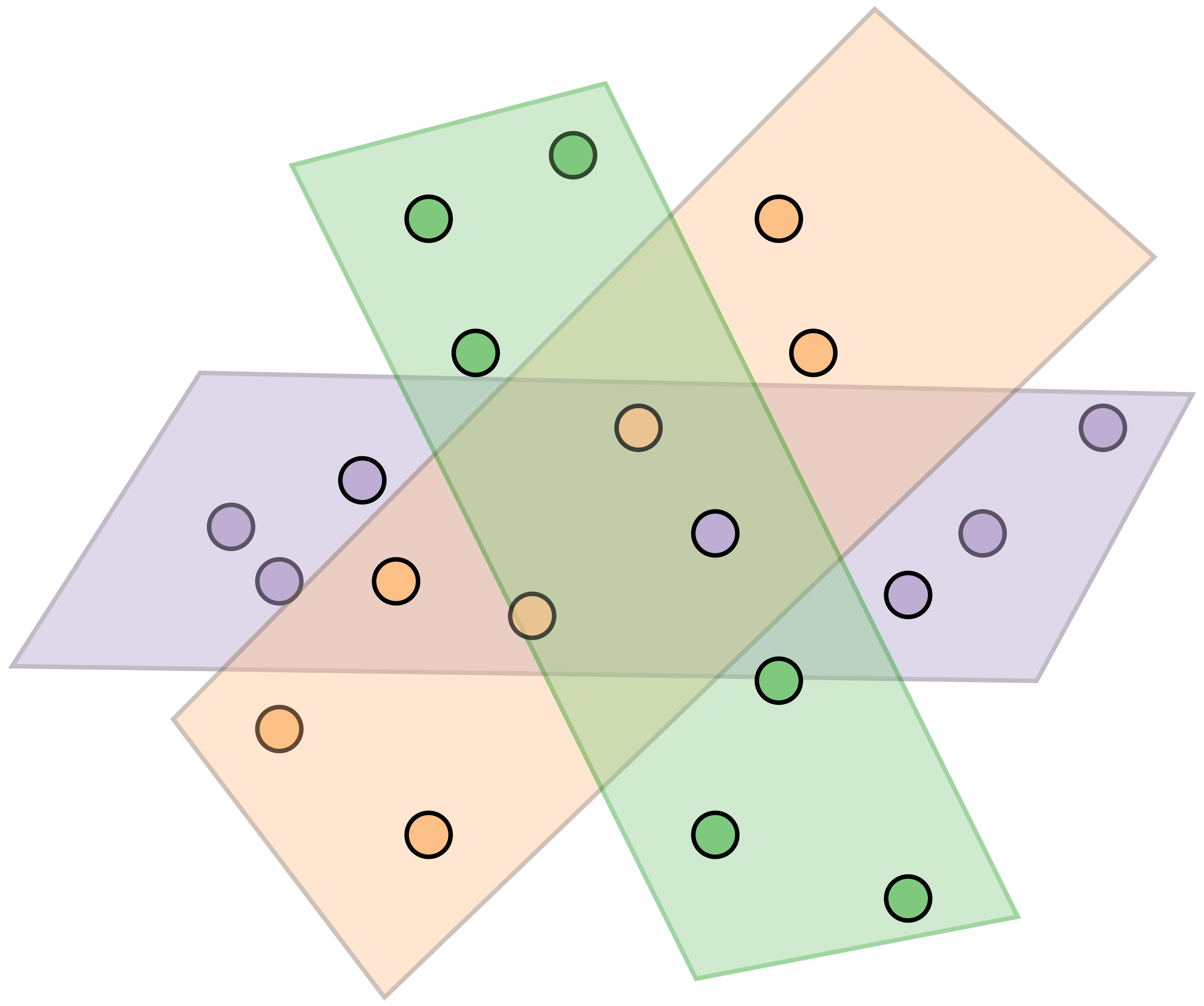
$X \subset \mathbb{R}^4, \Theta =$ homographies

epipolar geometry



$X \subset \mathbb{R}^4, \Theta =$ fundamental matrices

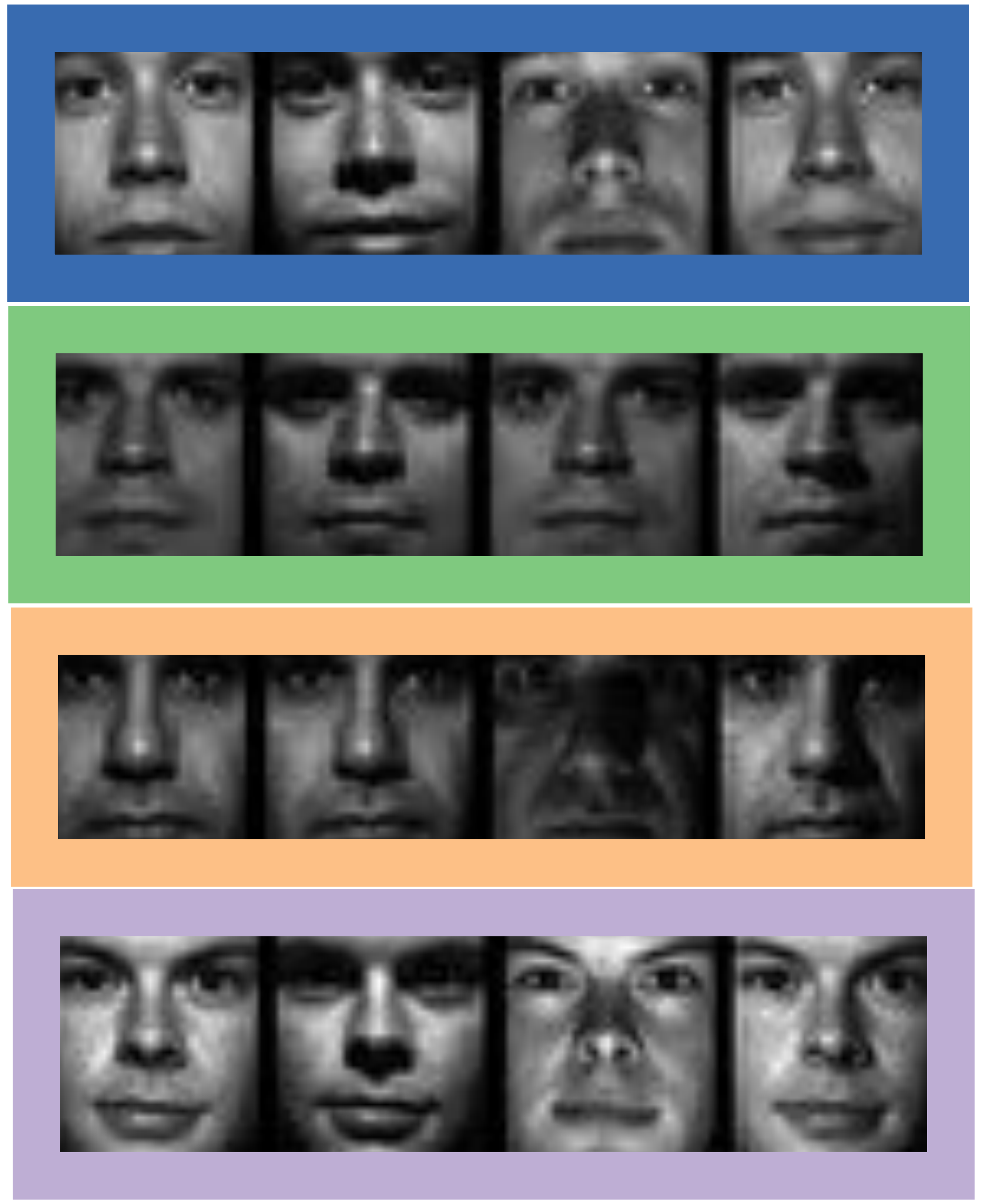
Multi model fitting applications: subspace clustering



$$X \subset \mathbb{R}^d, \Theta = \text{subspaces}$$

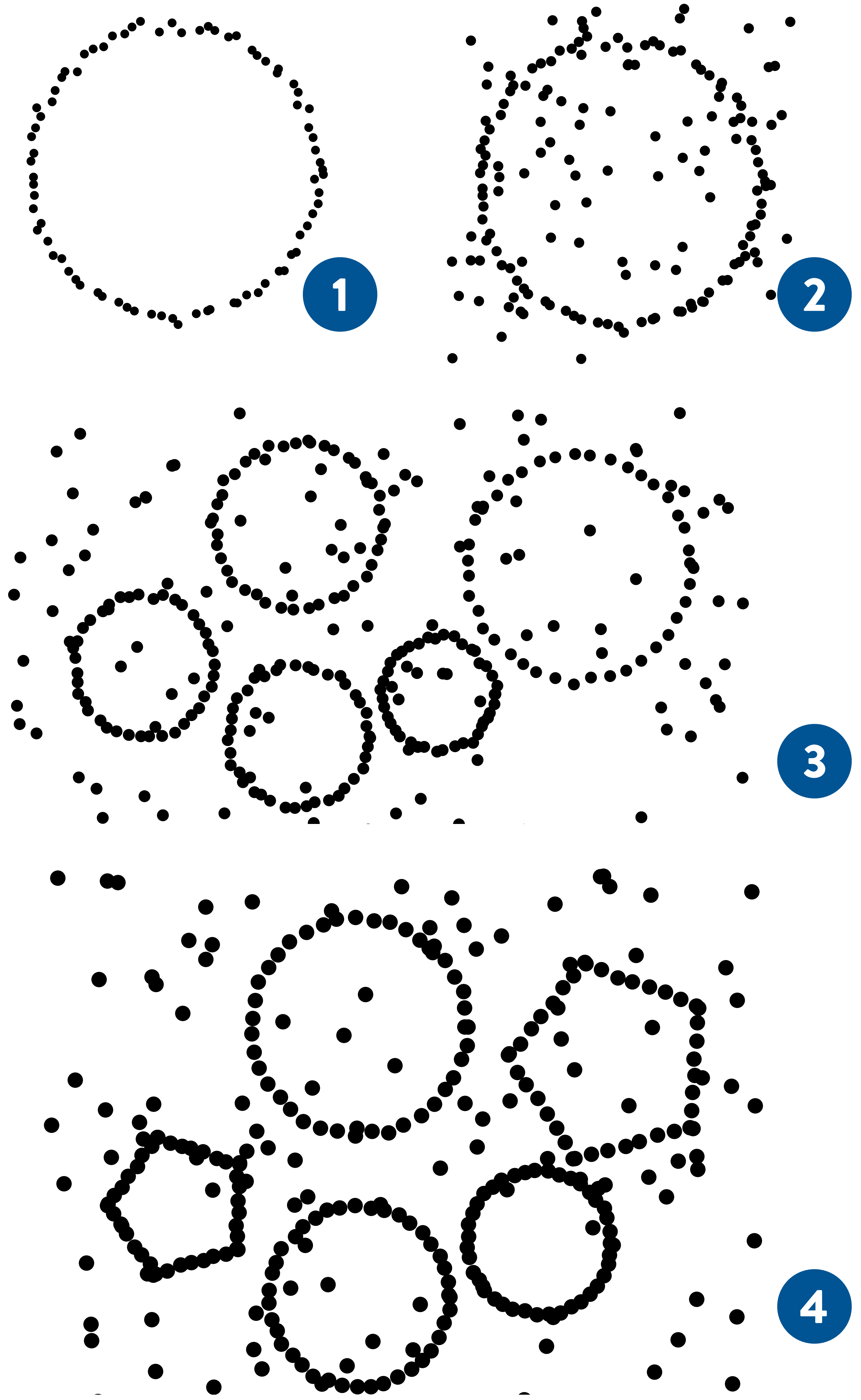
3D Video segmentation

Face clustering



Outline

1. Single model fitting
2. Robust single model fitting
3. Multi model fitting
4. Multi-class multi model fitting



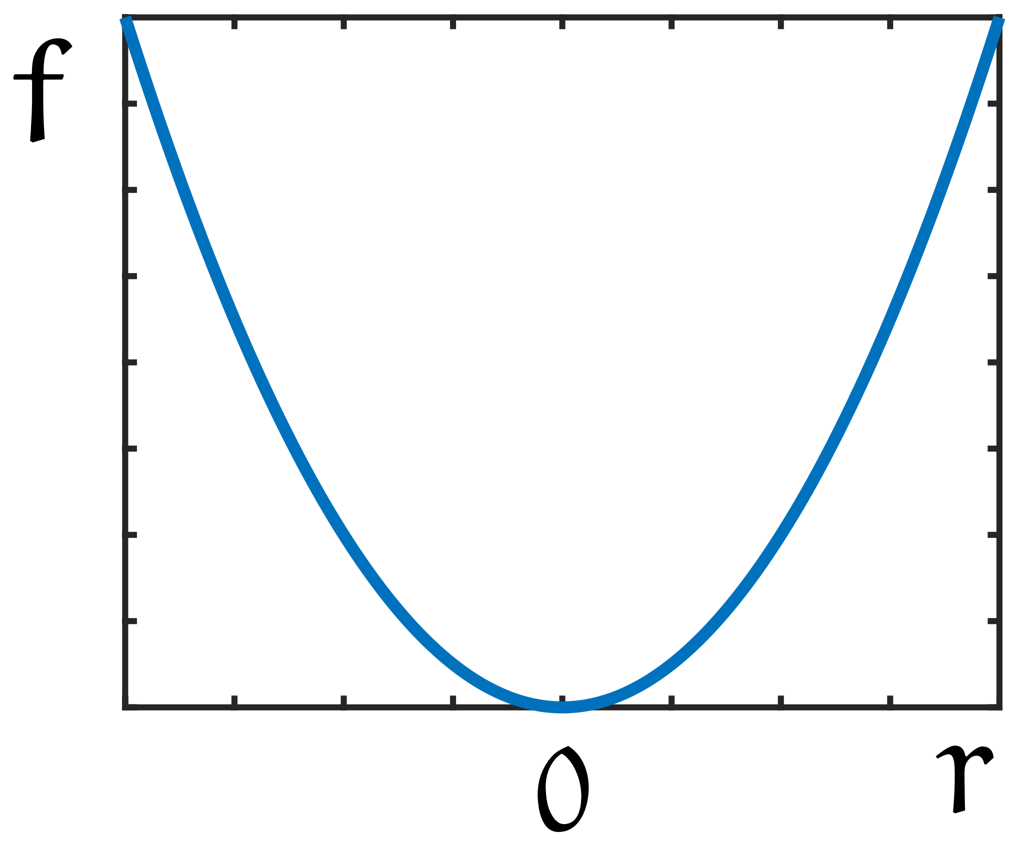
Single model fitting: least squares

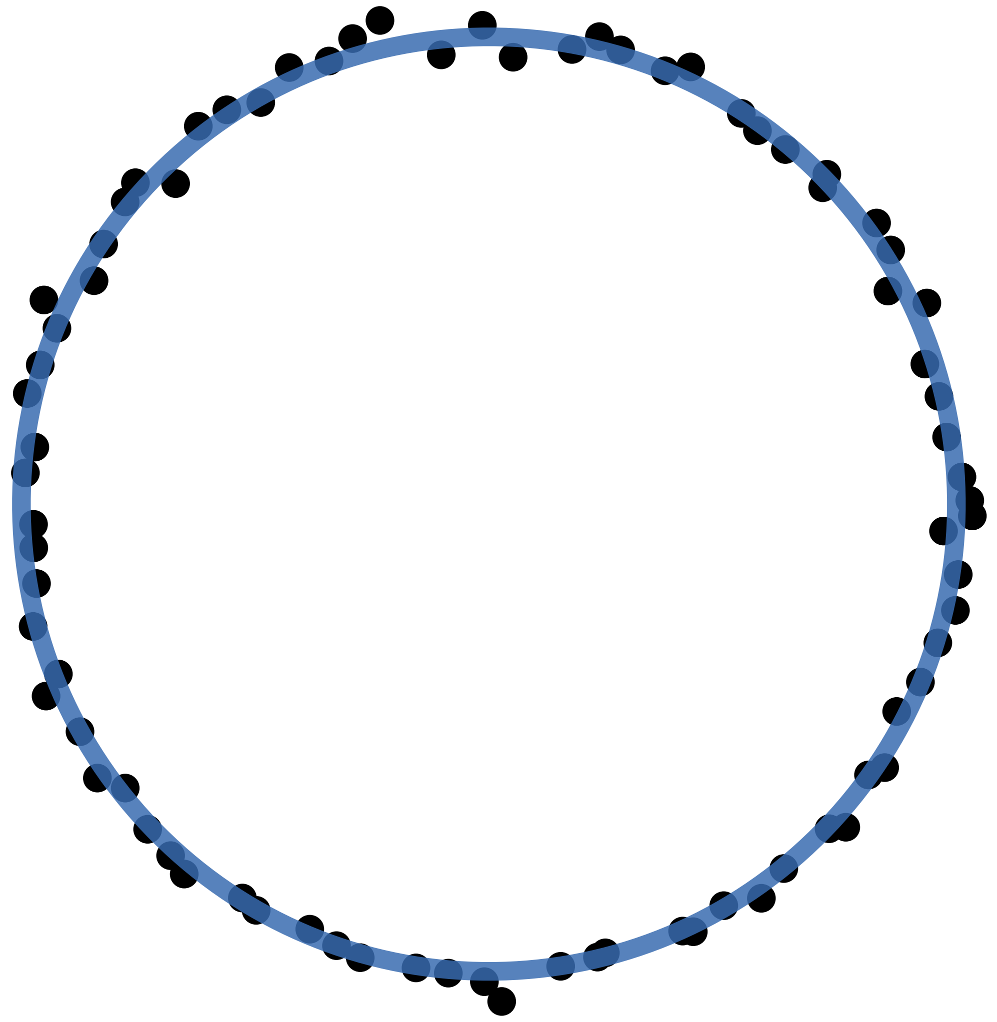
Find the circle that "best fit" the data points,

Least squares

$$\theta^* = \arg\min_{\theta} \sum_{x \in X} f(r)$$

$$f(r) = r^2$$



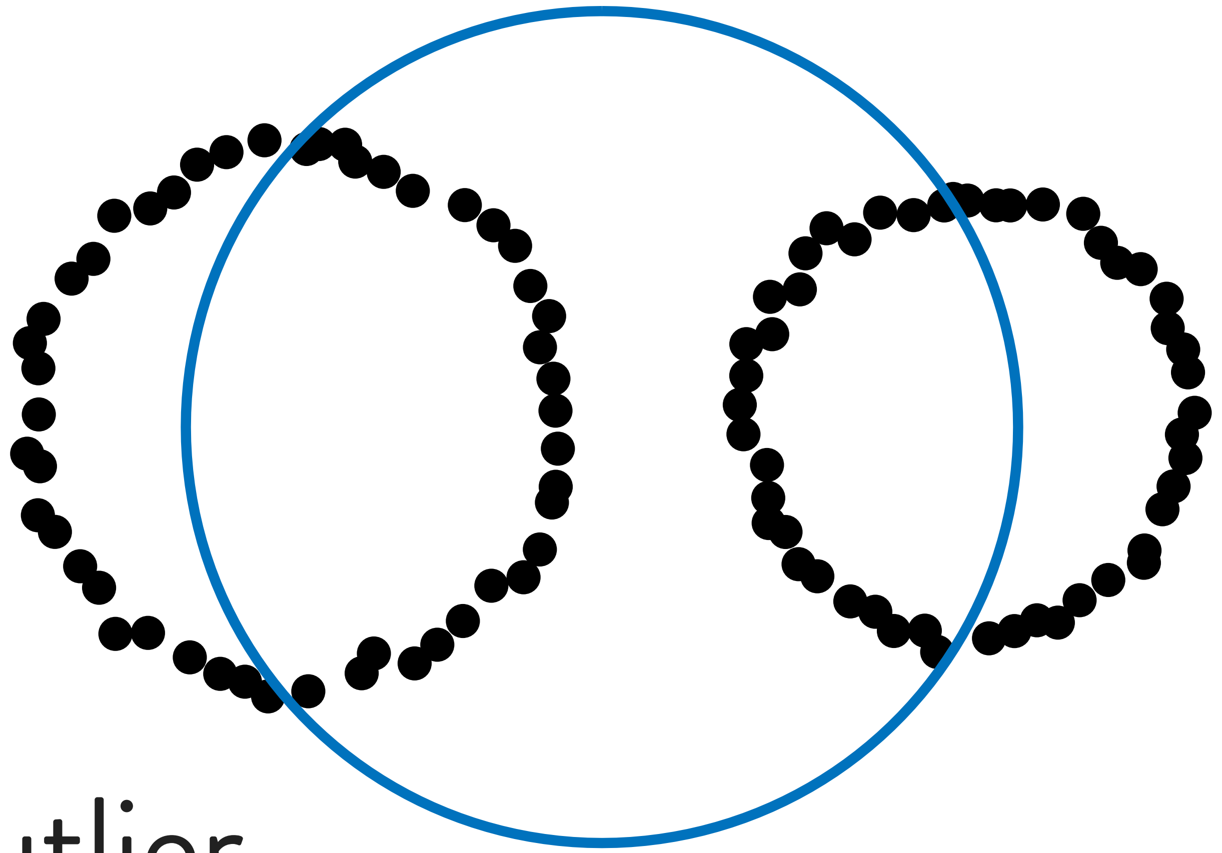


- data points $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^2$
- circle $x^2 + y^2 + ax + by + c = 0$
- models $\Theta = \{\theta = (a, b, c) \in \mathbb{R}^3\}$
- residuals $r = r(x, \theta) = \mathrm{dist}(x, \theta)$

The maximum likelihood method, in case of noise with normal distribution.
Algebraic residuals can be used, but geometric residuals provides better results.

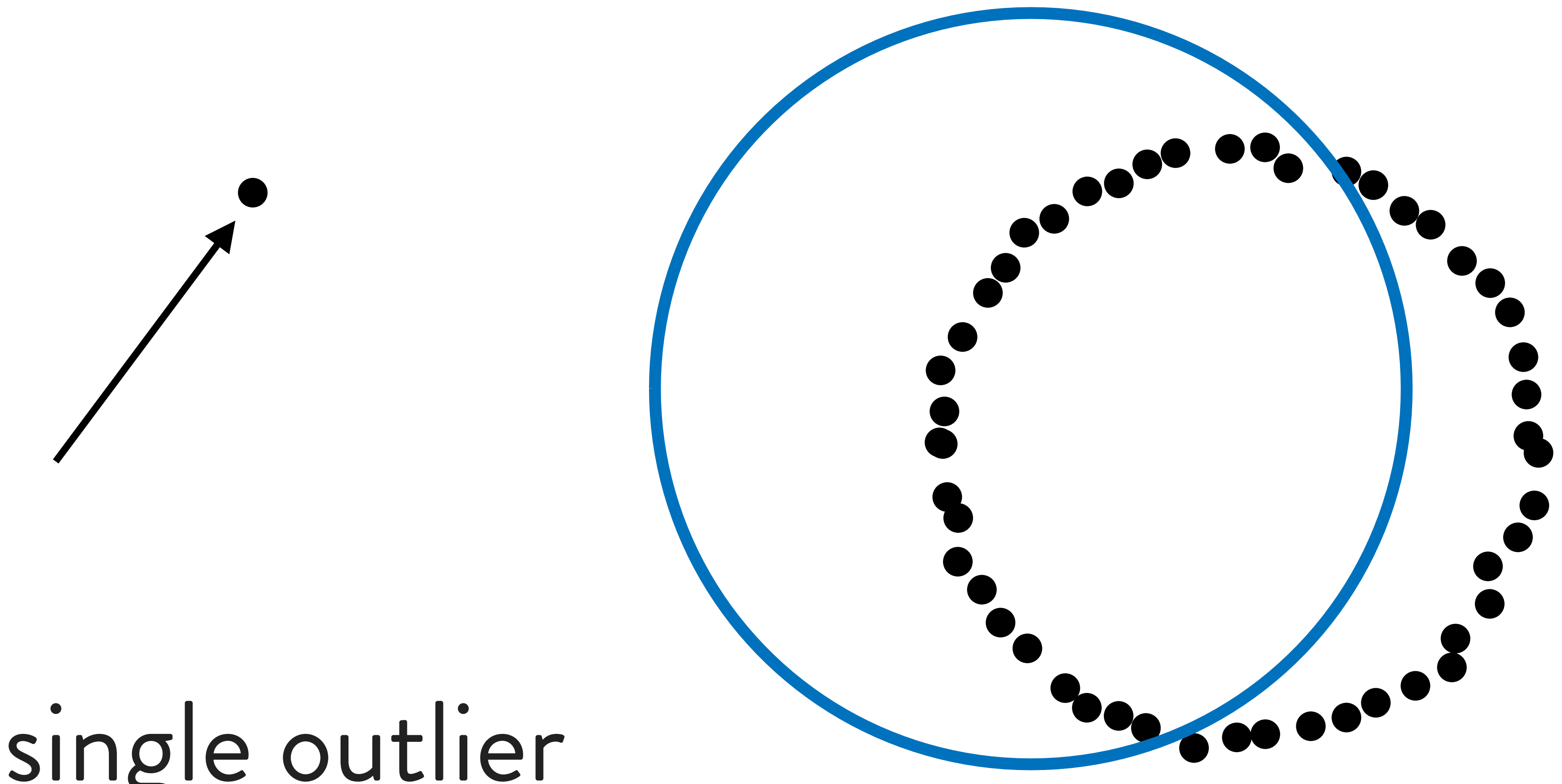
Single model fitting: least squares

Break down point = the proportion of incorrect observations that can be handled before giving an incorrect result.

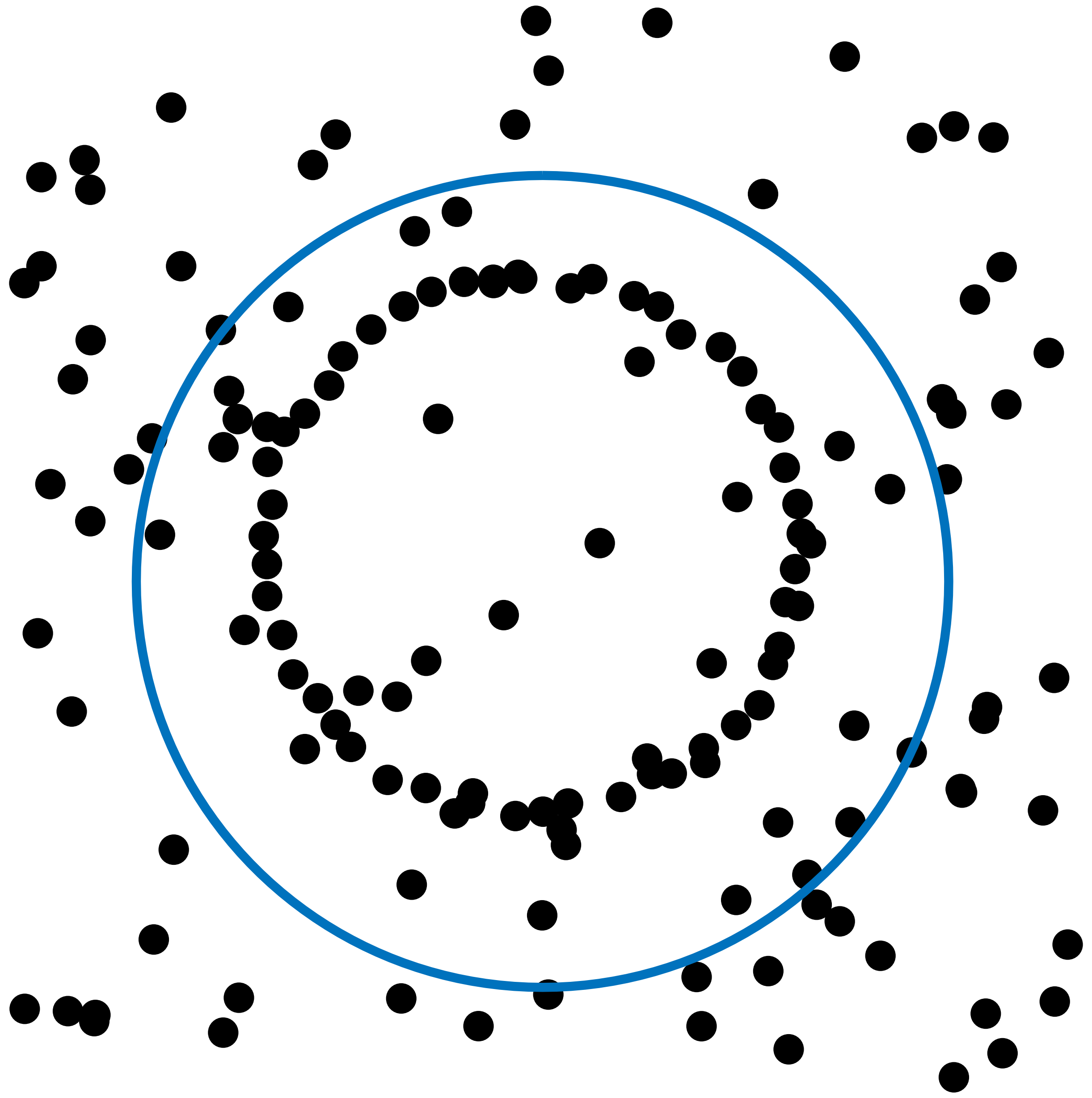


pseudo-outlier

Least squares has 0% breakdown point



single outlier



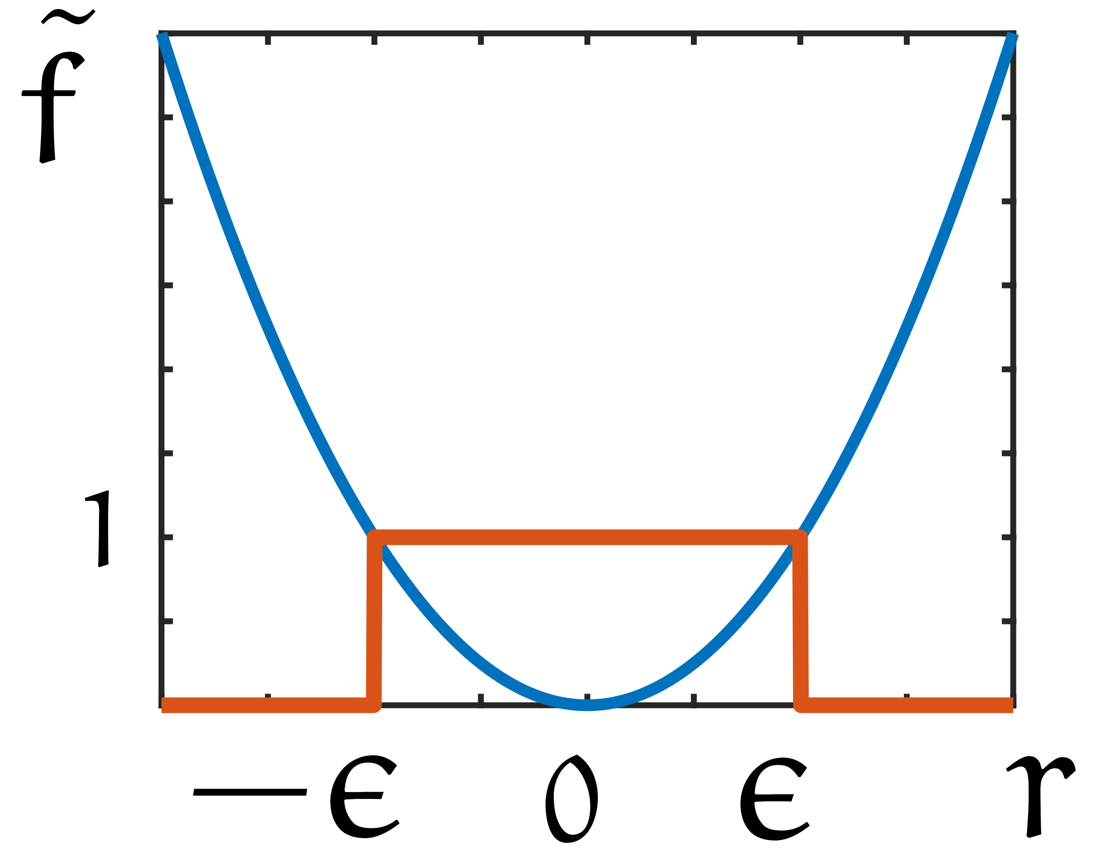
unstructured outliers

Robust single model fitting: consensus maximisation

Instead of r^2, consider a different cost function:

$$\theta^* = \arg\max_{\theta} \sum_{x \in X} \tilde{f}(r)$$

$$\tilde{f}(r) = \begin{cases} 1 & r \leq \epsilon \\ 0 & r > \epsilon \end{cases}$$

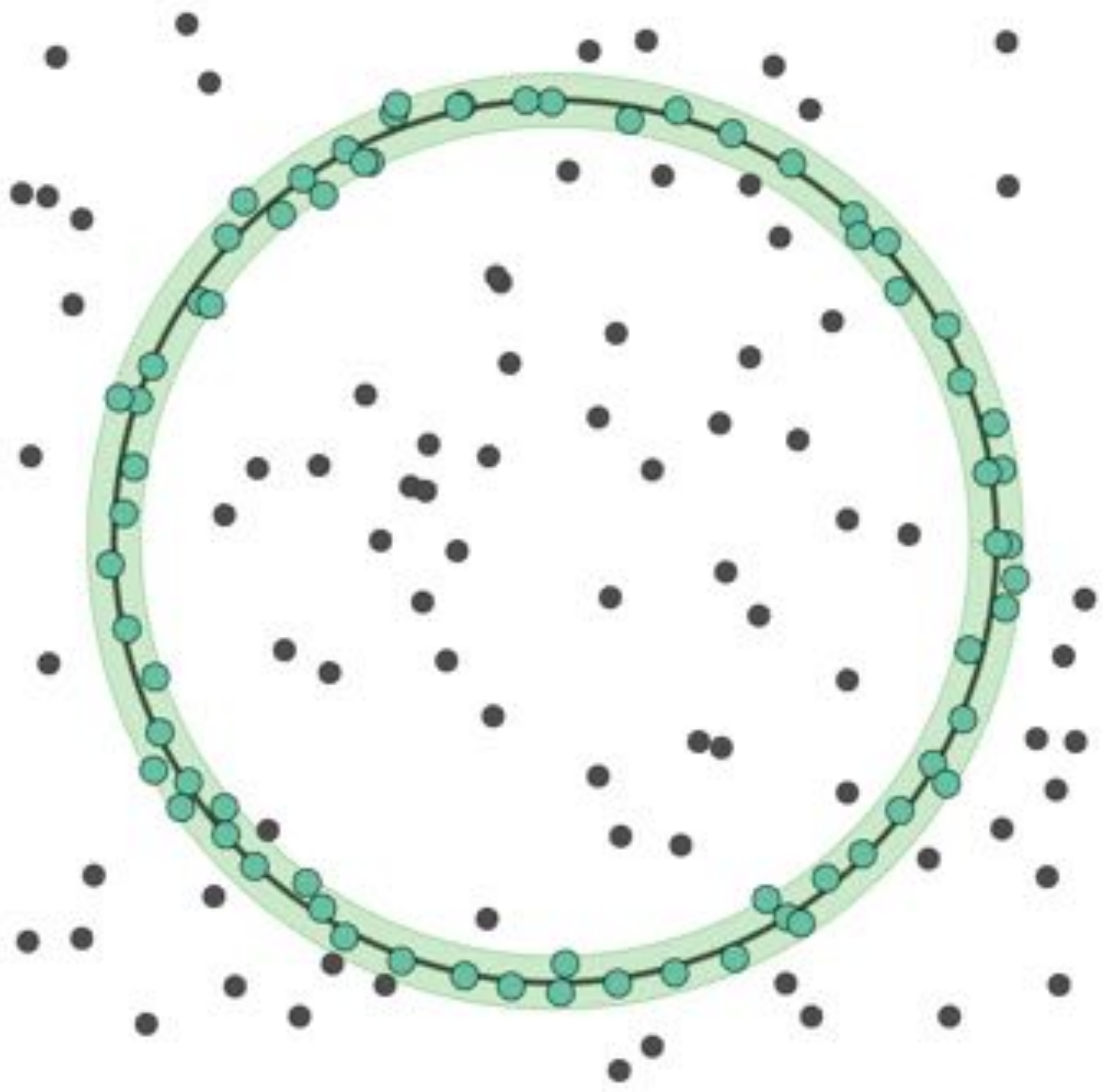


- inlier threshold ϵ
- consensus of model θ

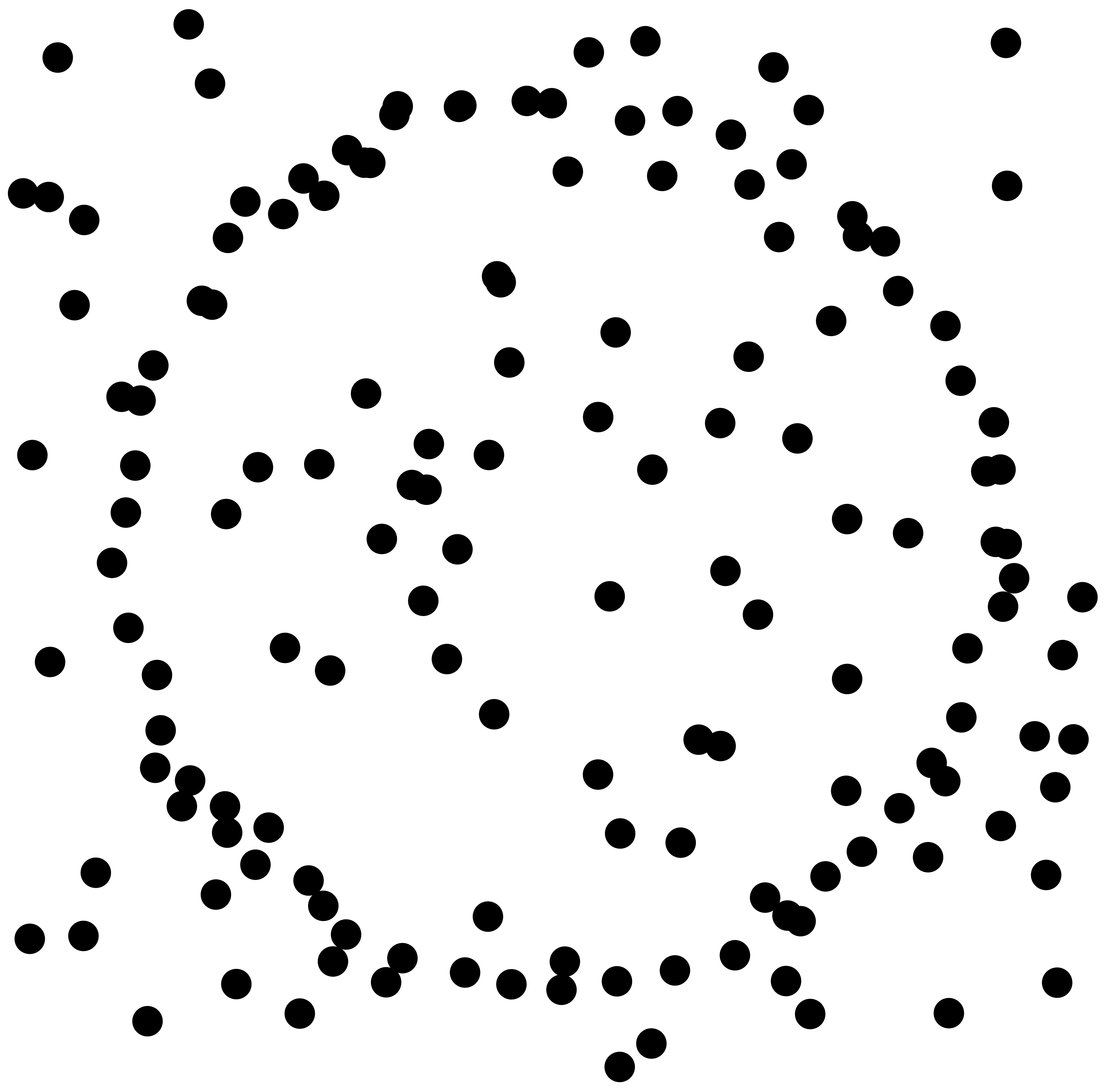
$$\sum_{x \in X} \tilde{f}_{\epsilon}(r(x, \theta))$$

- consensus set

$$\mathrm{CS}(\theta, \epsilon) = \{x \colon r(x, \theta) < \epsilon\}$$

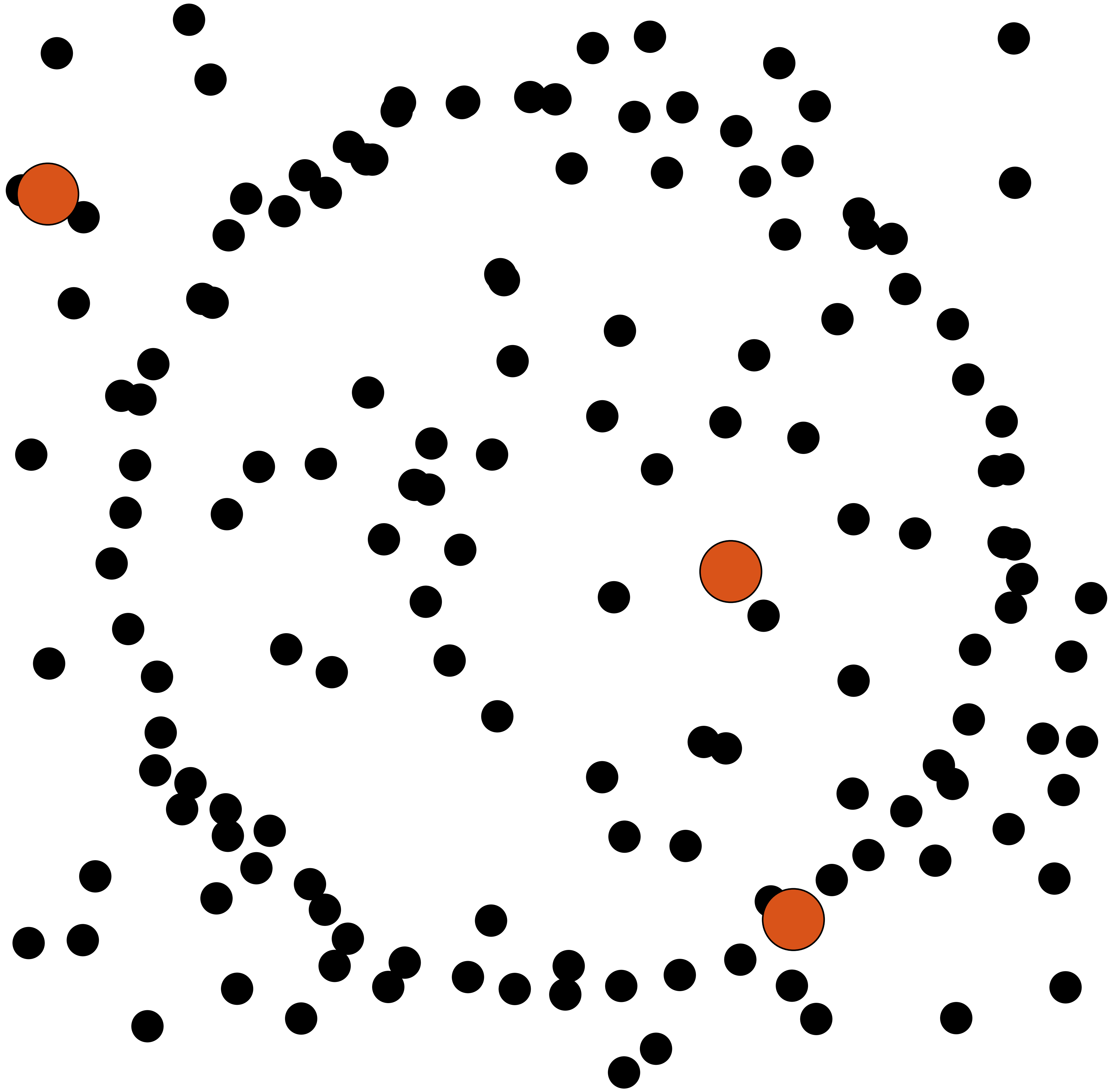


Randomized Sample Consensus [Fischler and Bolles 1981]



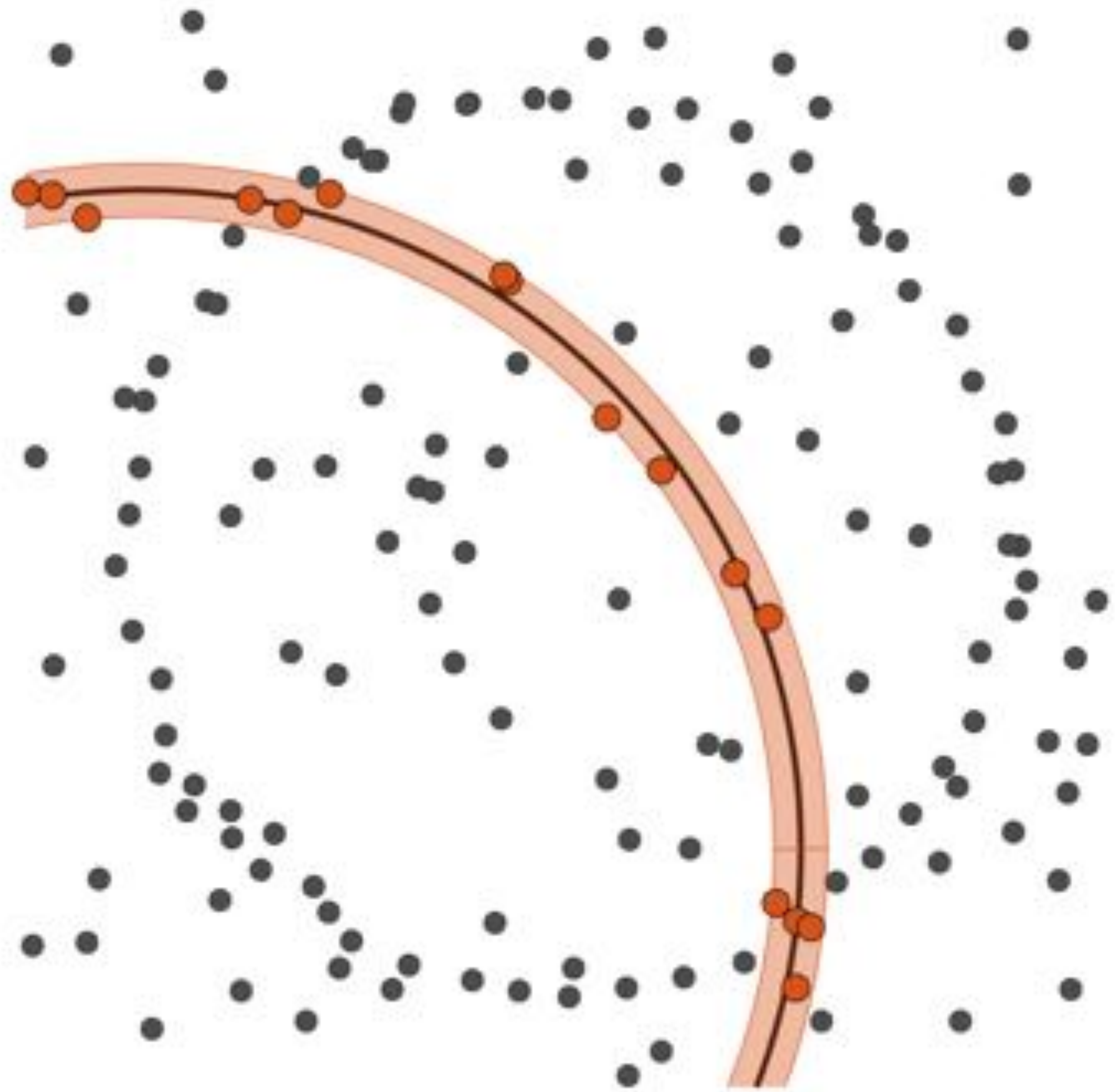
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta));$
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max};$
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



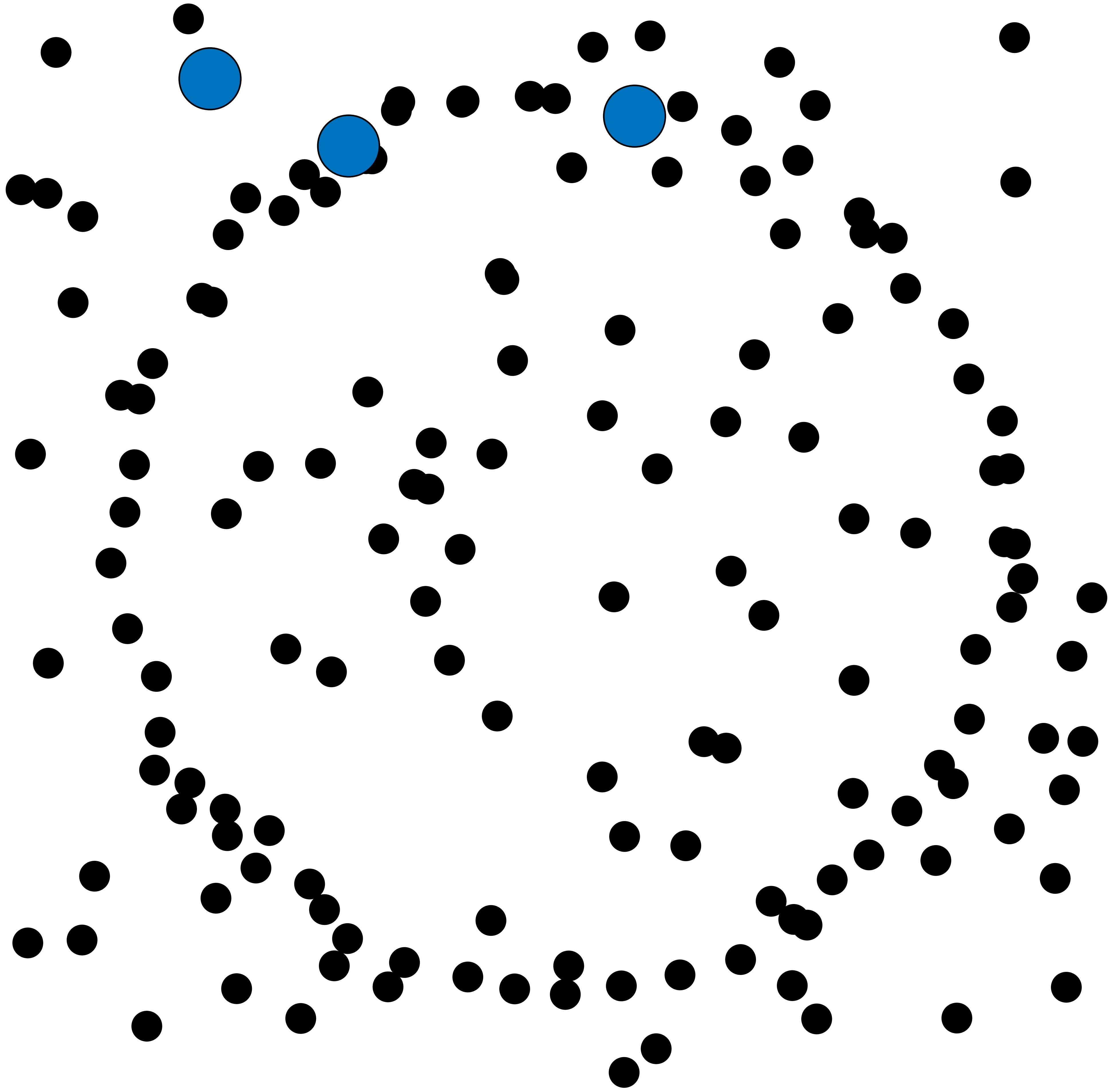
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{max}$;
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



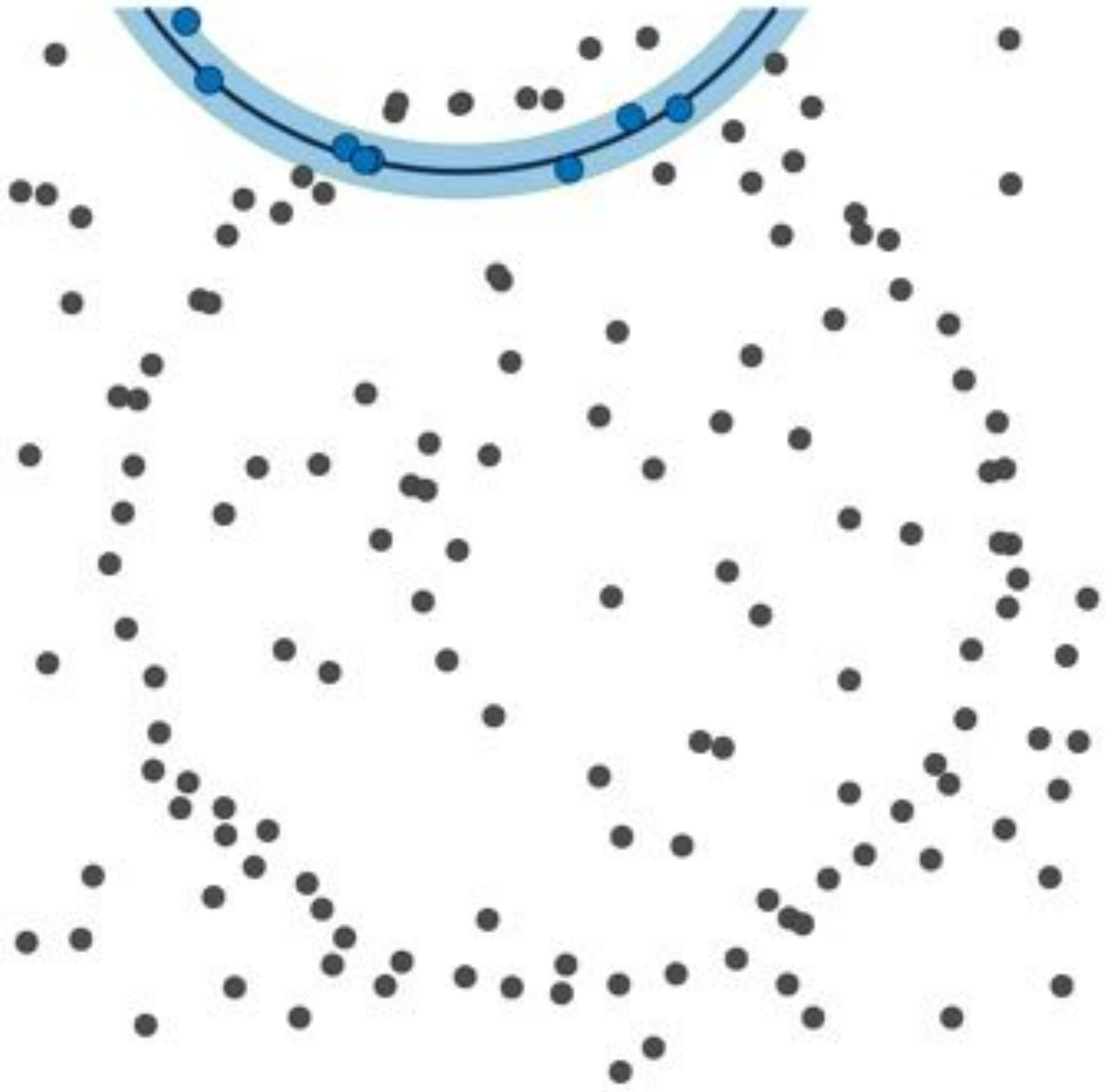
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta));$
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max}$;
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



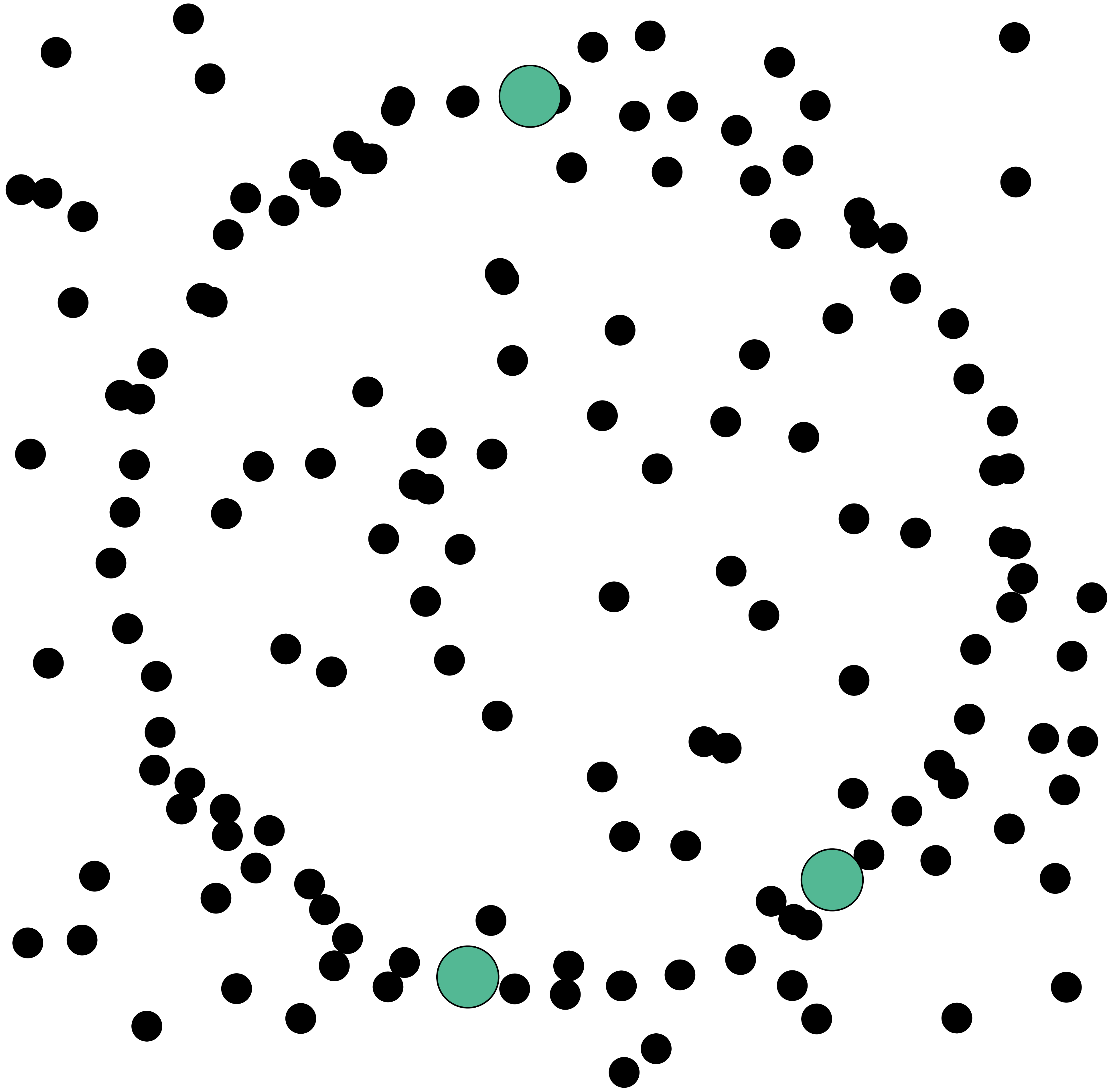
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max}$;
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



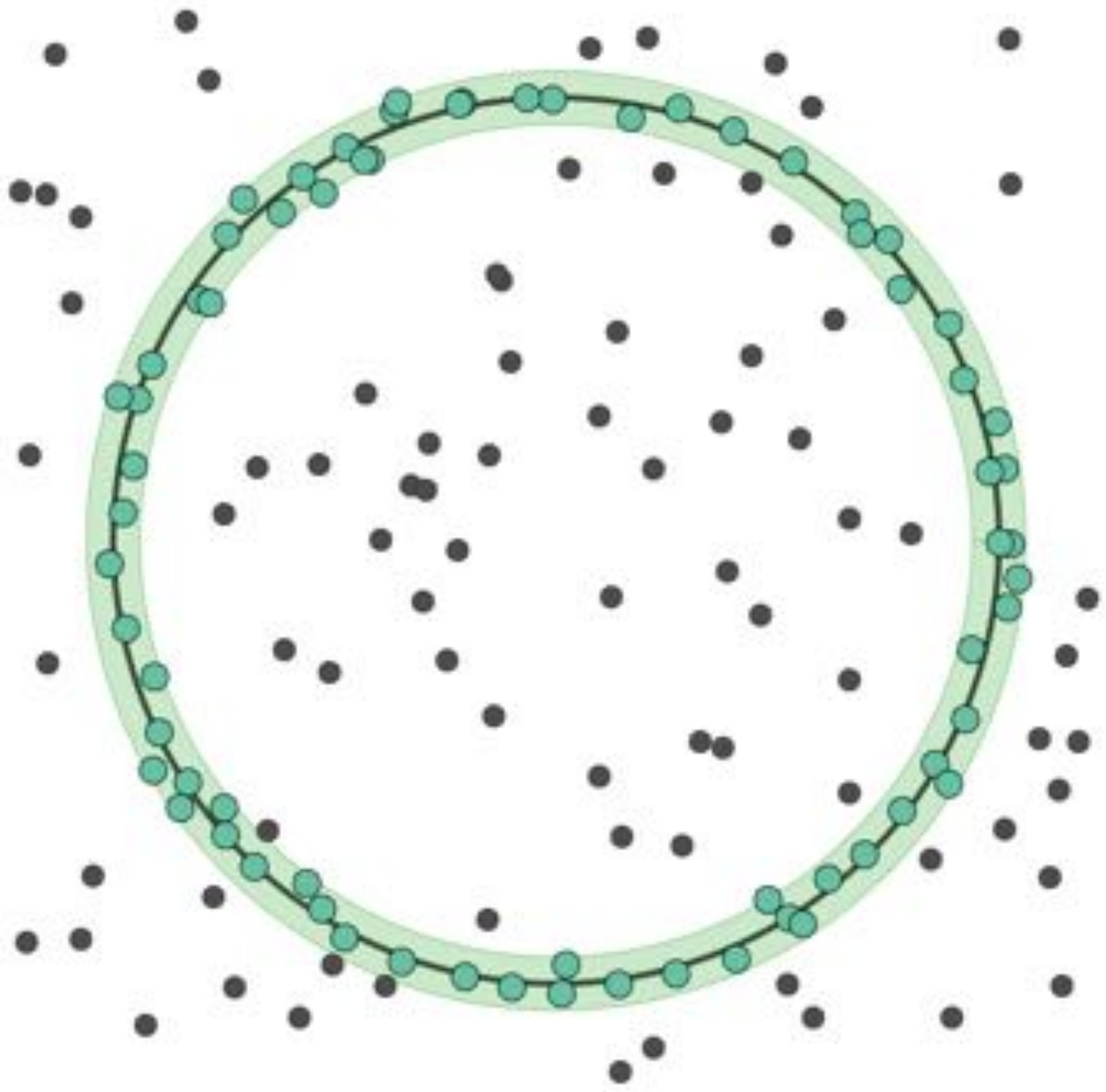
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta$;
    $J^* = J(\theta)$;
  end
  $k = k + 1$;
until $k > k_{\text{max}}$;
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



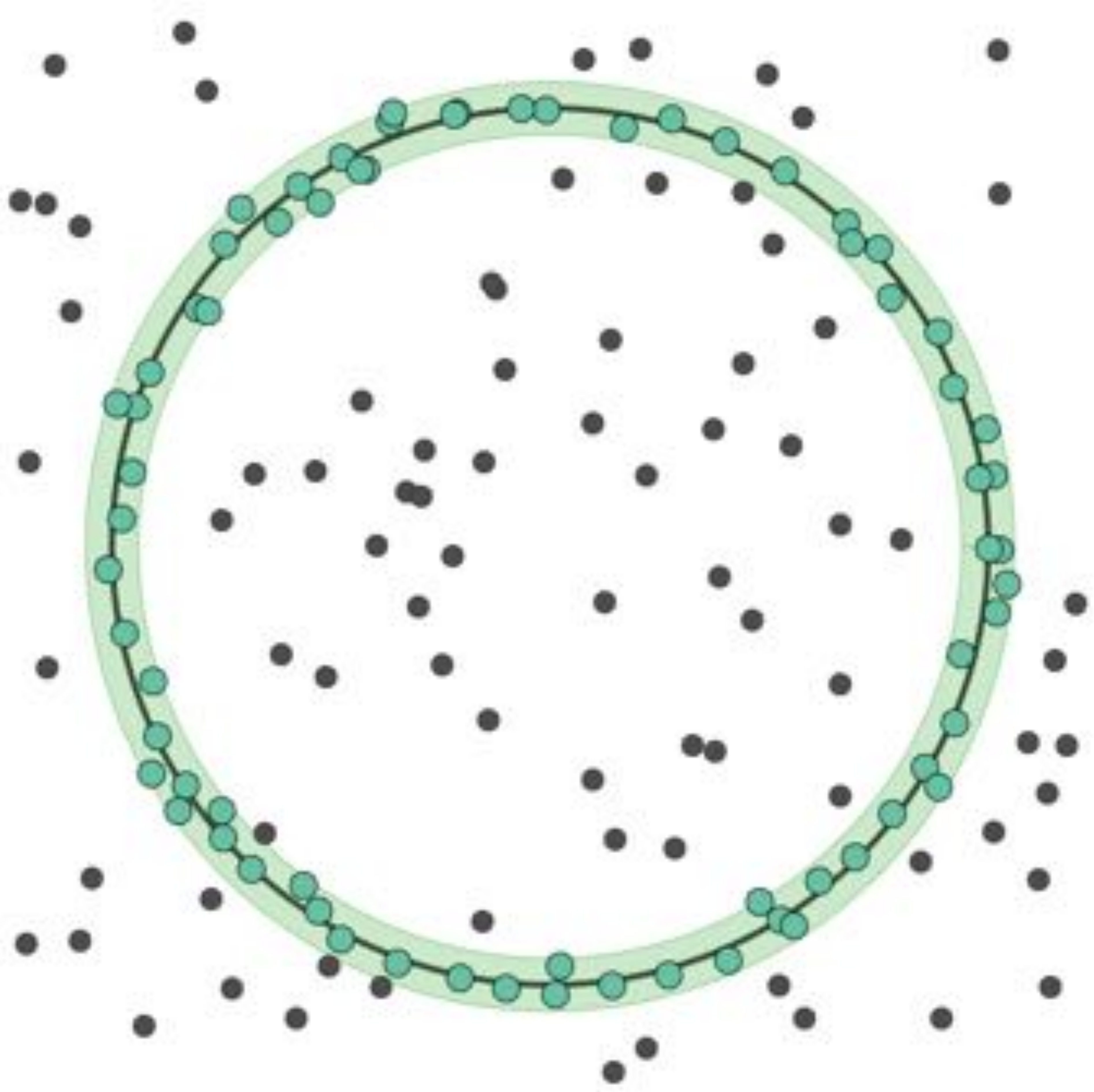
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X$;
  Estimate parameters θ on S;
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta))$;
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max}$;
Optimize θ^* on its inliers.

Randomized Sample Consensus [Fischler and Bolles 1981]



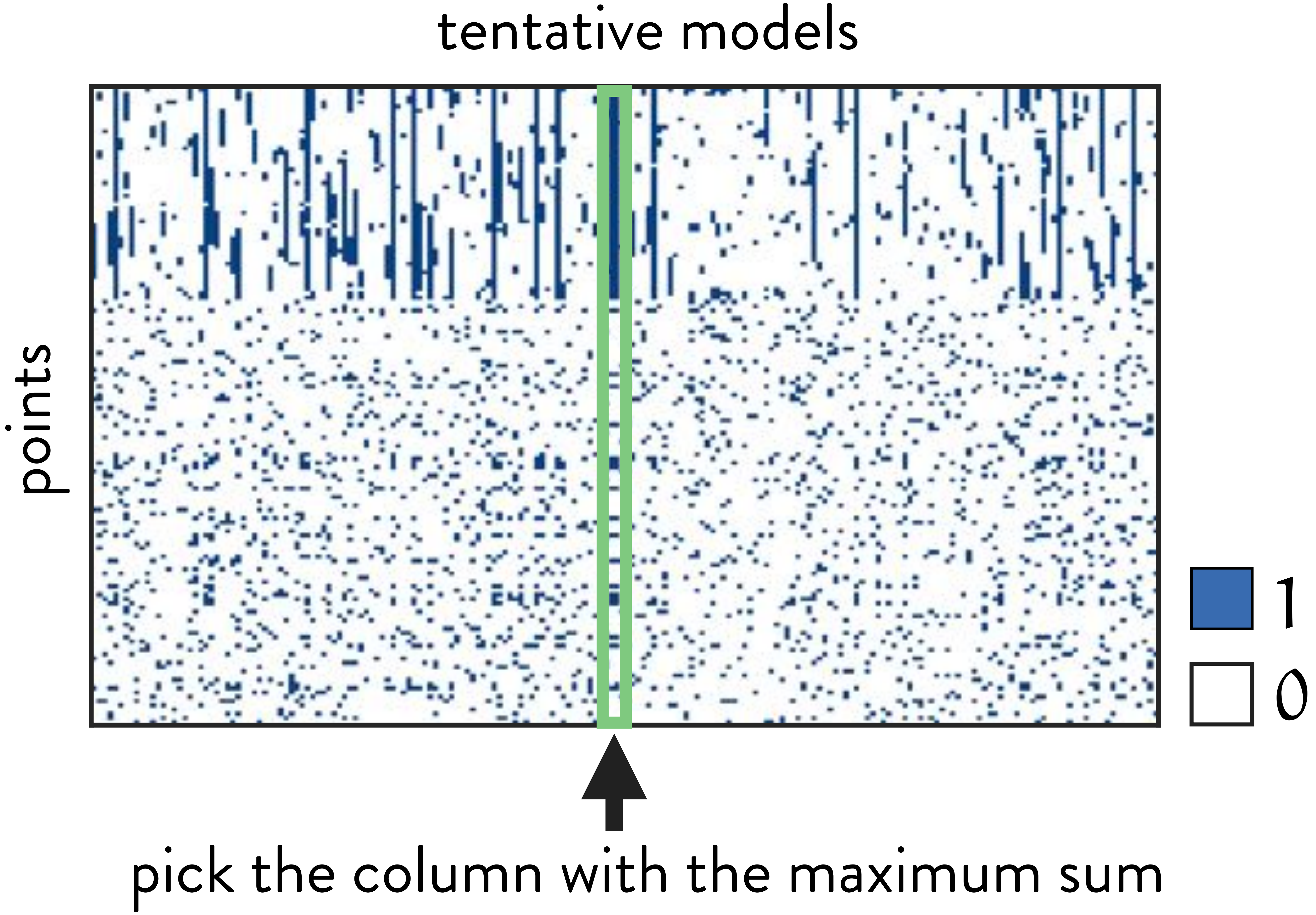
Input: X data, ϵ inlier threshold, k_{max} max iteration
Output: θ^* model estimate
$J^* = -\infty, k = 0;$
repeat
  Select randomly a minimal sample set $S \subset X;$
  Estimate parameters θ on $S;$
  Evaluate $J(\theta) = \sum_{x \in X} \hat{f}_\epsilon(r(x, \theta));$
  if $J(\theta) > J^*$ **then**
    $\theta^* = \theta;$
    $J^* = J(\theta);$
  end
  $k = k + 1;$
until $k > k_{max};$
Optimize θ^* on its inliers.

Randomized Sample Consensus

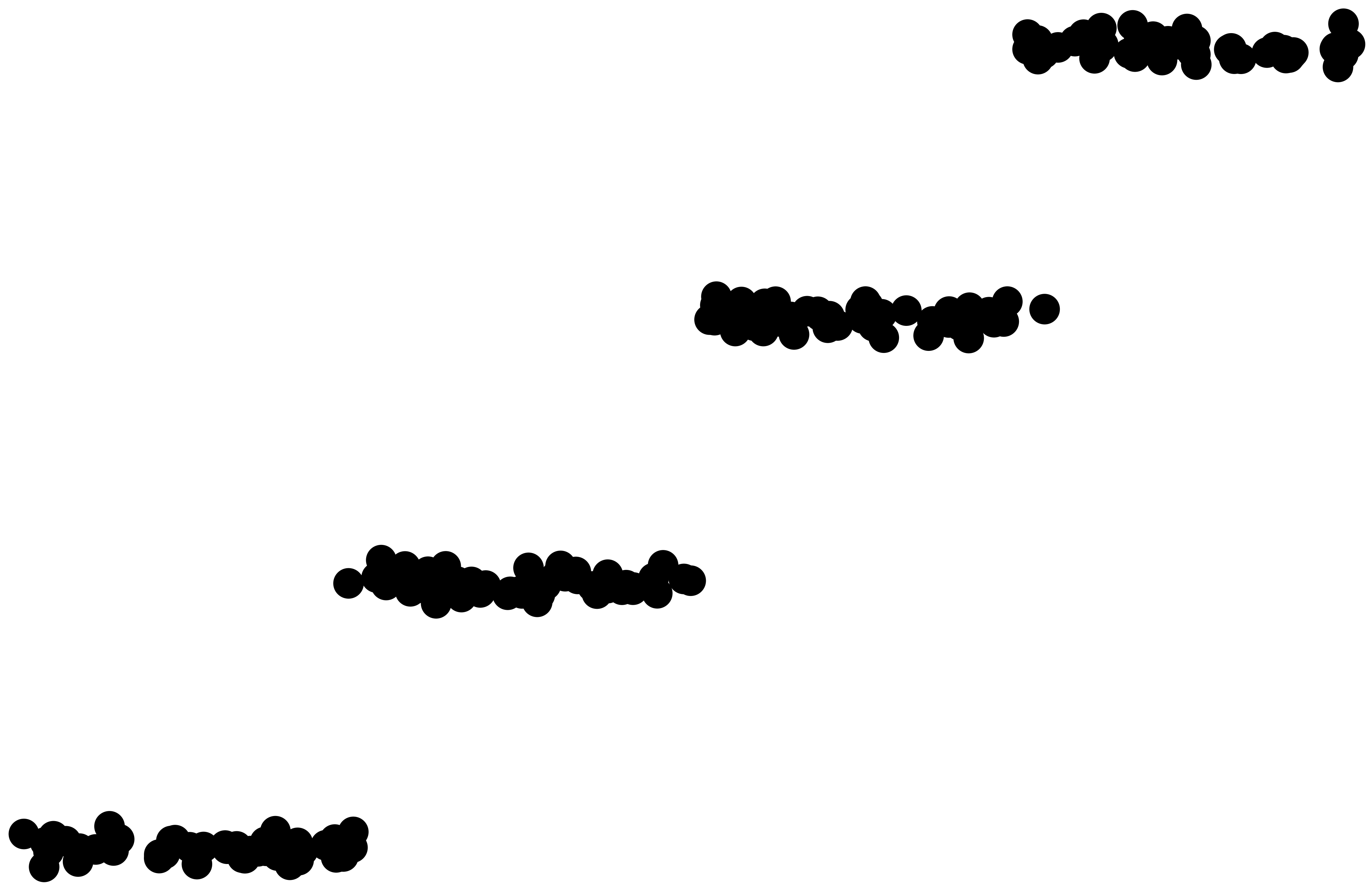


Data driven search of model space

$$H = \{\theta_1, \theta_2, \ldots, \theta_m\} \approx \Theta$$



Randomized Sample Consensus



Line fitting example

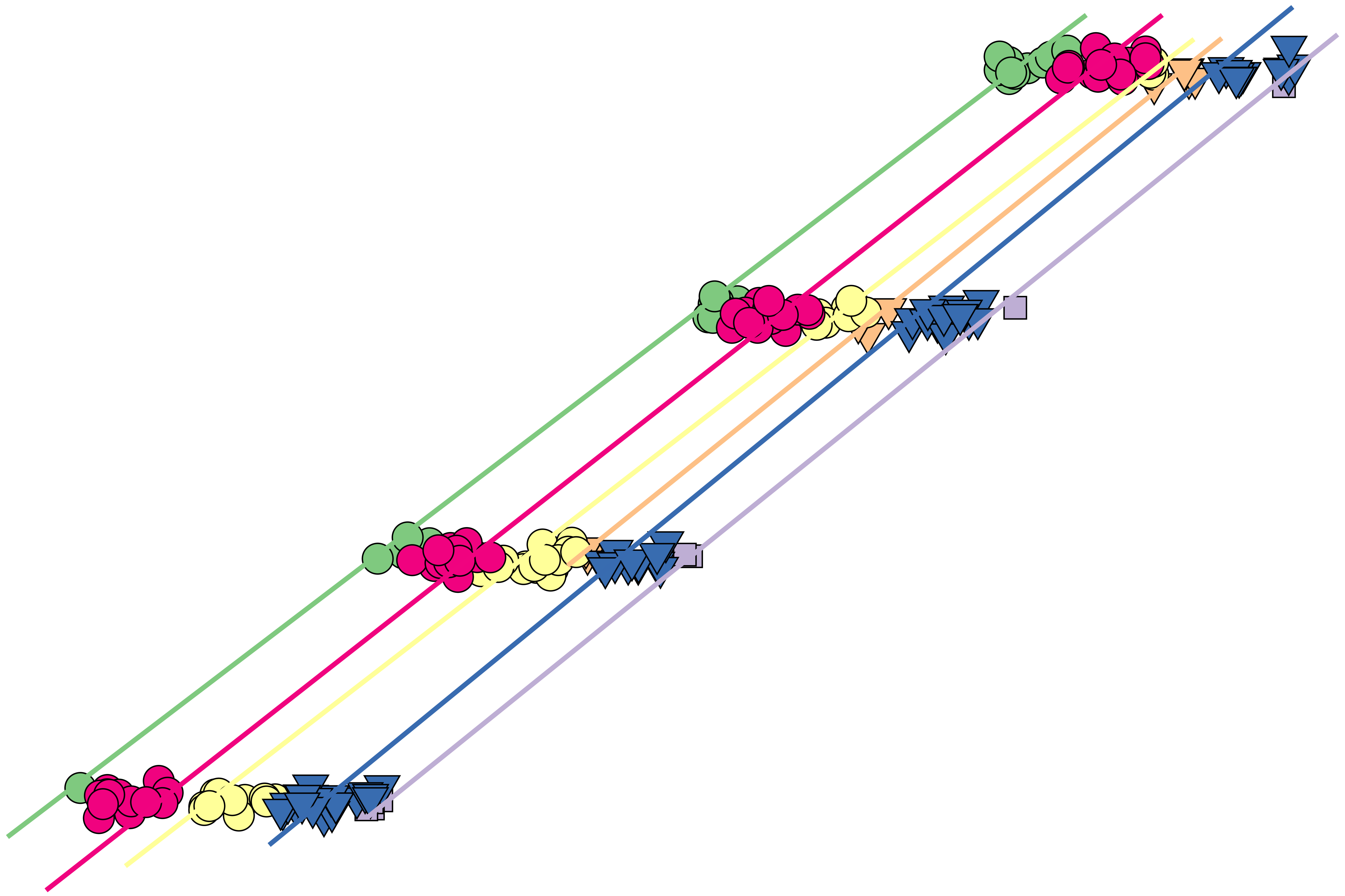
Pros:

- very popular (>22900 citations in Google Scholar)
- many improvements have been proposed
- very versatile
- agnostic on outlier percentage
- mild assumption: know the scale noise

Cons:

- can take longer than expected
- does not fit well with the multi-model scenario

Randomized Sample Consensus [Zuliani 05]



Line fitting example

Pros:

- very popular (>17000 citations in Google Scholar)
- many improvements have been proposed
- very versatile
- agnostic on outlier percentage
- mild assumption: know the scale noise

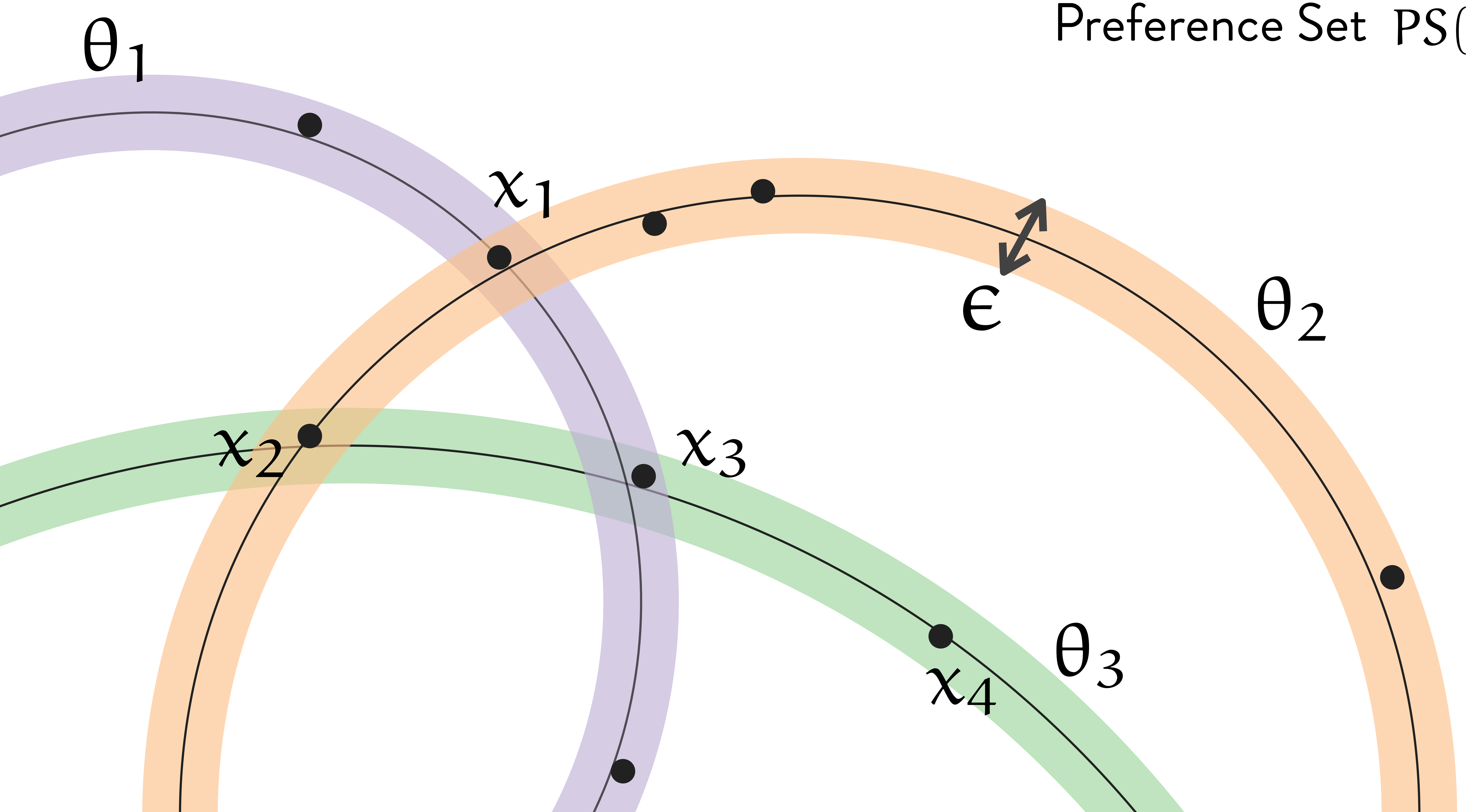
Cons:

- can take longer than expected
- does not fit well with the multi-model scenario

Multi model fitting: from consensus to preferences

Consensus Set $CS(\theta) = \{x \colon r(x, \theta) < \epsilon\}$

Preference Set $PS(x) = \{\theta \colon r(x, \theta) < \epsilon\}$

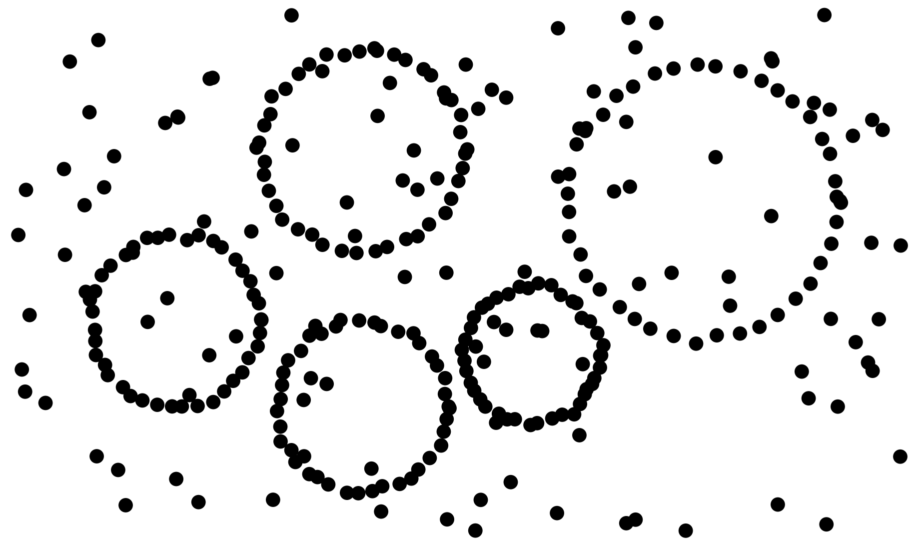


$CS(\theta_3) = \{x_2, x_3, x_4\}$

$PS(x_1) = \{\theta_1, \theta_2\}$

$PS(x_2) = \{\theta_2, \theta_3\}$

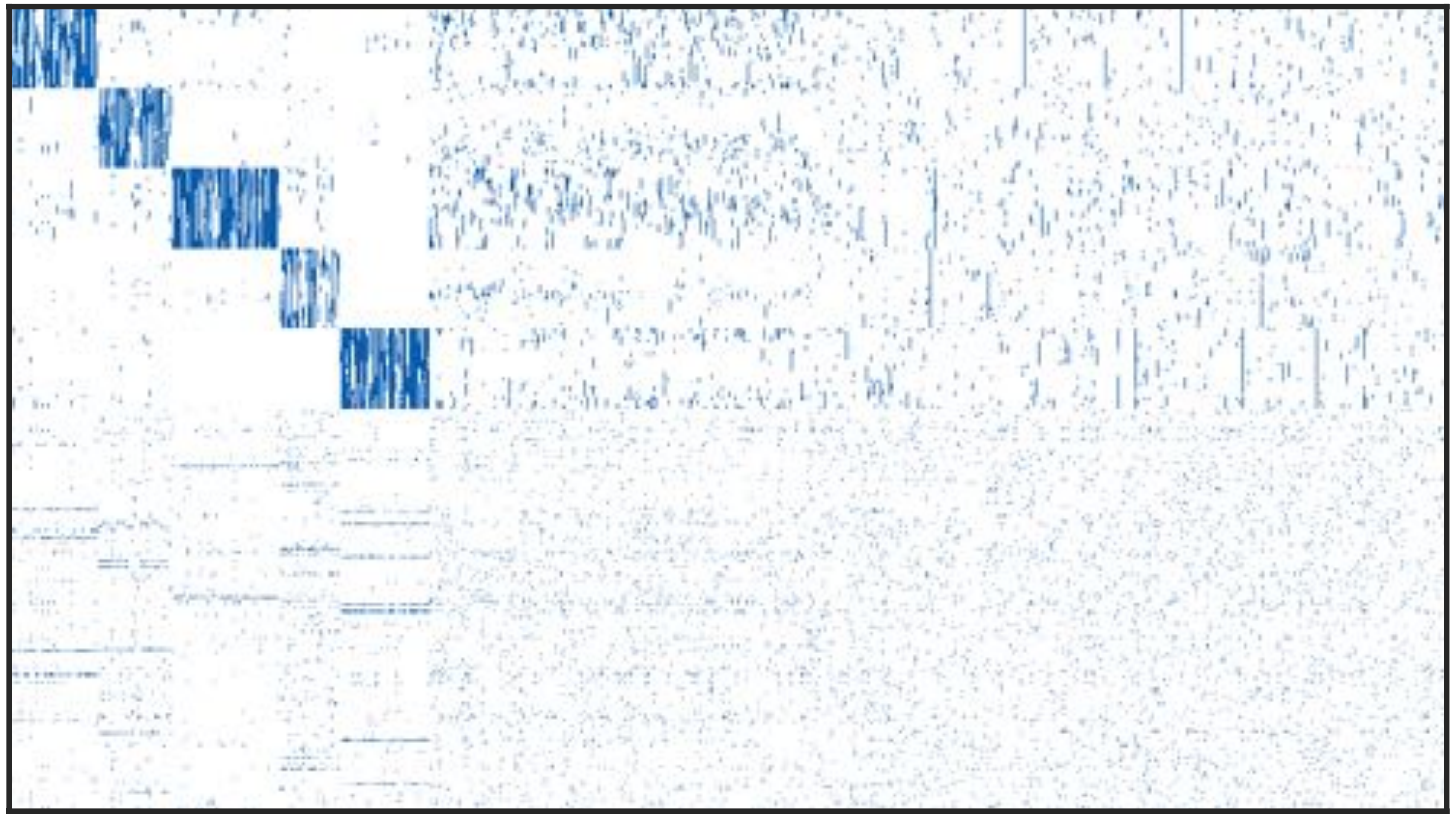
Multi model fitting: the preference trick



- Generate a pool of m random models (as in RanSaC)

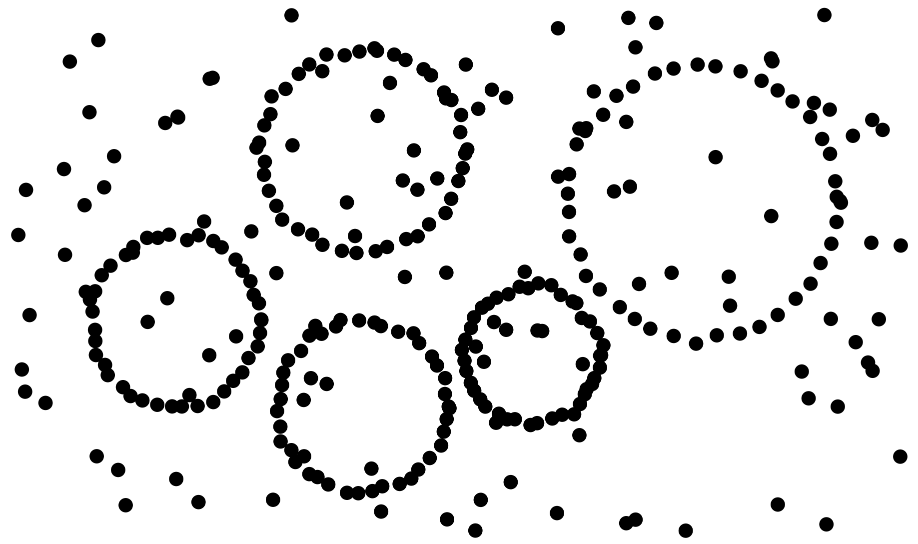
$$H = \{\theta_1, \theta_2, \ldots, \theta_m\}$$

- Build the matrix:



points and models reordered for visualisation purposes

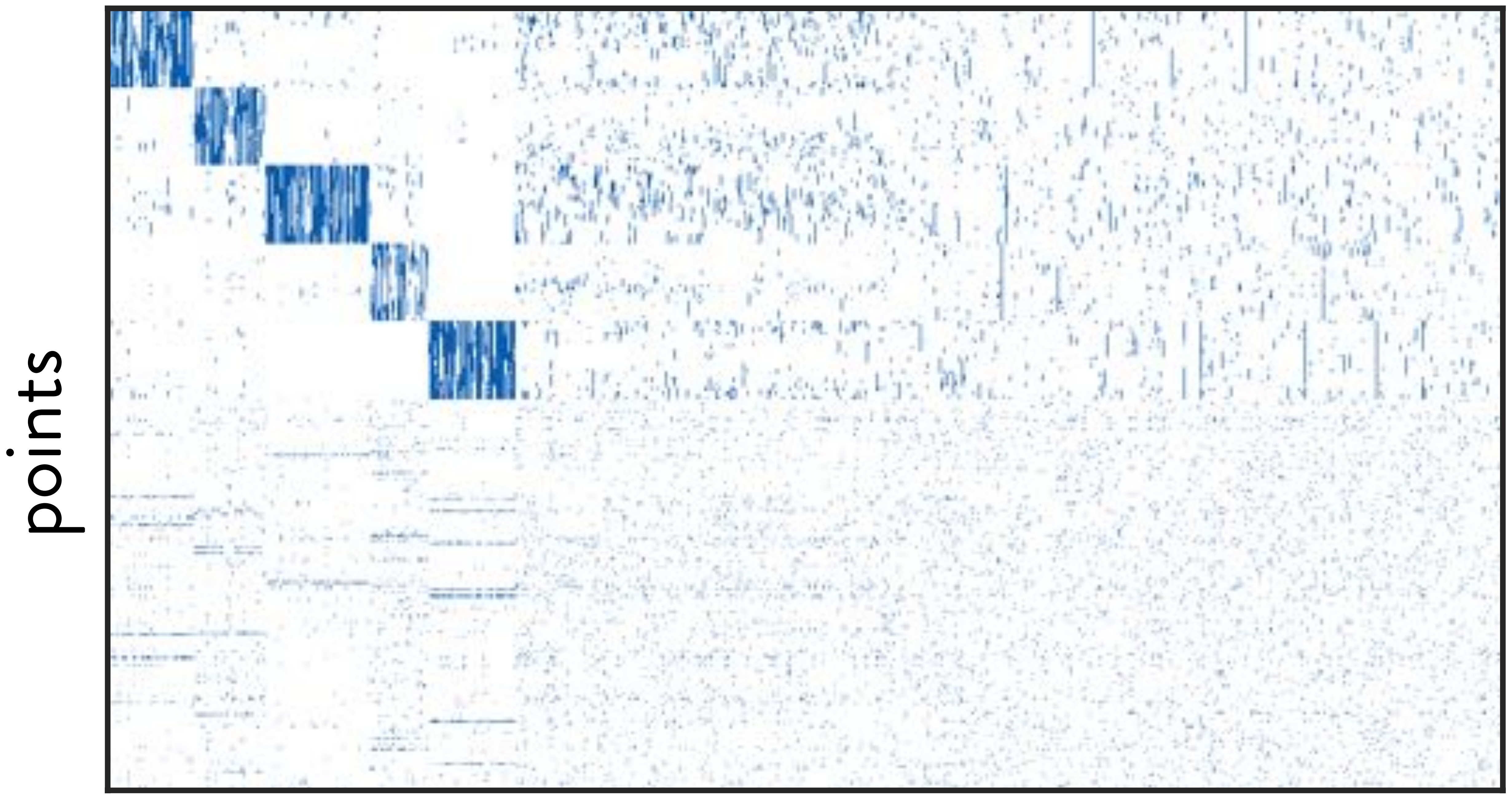
Multi model fitting: the preference trick



Consensus Set $CS(\theta) = \{x \colon r(x, \theta) < \epsilon\}$ columns

Preference Set $PS(x) = \{\theta \colon r(x, \theta) < \epsilon\}$ rows

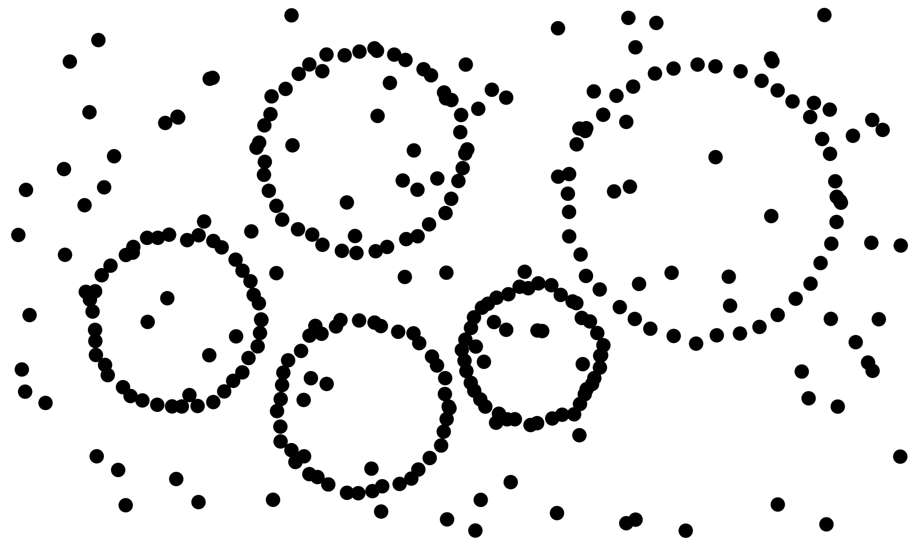
tentative models



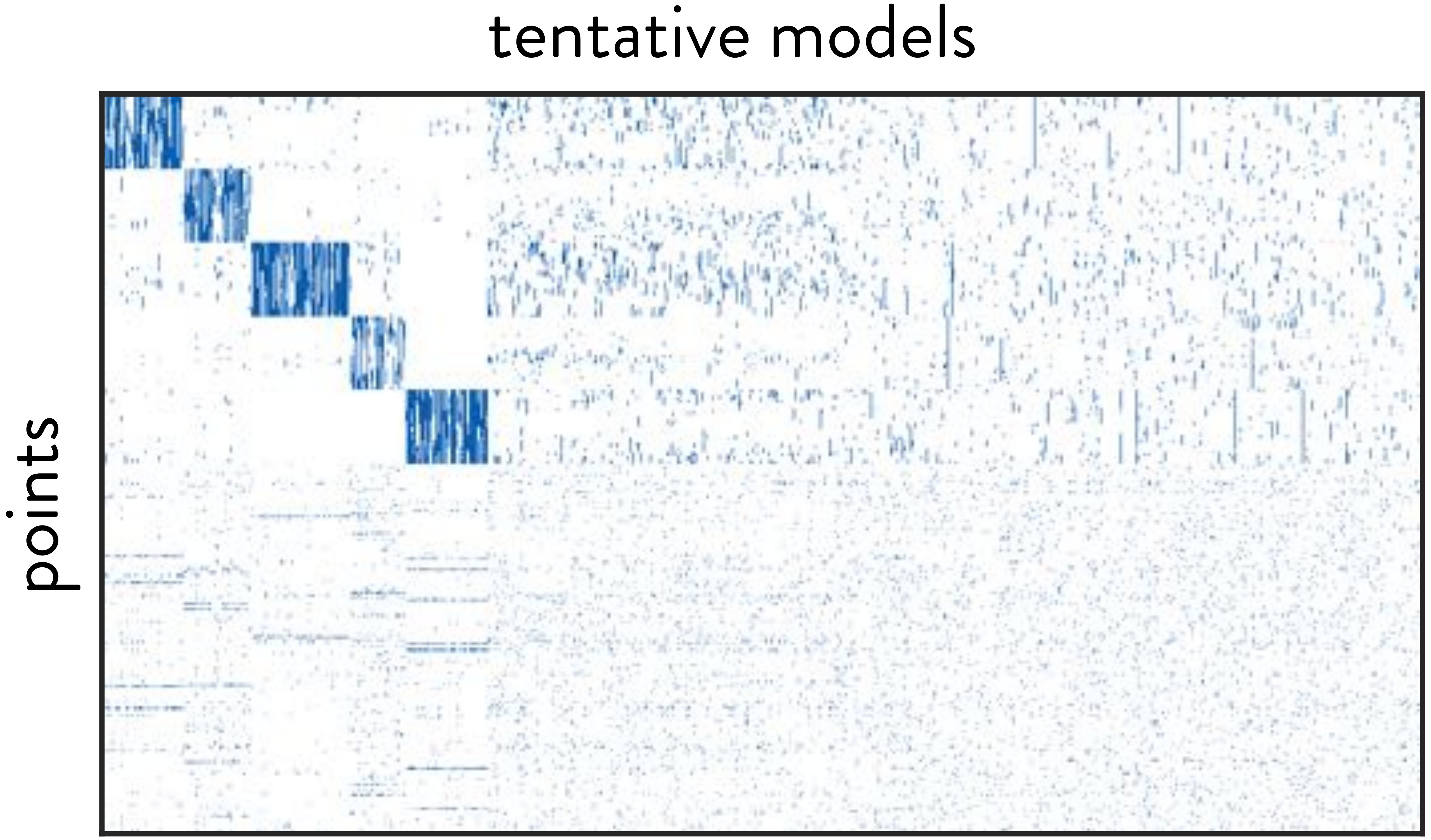
points and models reordered for visualisation purposes

Preference Matrix

Multi model fitting: the preference trick



- Point $\longleftrightarrow$ subset of preferred sampled models
- Block diagonal matrix $\Longrightarrow$ point of the same structure have similar preferences

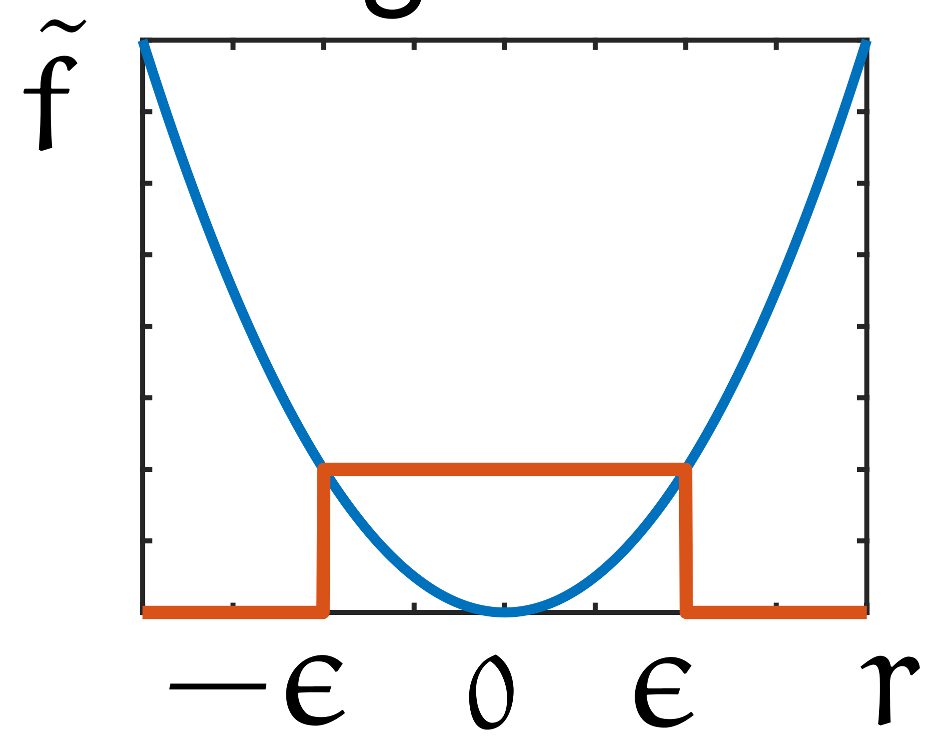


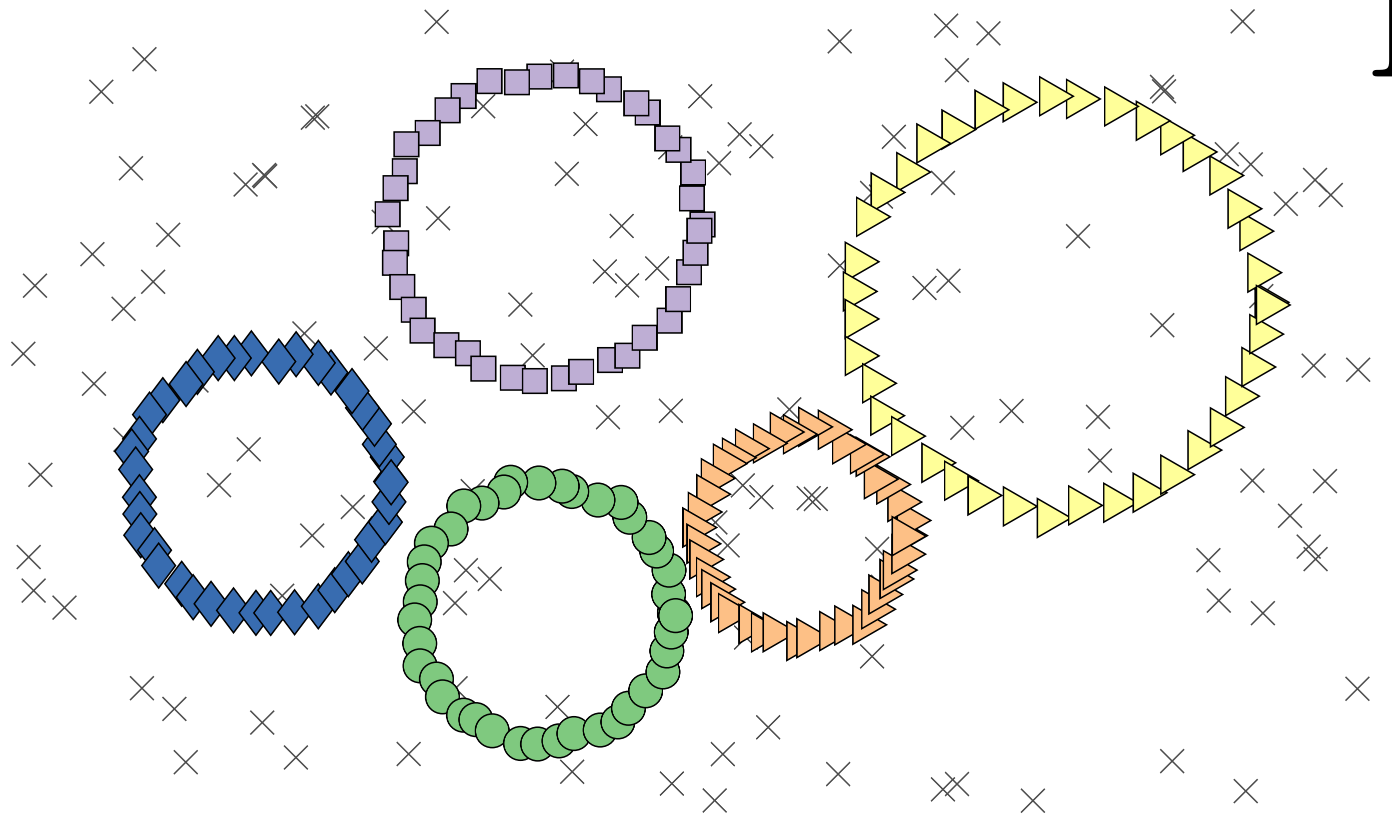
points and models reordered for visualisation purposes

Preference Matrix

Multi model fitting: a lift to Preference Space

Voting function

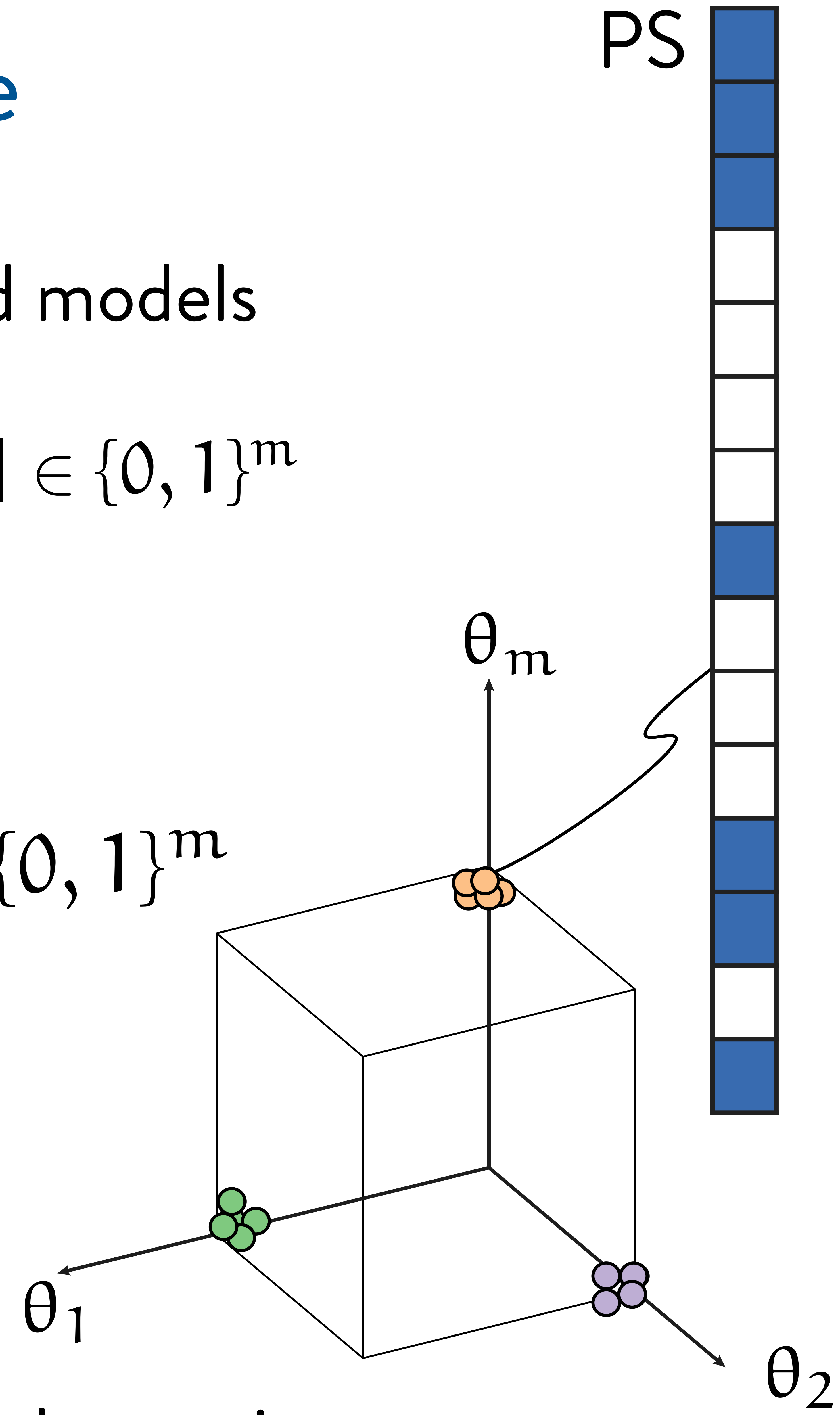




vector of binary votes to sampled models

$$x \mapsto [\hat{f}(r(x,\theta_1)),\ldots,\hat{f}(r(x,\theta_m))] \in \{0,1\}^m$$

$\mathbb{R}^d \longrightarrow \{0,1\}^m$

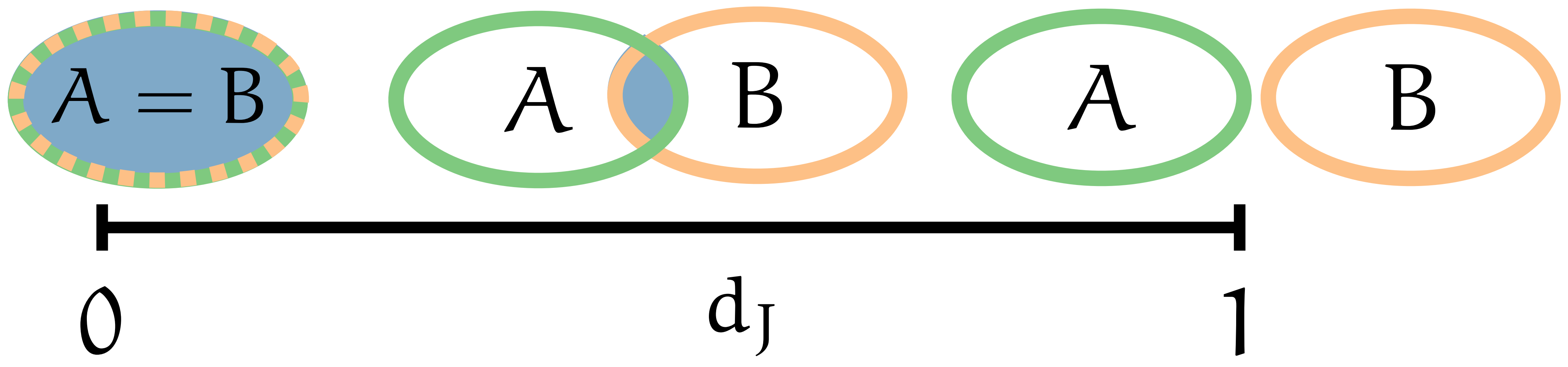


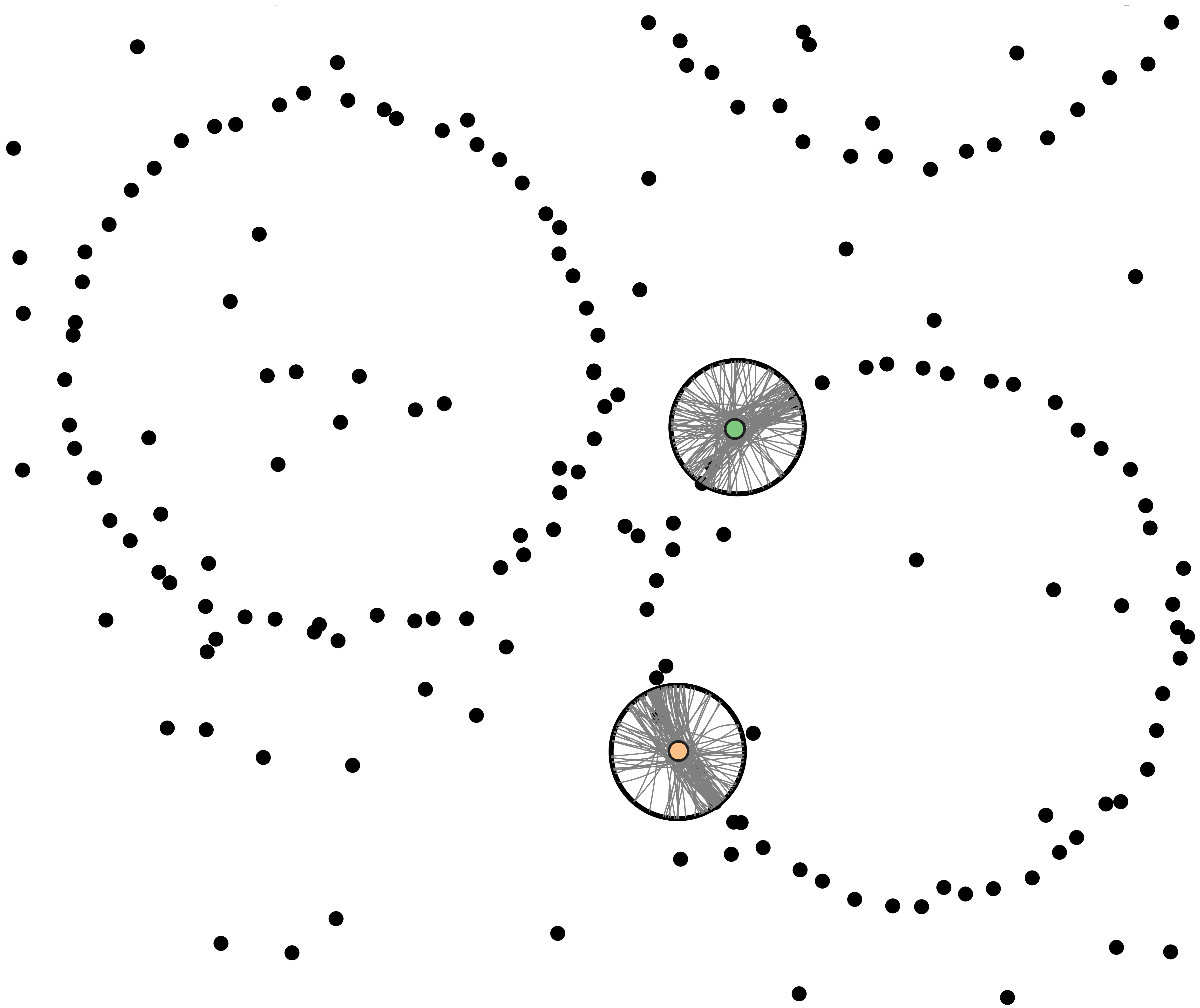
structures are regions of close points

Multi model fitting: a lift to Preference Space

The **Jaccard distance** can be used to measure distance between PS.

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

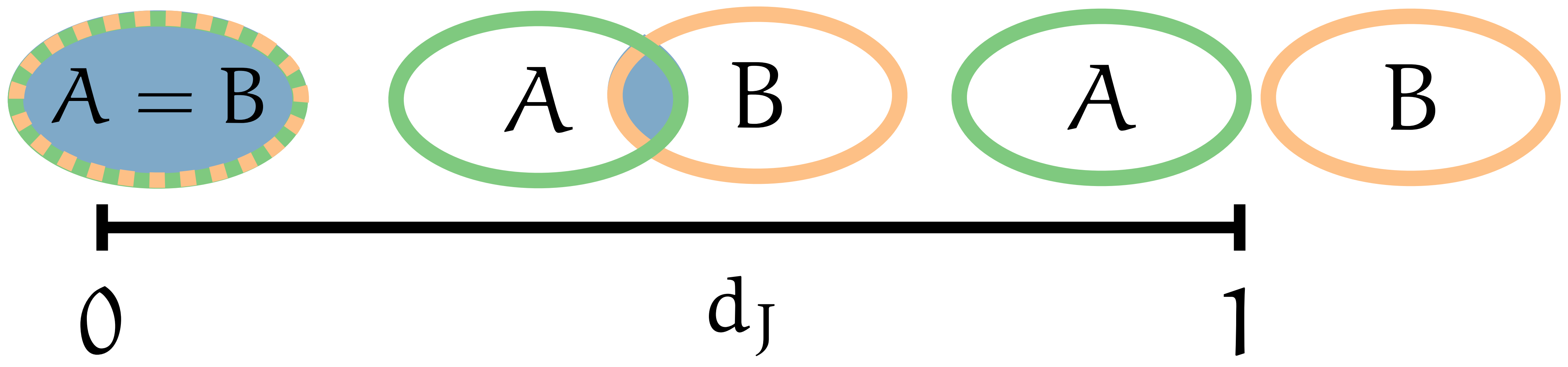


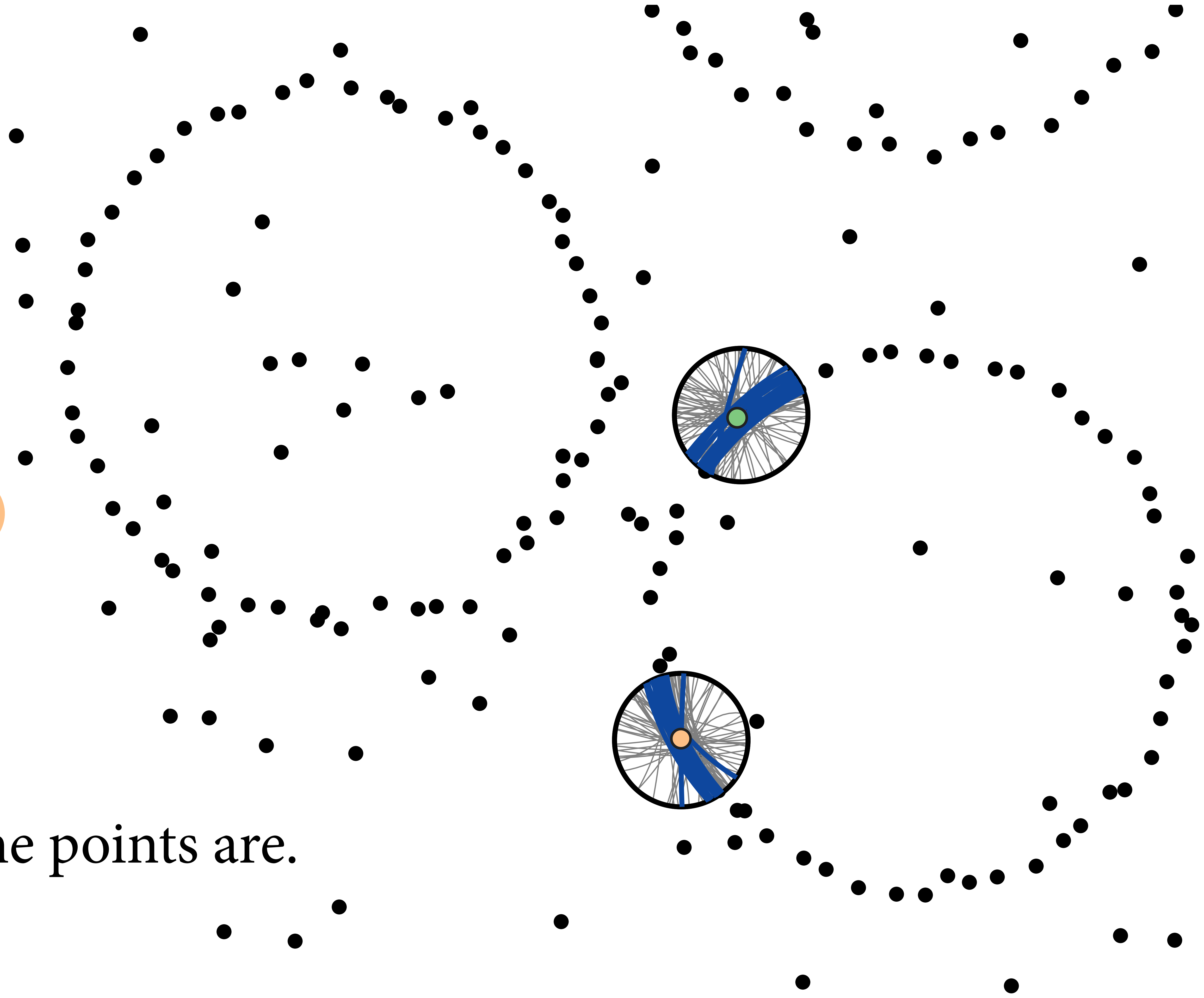


Multi model fitting: a lift to Preference Space

The **Jaccard distance** can be used to measure distance between PS.

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

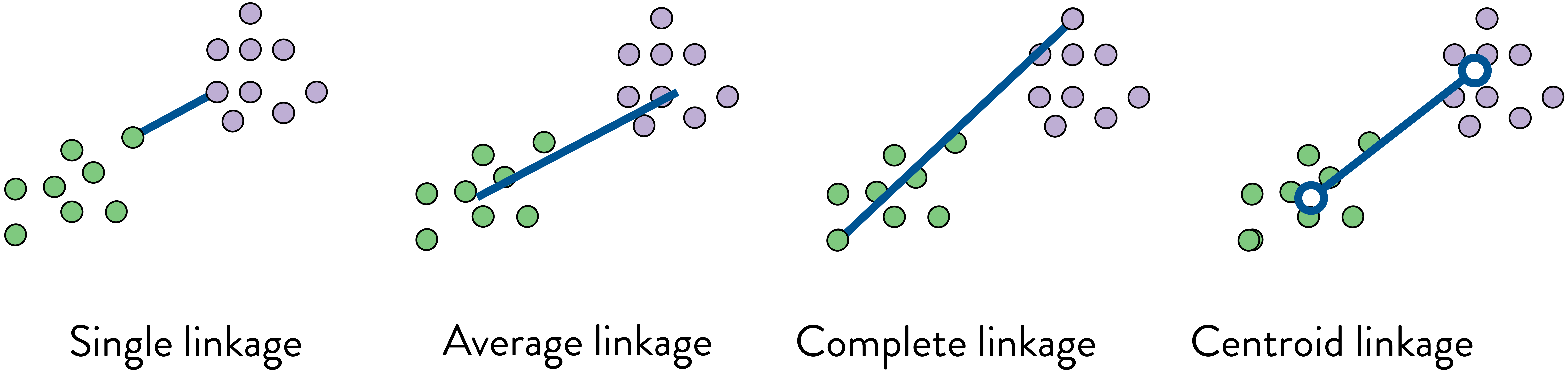




the more **models in common**, the closer the points are.

Linkage clustering in Preference Space

Hierarchical clustering can be used in the Preference Space to recover the structures



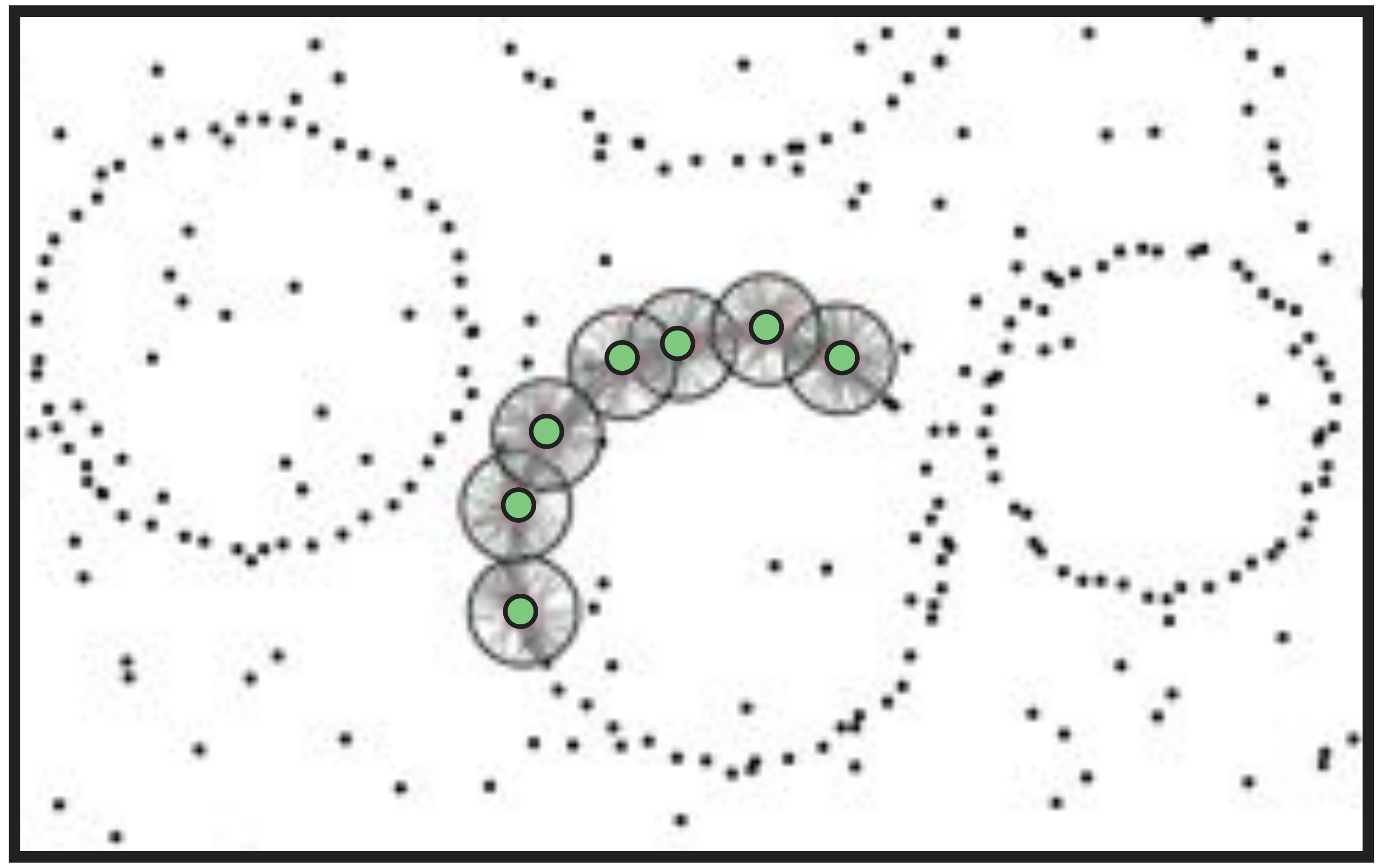
Instead of using centroids, we can derive a conceptual representation for a cluster…

J-linkage clustering [Toldo and Fusiello, ECCV 08]

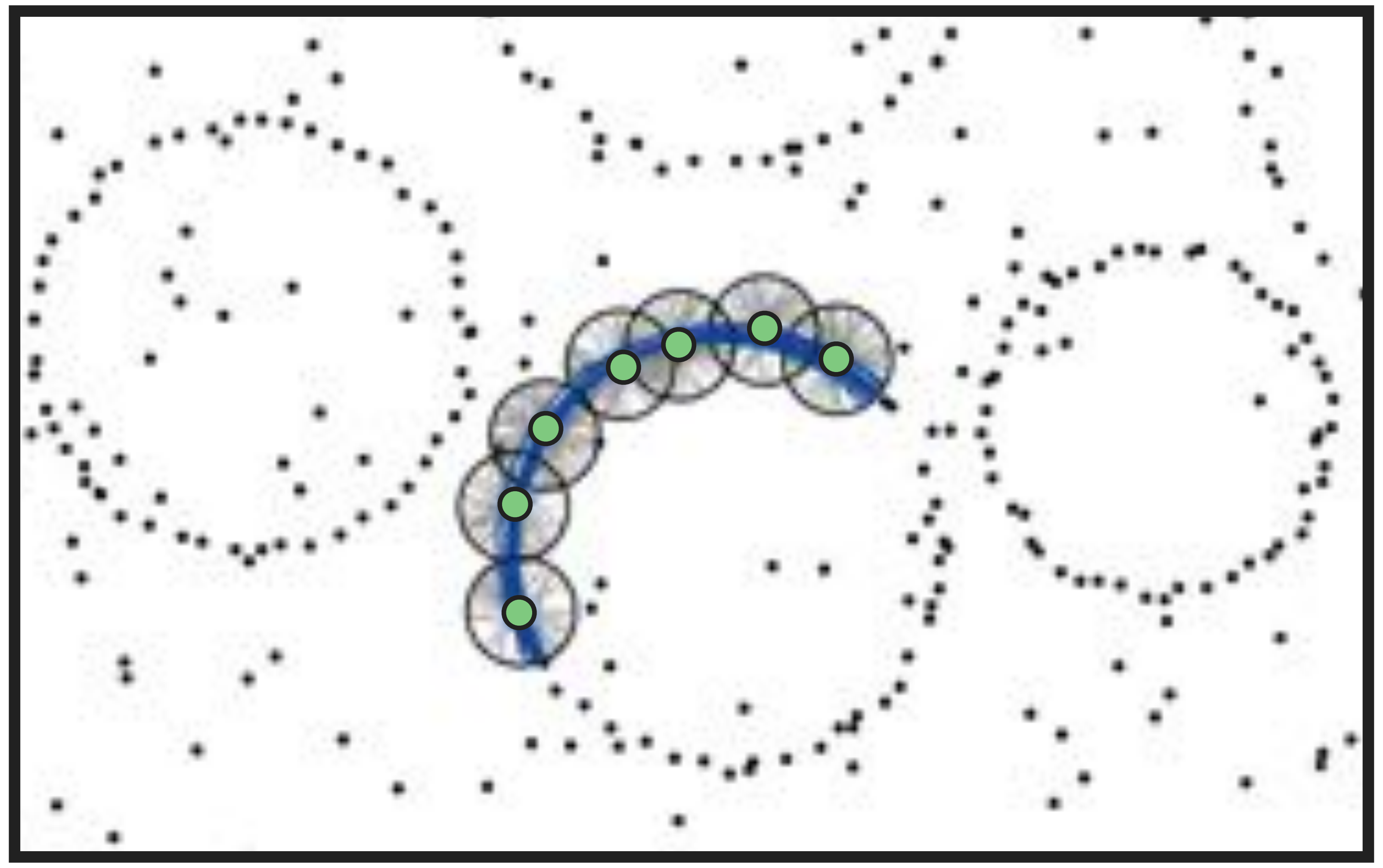
The representation of a cluster in the conceptual space is the intersection of the PS of its points

$$U \subseteq X, \mathrm{PS}(U) = \bigcap_{x \in U} \mathrm{PS}(x)$$

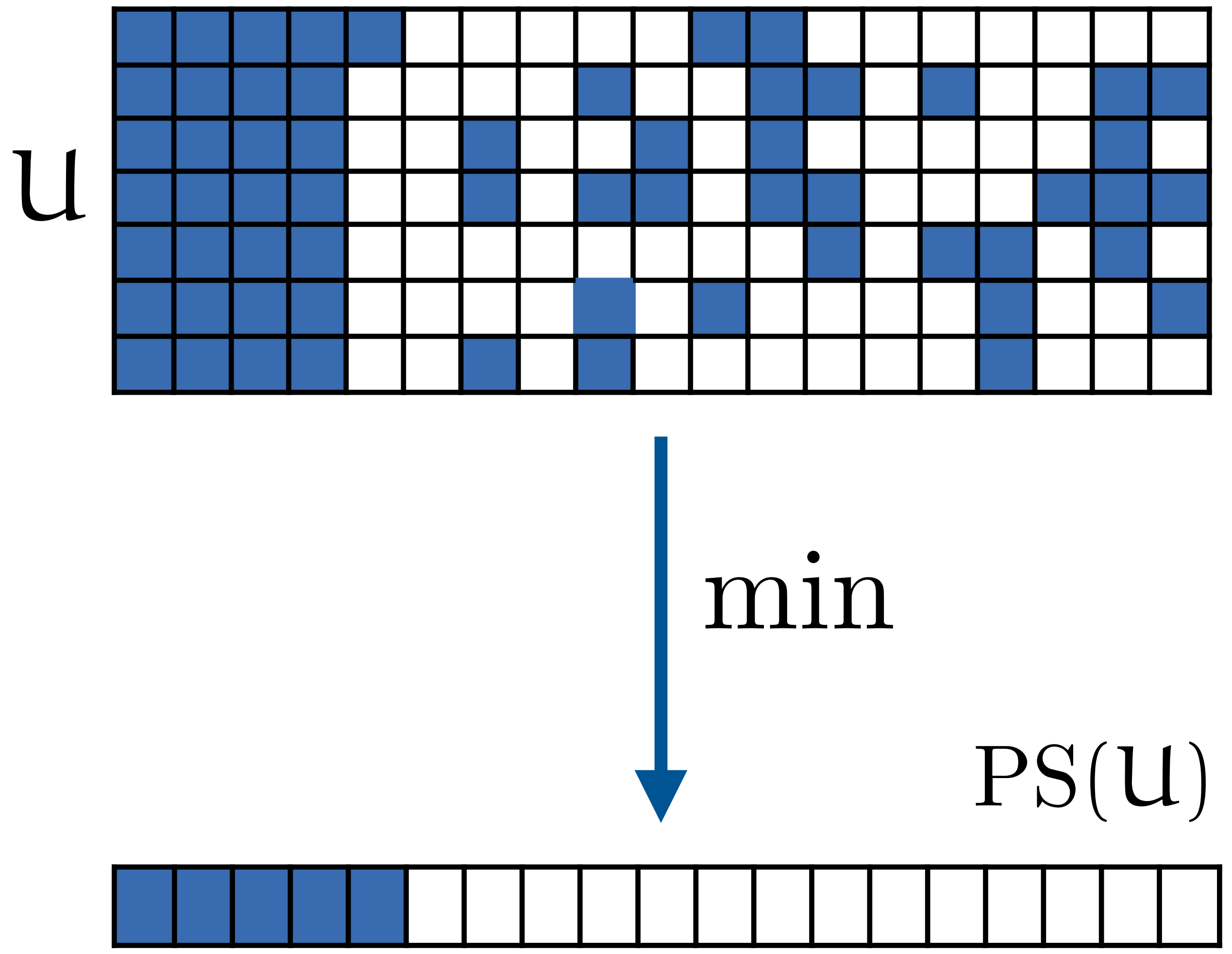
Simply computed as the component-wise min of rows in the preference matrix.



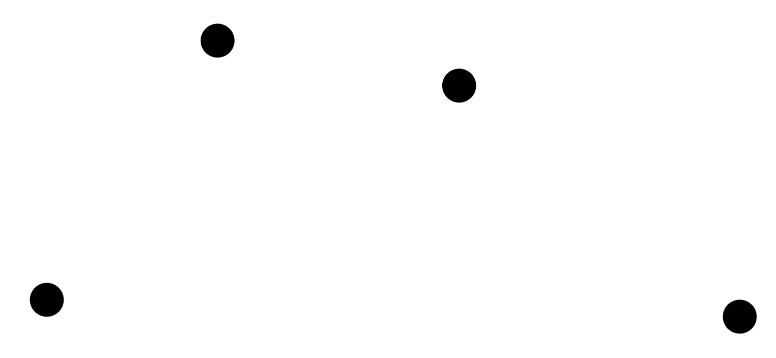
$U \subseteq X$

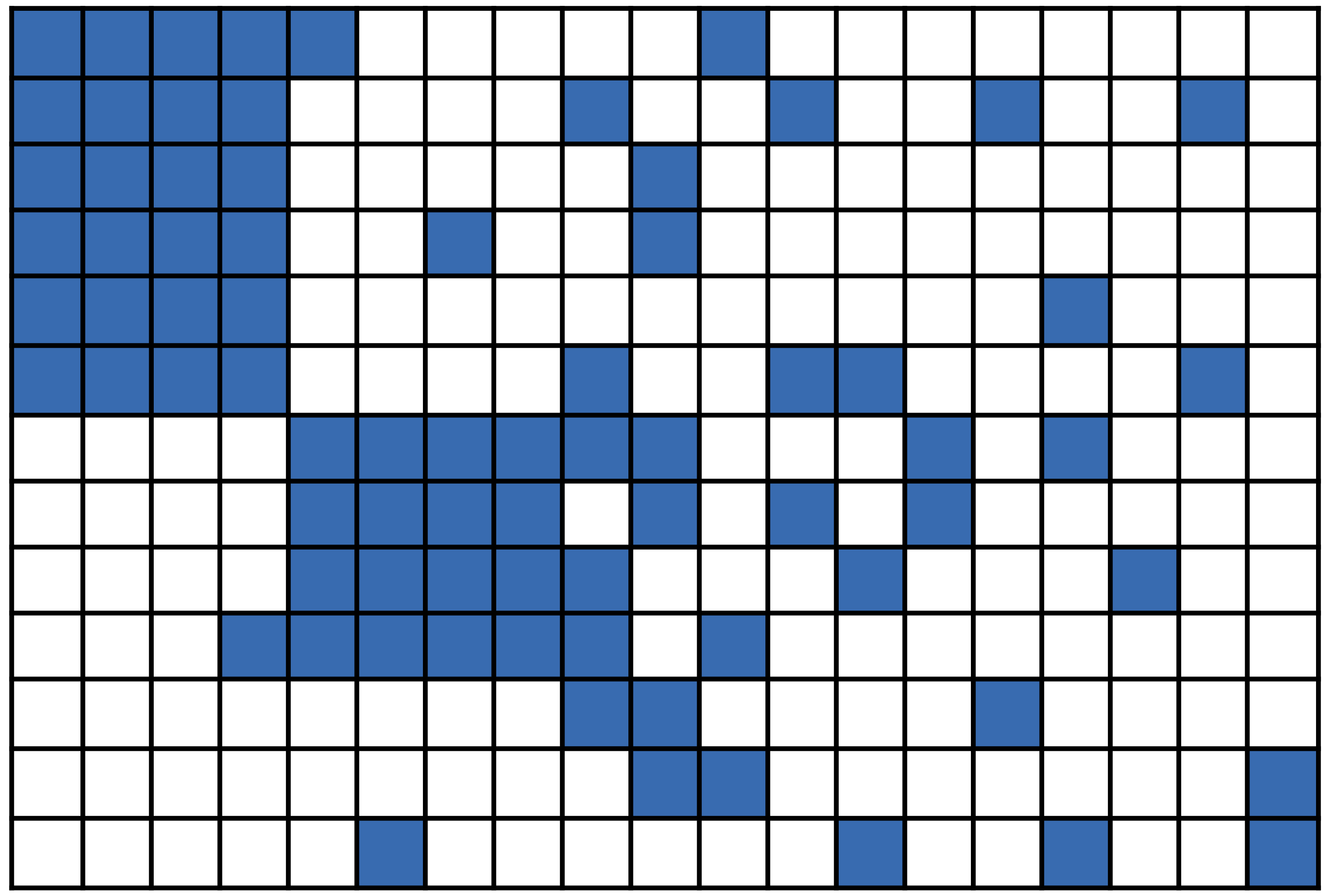


$\mathrm{PS}(U)$



J-linkage clustering [Toldo and Fusiello, ECCV 08]





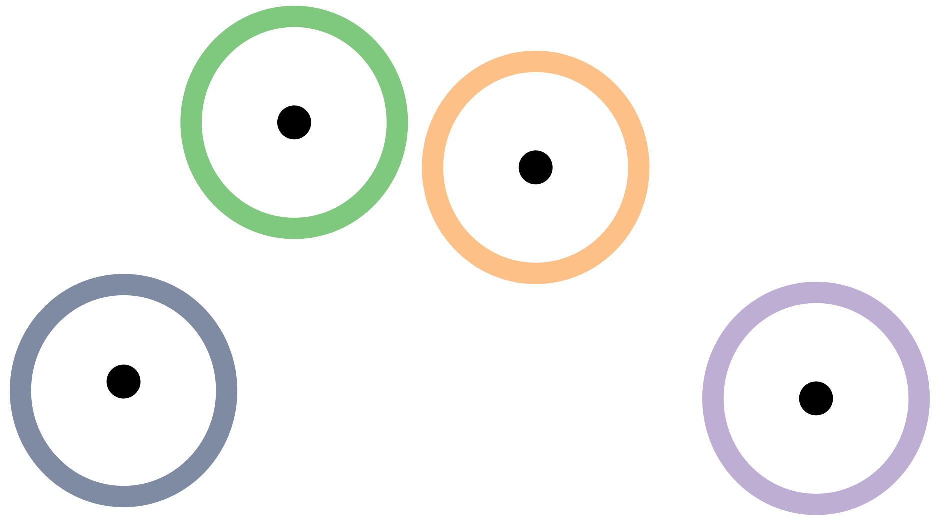
Preference matrix

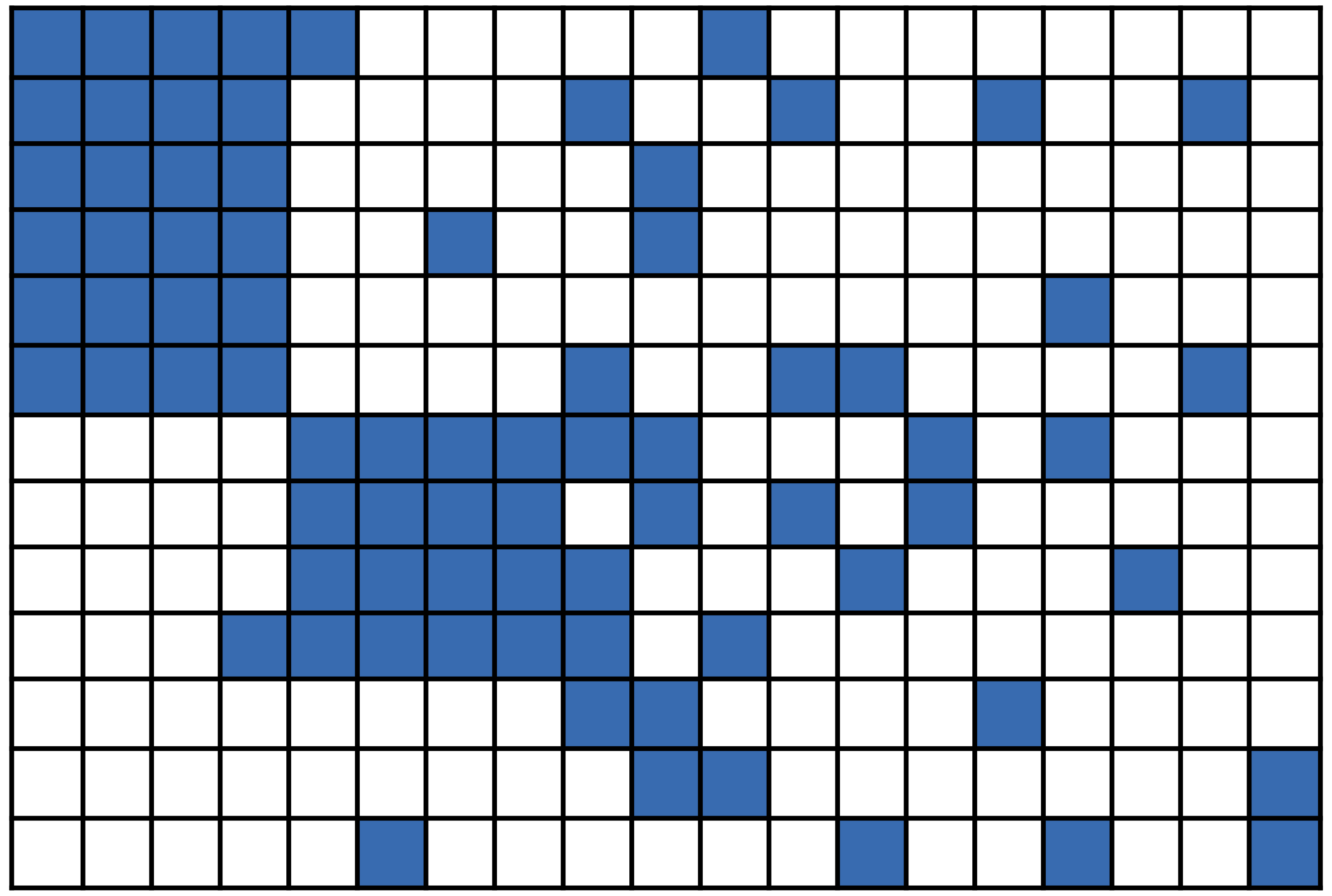
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Put each point in its own cluster





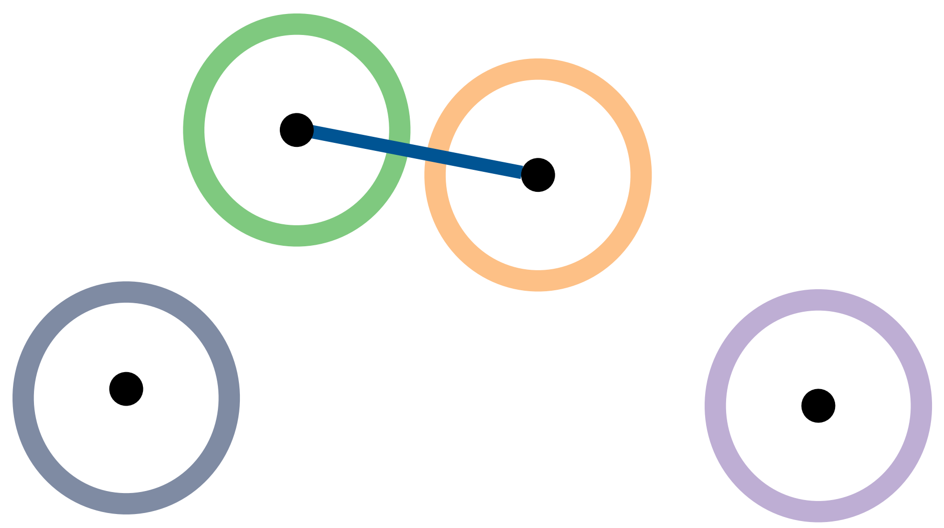
Preference matrix

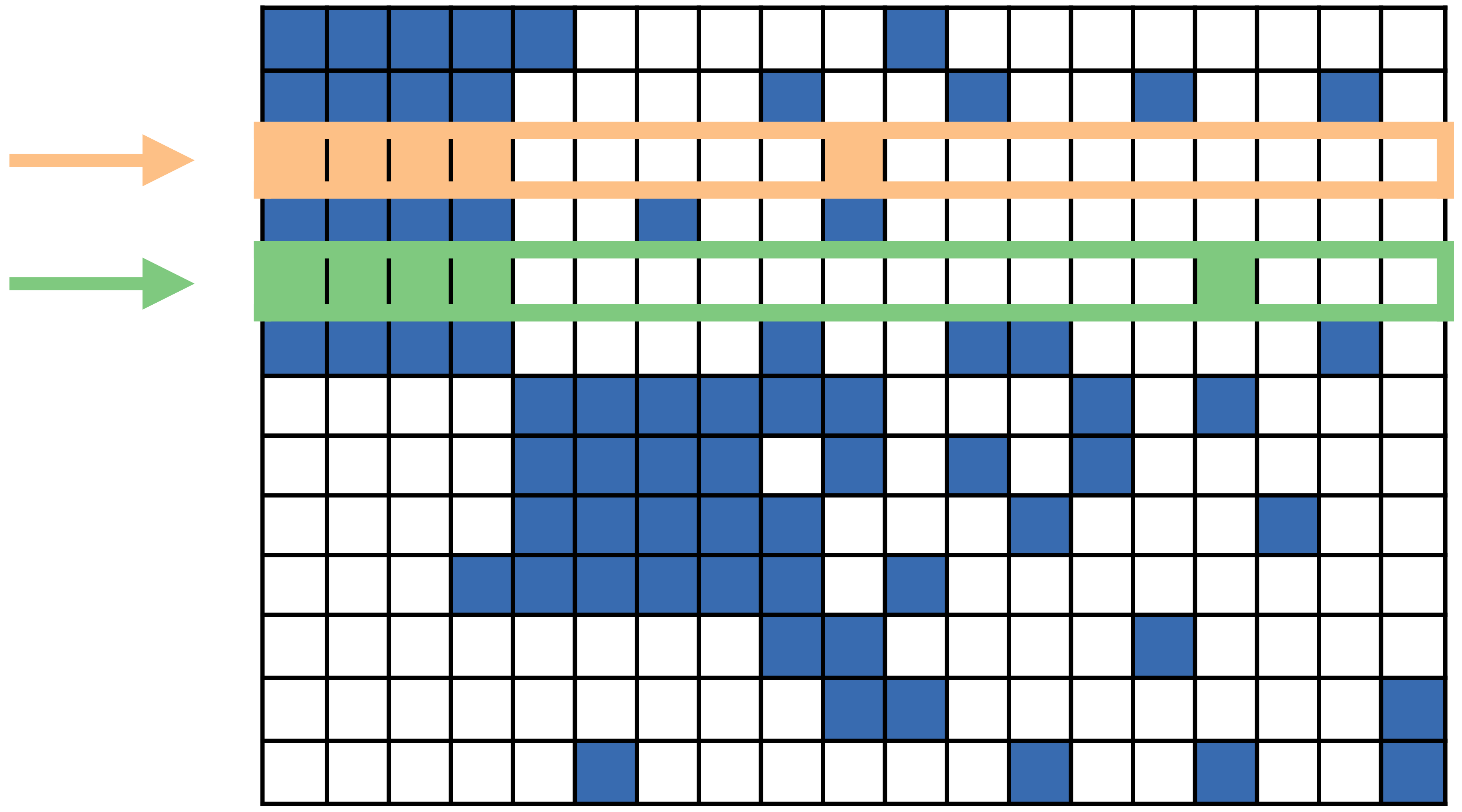
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
  Find pair (C_i, C_j) of clusters with the min d_J;
  Replace the clusters with their union;
  Compute the PS of $C_i \cup C_j$;
  Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Find the closest points in Preference Space





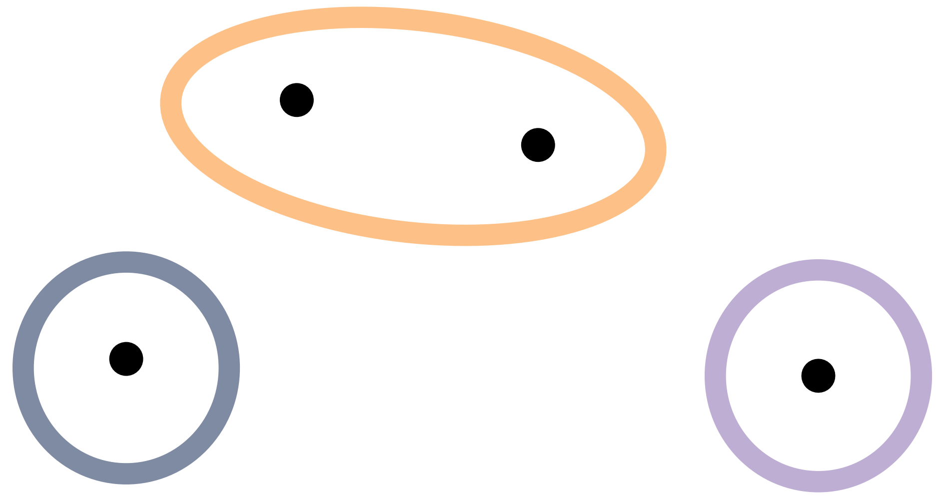
Preference matrix

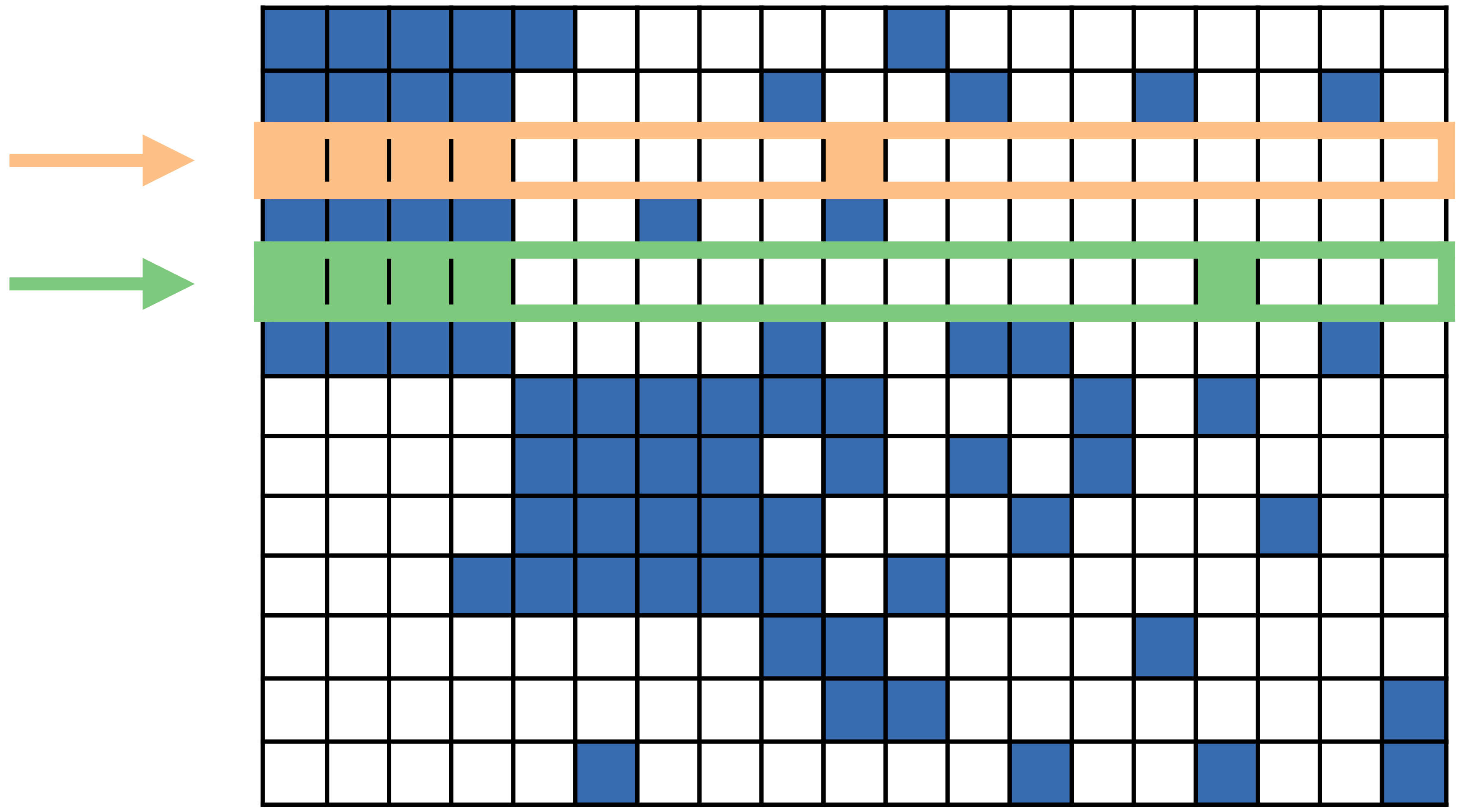
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
  Find pair (C_i, C_j) of clusters with the min d_J;
  Replace the clusters with their union;
  Compute the PS of $C_i \cup C_j$;
  Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Merge cluster



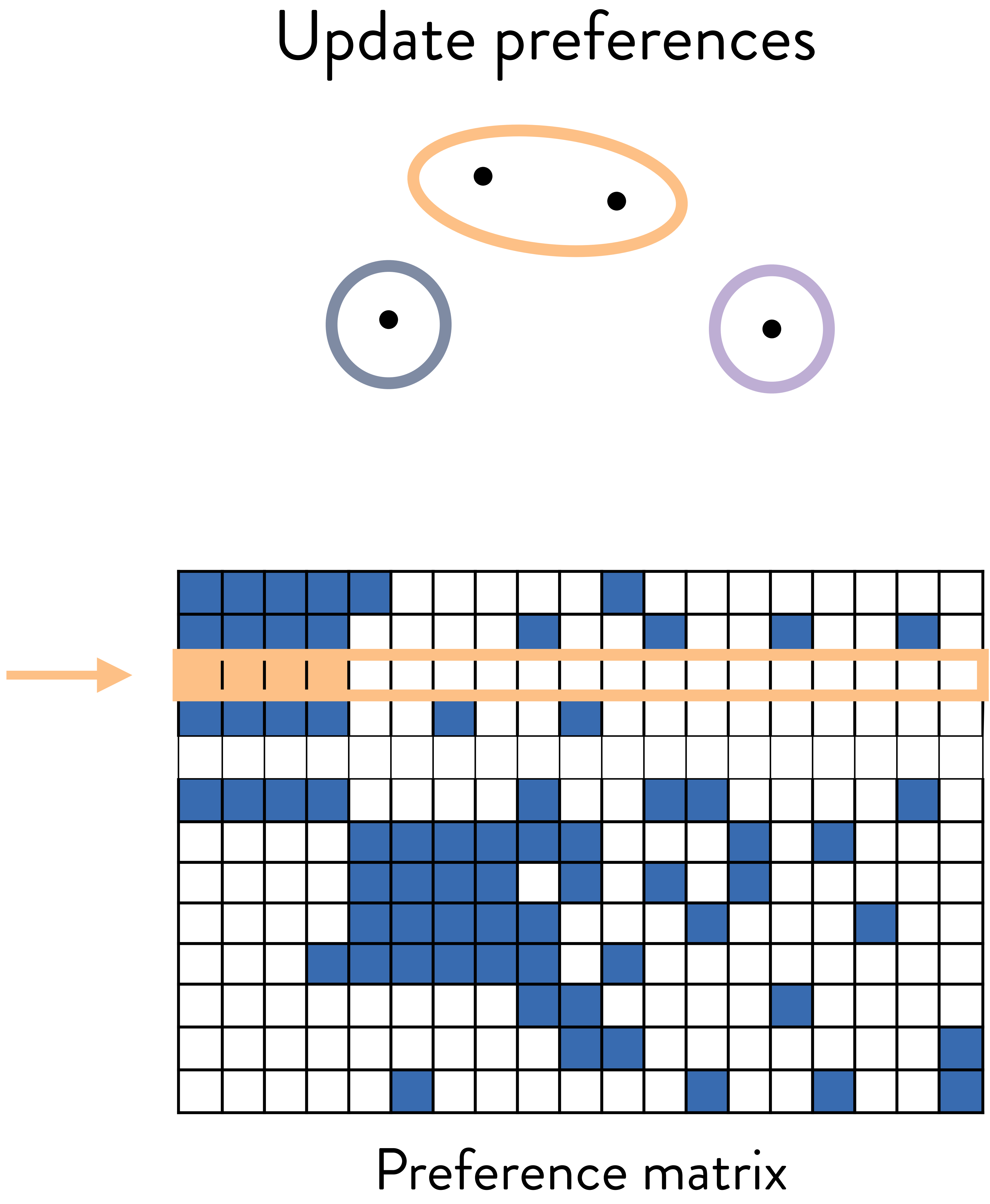


Preference matrix

Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
  Find pair (C_i, C_j) of clusters with the min d_J;
  Replace the clusters with their union;
  Compute the PS of $C_i \cup C_j$;
  Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

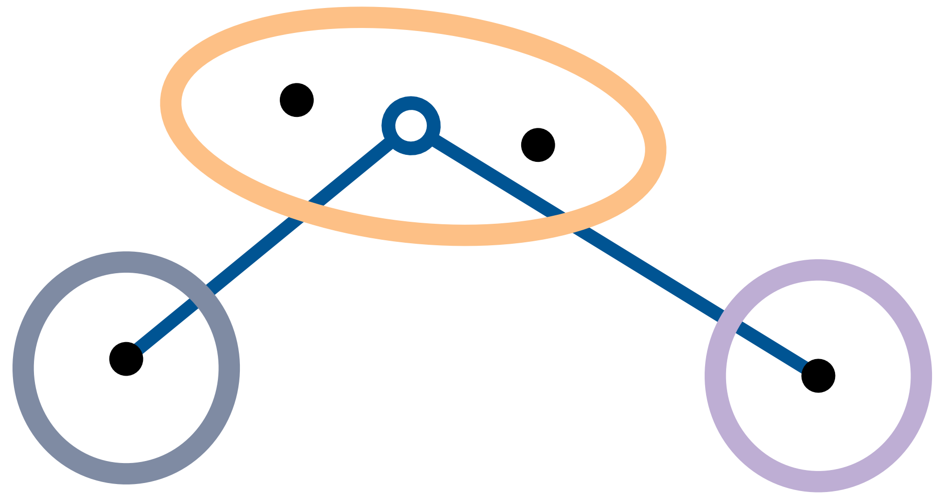


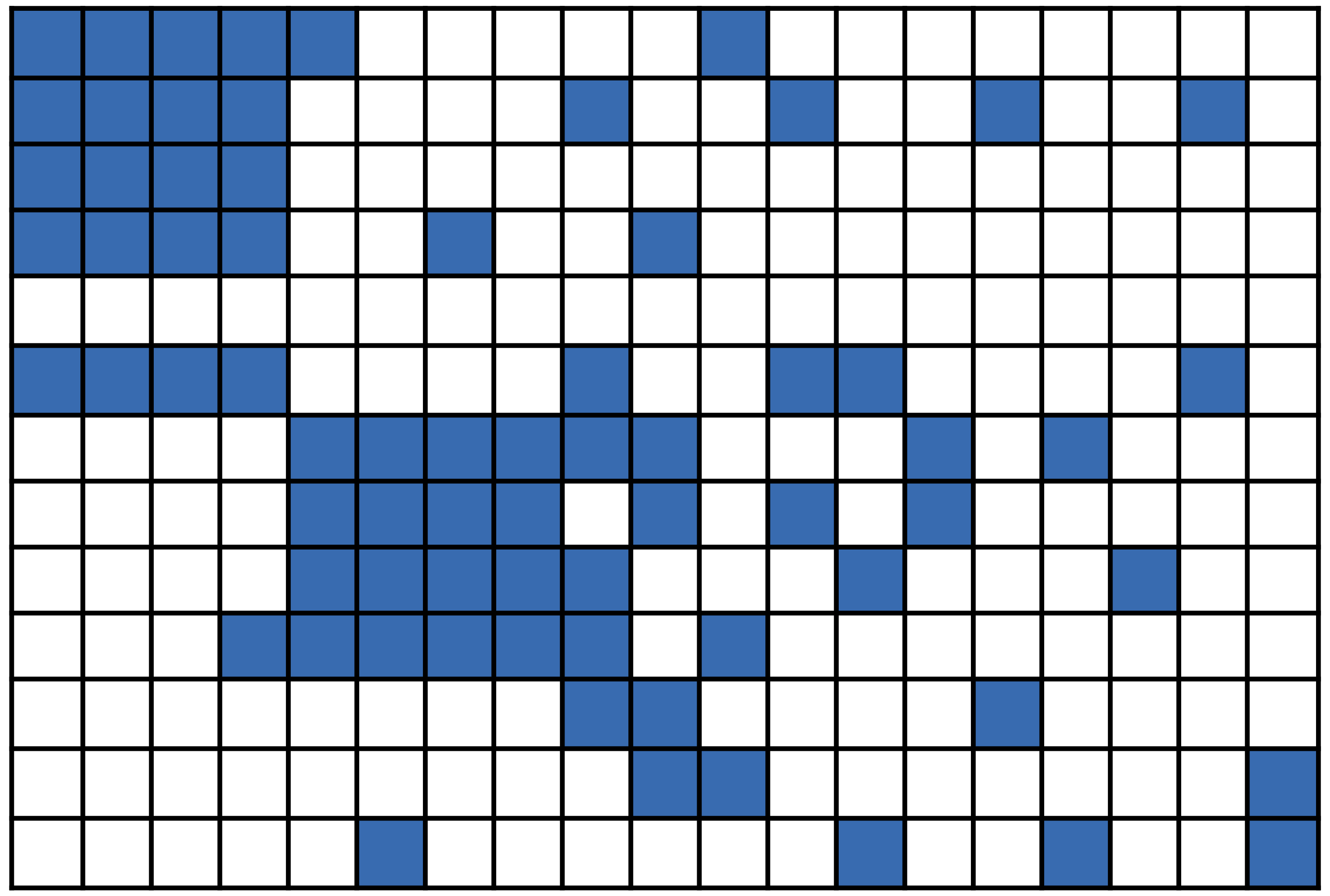
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
  Find pair (C_i, C_j) of clusters with the min d_J;
  Replace the clusters with their union;
  Compute the PS of $C_i \cup C_j$;
  Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Update distances





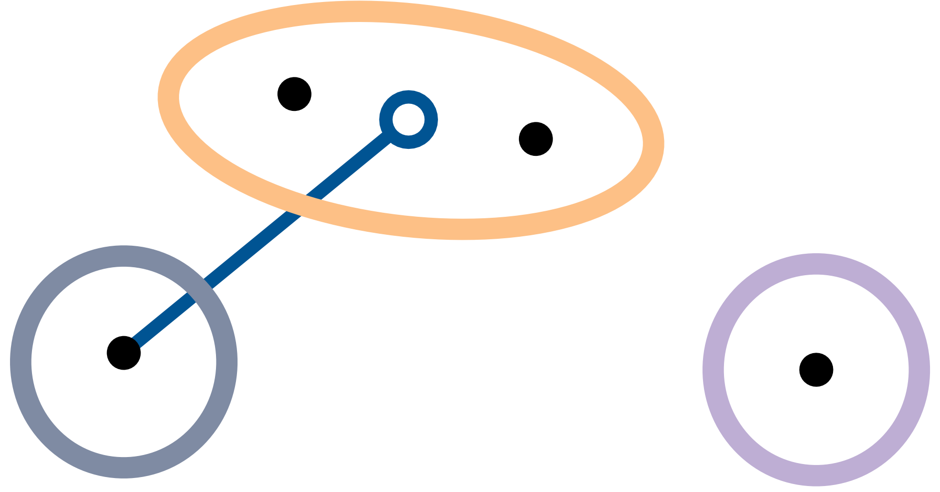
Preference matrix

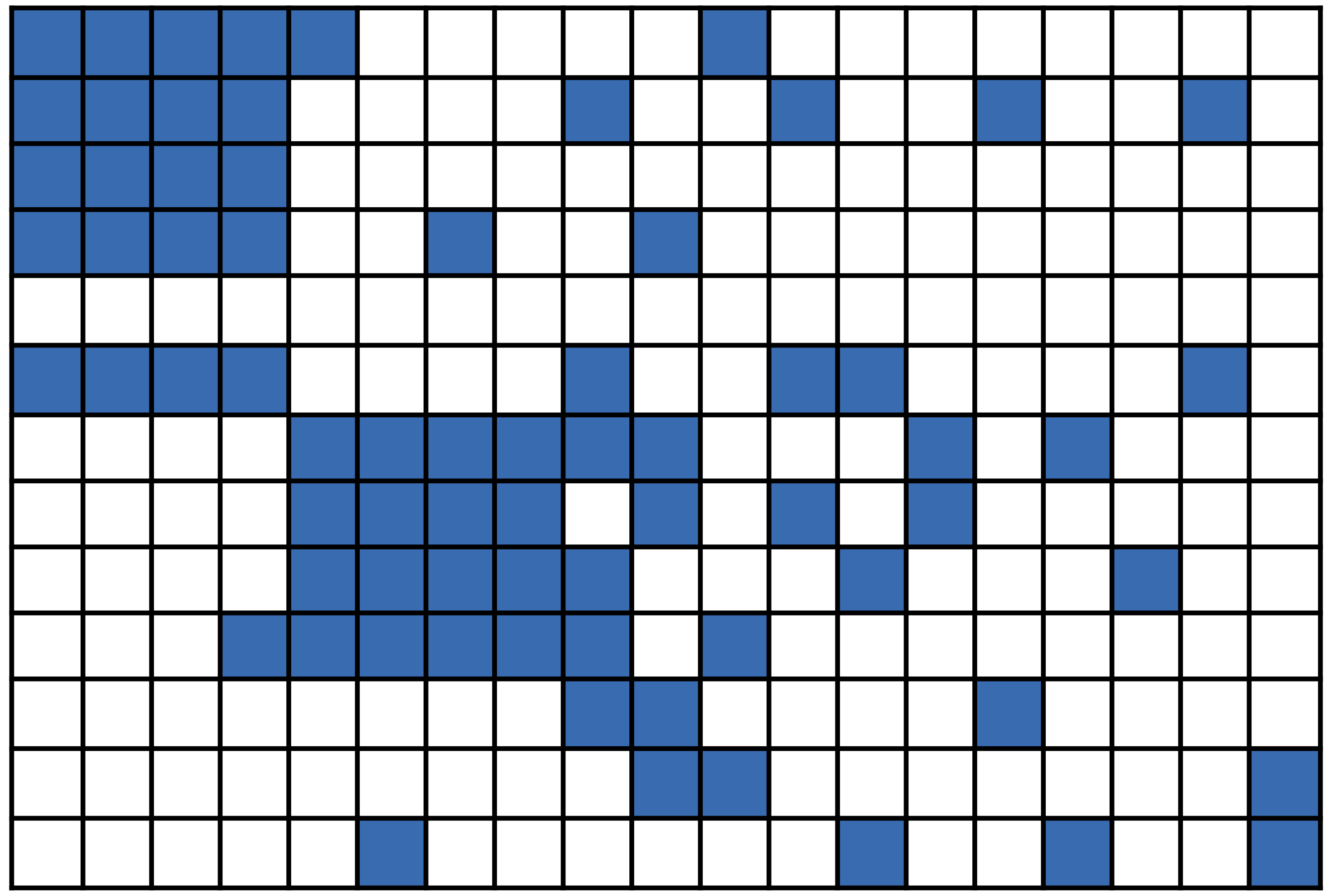
Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]

Continue until all PS are disjoint…



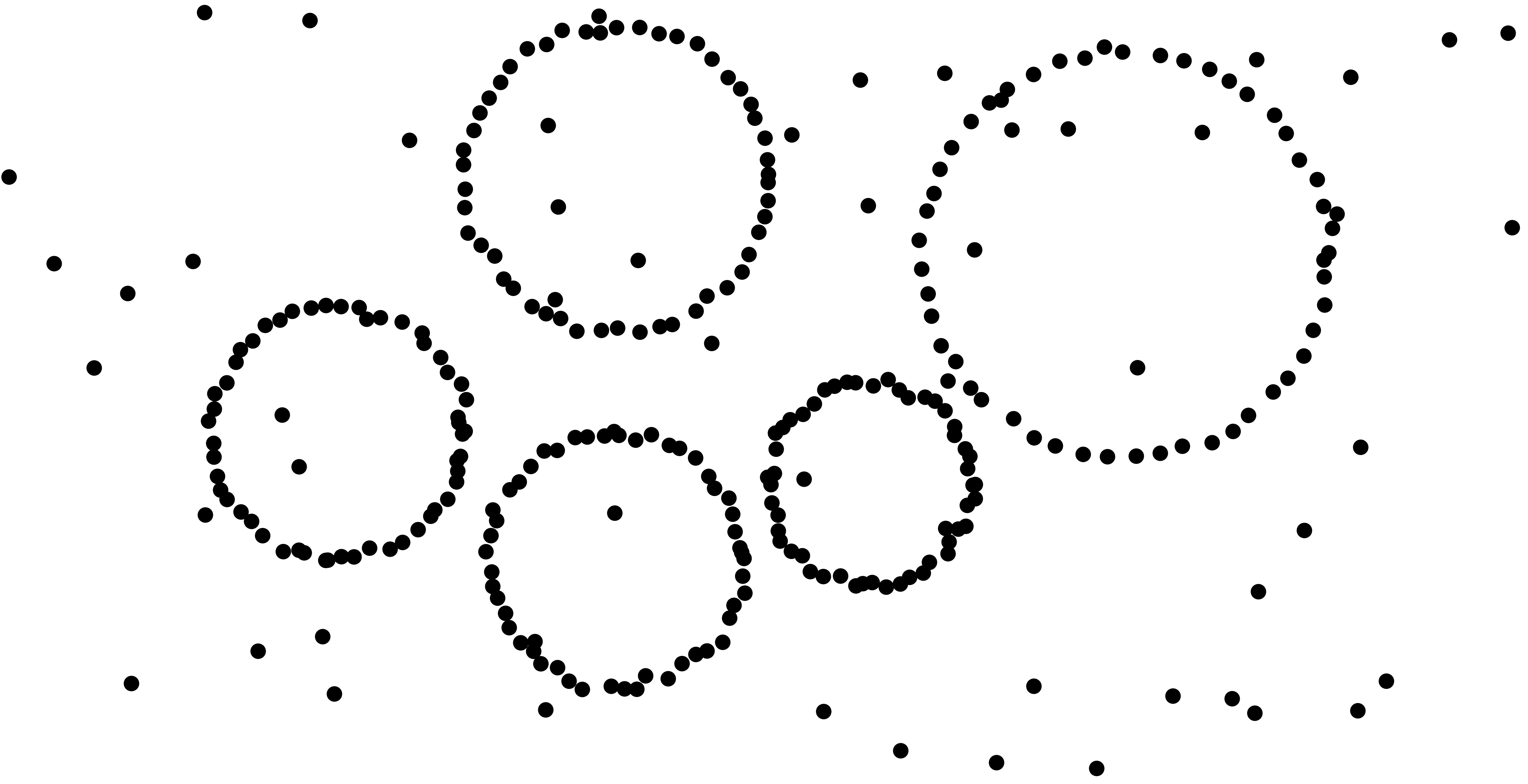


Preference matrix

Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

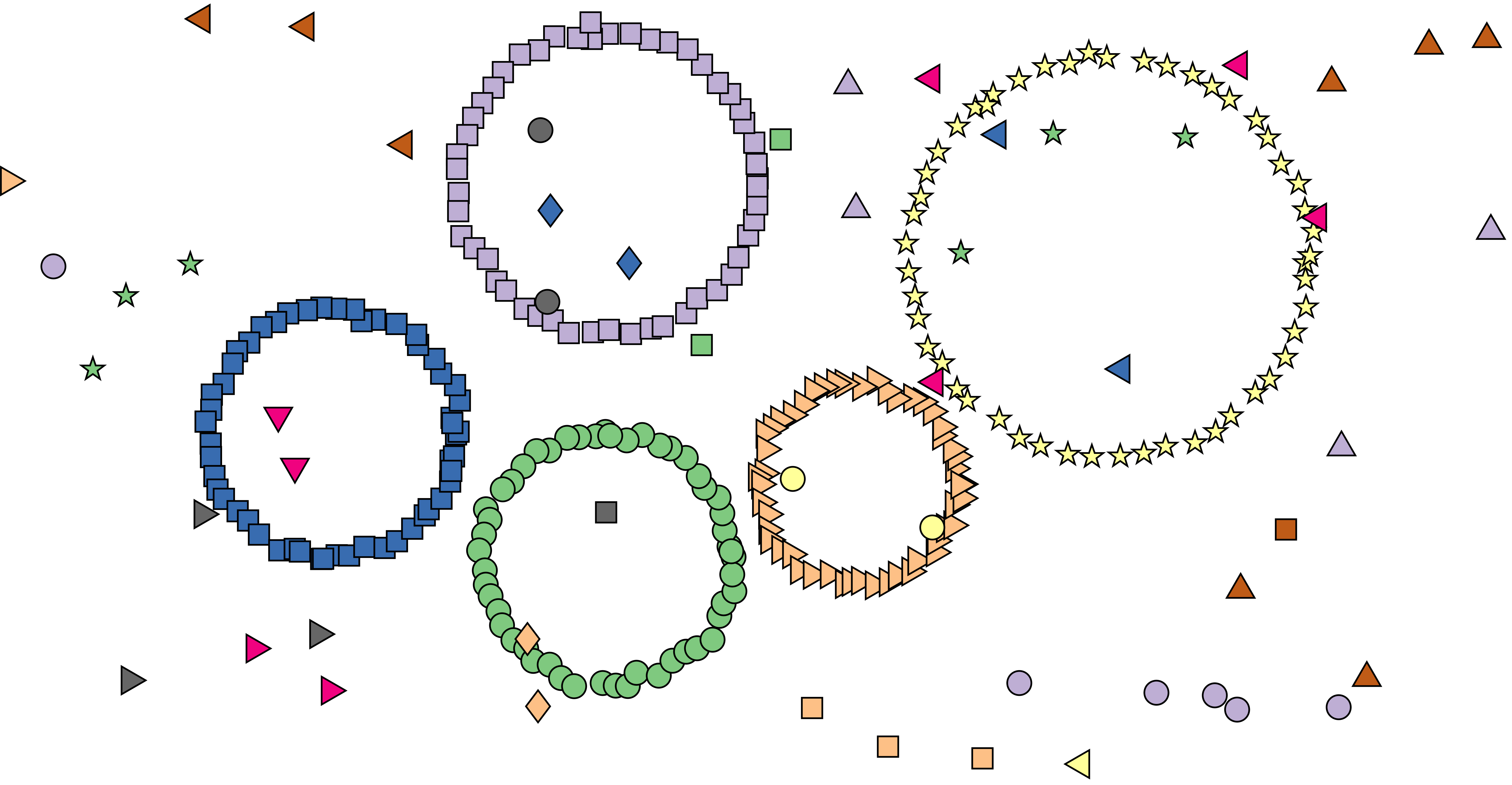
J-linkage clustering [Toldo and Fusiello, ECCV 08]



Input: X data, ϵ inlier threshold
Output: Partition in structures and models

Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

J-linkage clustering [Toldo and Fusiello, ECCV 08]



Input: X data, ϵ inlier threshold
Output: Partition in structures and models

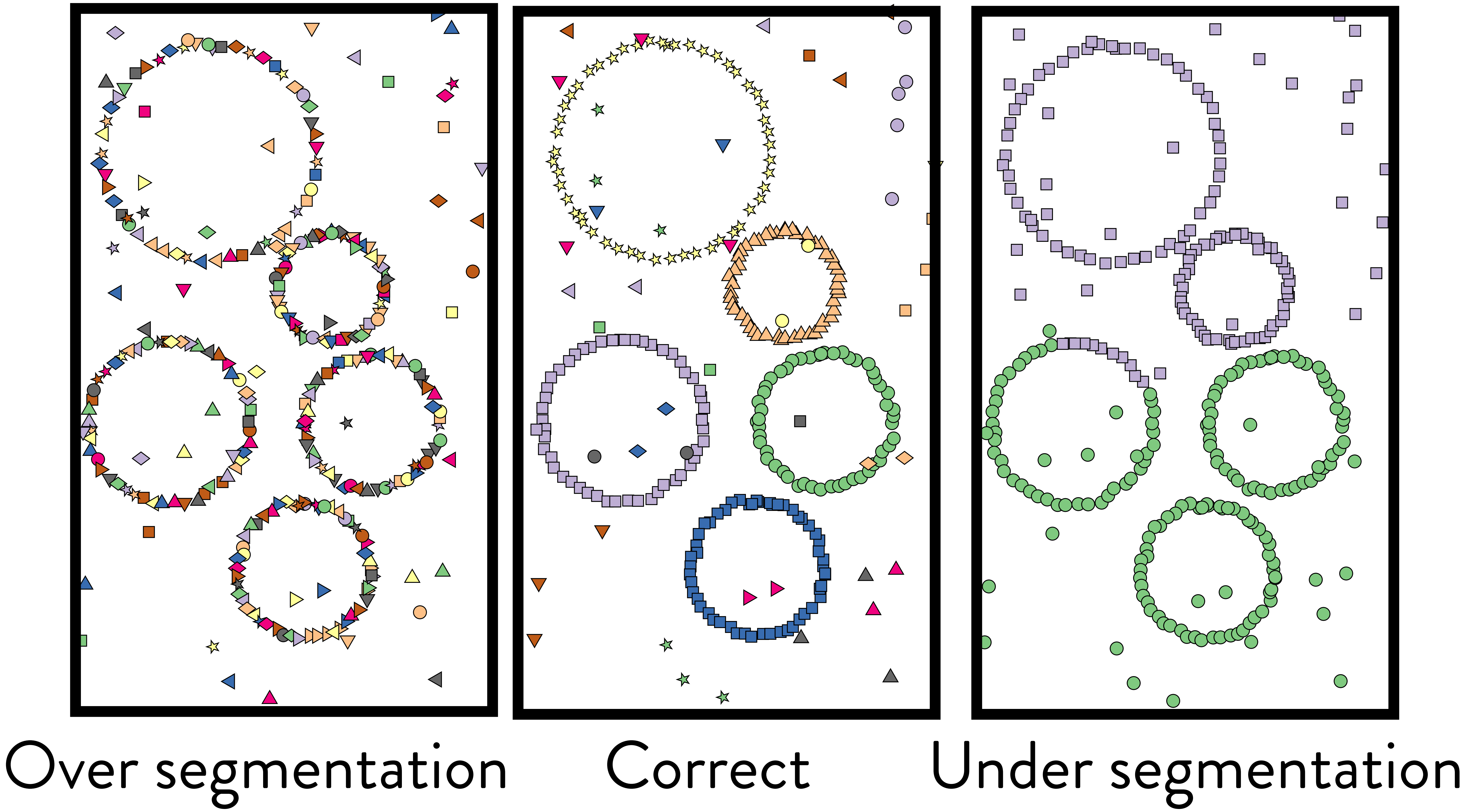
Randomly sample model hypotheses $H \subset \Theta$;
Compute PS;
Put each point in its own cluster $C_i = \{x_i\}$;
Compute d_J Jaccard distance between PS;
while $\min(d_J) < 1$ **do**
    Find pair (C_i, C_j) of clusters with the min d_J;
    Replace the clusters with their union;
    Compute the PS of $C_i \cup C_j$;
    Update d_J;
end
Local fit of models to clusters;

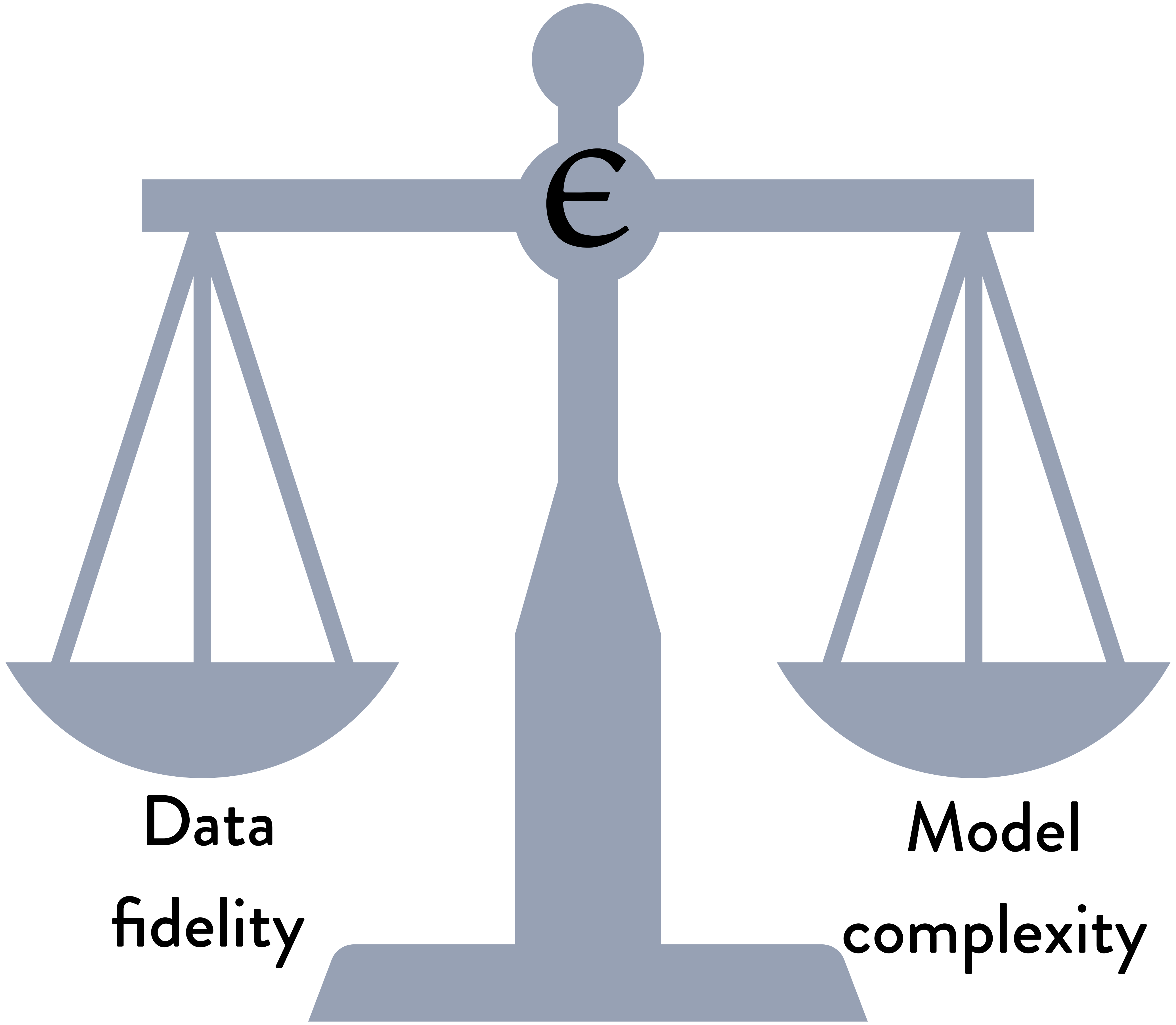
J-linkage clustering [Toldo and Fusiello, ECCV 08]

- The number of structures is automatically determined
- For each cluster there exists at least one model that fits all the points of the cluster
- Clusters are "maximal" in the sense that does not exist a model that explain all the points of two distinct clusters

The role of the inlier threshold

1. explicitly dichotomizes between outliers and inliers
2. implicitly controls disjointness between PS and the final number of structures.





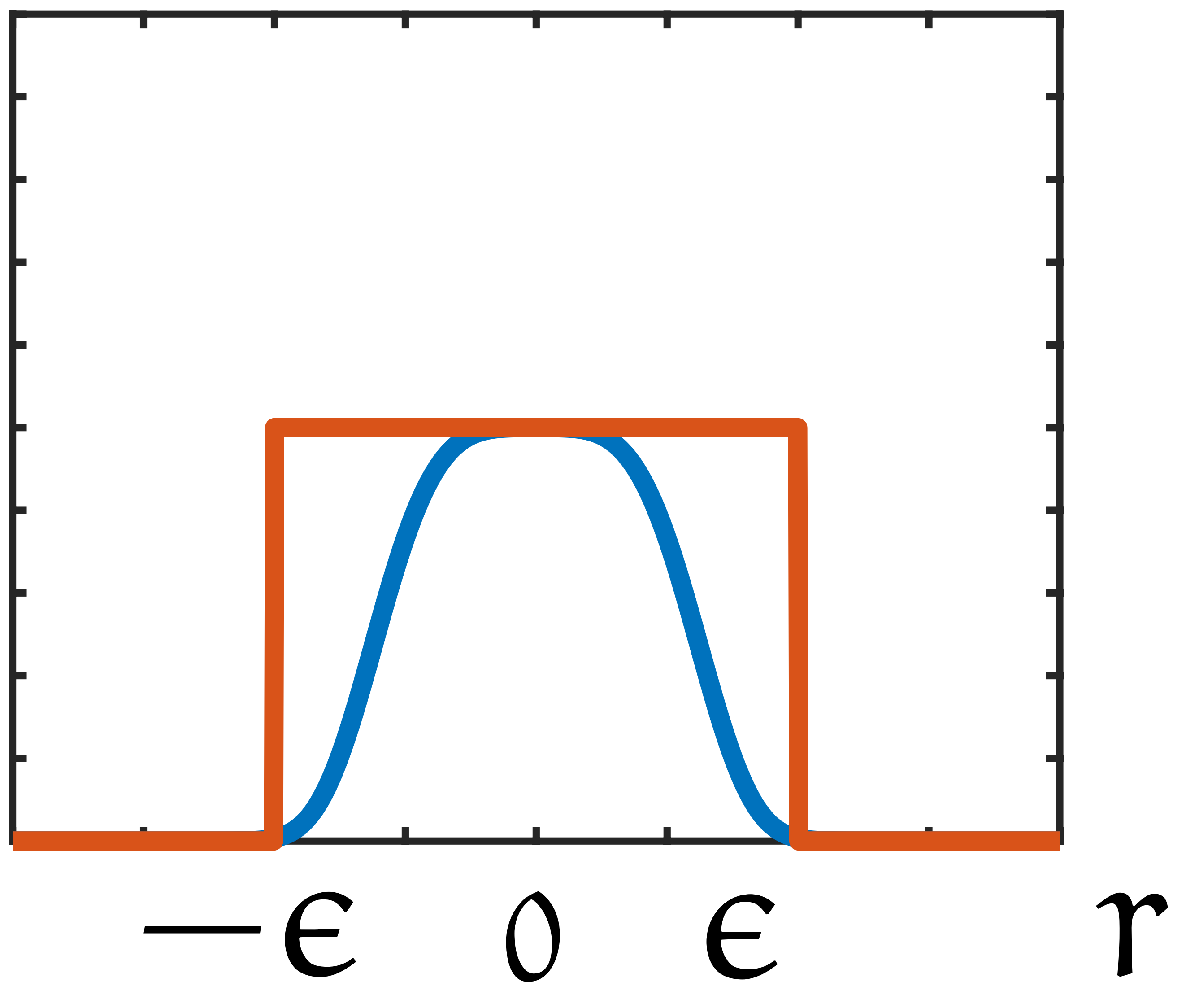
T-linkage relaxation [Magri and Fusiello CVPR 14]

Continuous relaxation of PS.
Points are represented as soft Preference Function (PF) w.r.t. the pool of models.

Representation

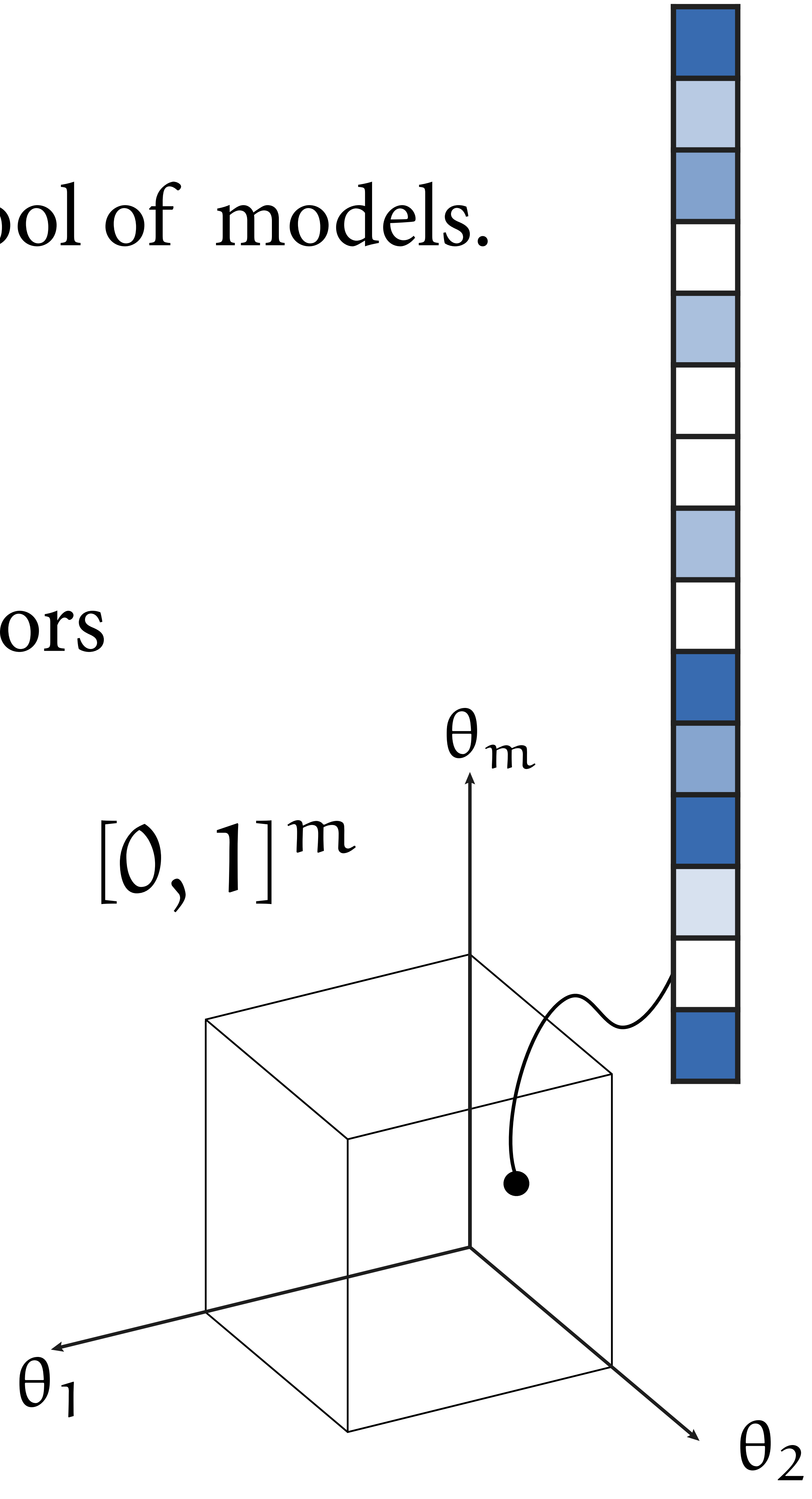
$$PF(x) = [g(r(x,\theta_1)), \dots, g(r(x,\theta_m))]$$

Instead of binary votes, preferences are expressed using M-estimators



$$\tilde{f}(r) = \begin{cases} 1 & r \leq \epsilon \\ 0 & r > \epsilon \end{cases}$$

$$g(r) = \begin{cases} \exp(-\frac{r}{\sigma(\epsilon)}^2) & r \leq \epsilon \\ 0 & r > \epsilon \end{cases}$$



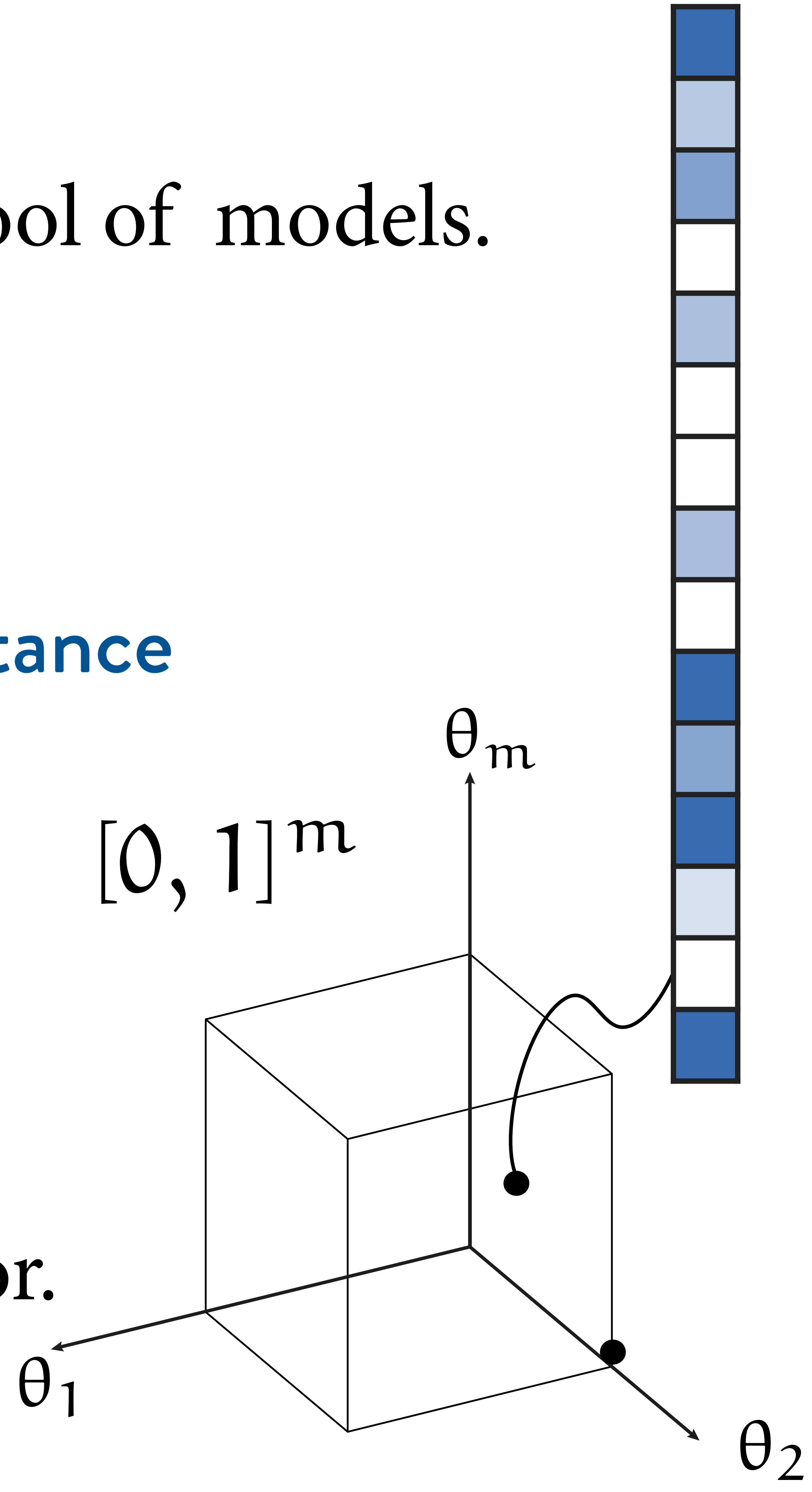
T-linkage relaxation [Magri and Fusiello CVPR 14]

Continuous relaxation of PS.
Points are represented as soft Preference Function (PF) w.r.t. the pool of models.

Distance

The Jaccard distance between PS is generalised by the **Tanimoto distance** between PF

$$d_T(p, q) = 1 - \frac{\langle p, q \rangle}{\|p\|^2 + \|q\|^2 - \langle p, q \rangle}$$



Tanimoto distance specialised to Jaccard in the case of binary vector.
Disjointness is replaced by orthogonality.

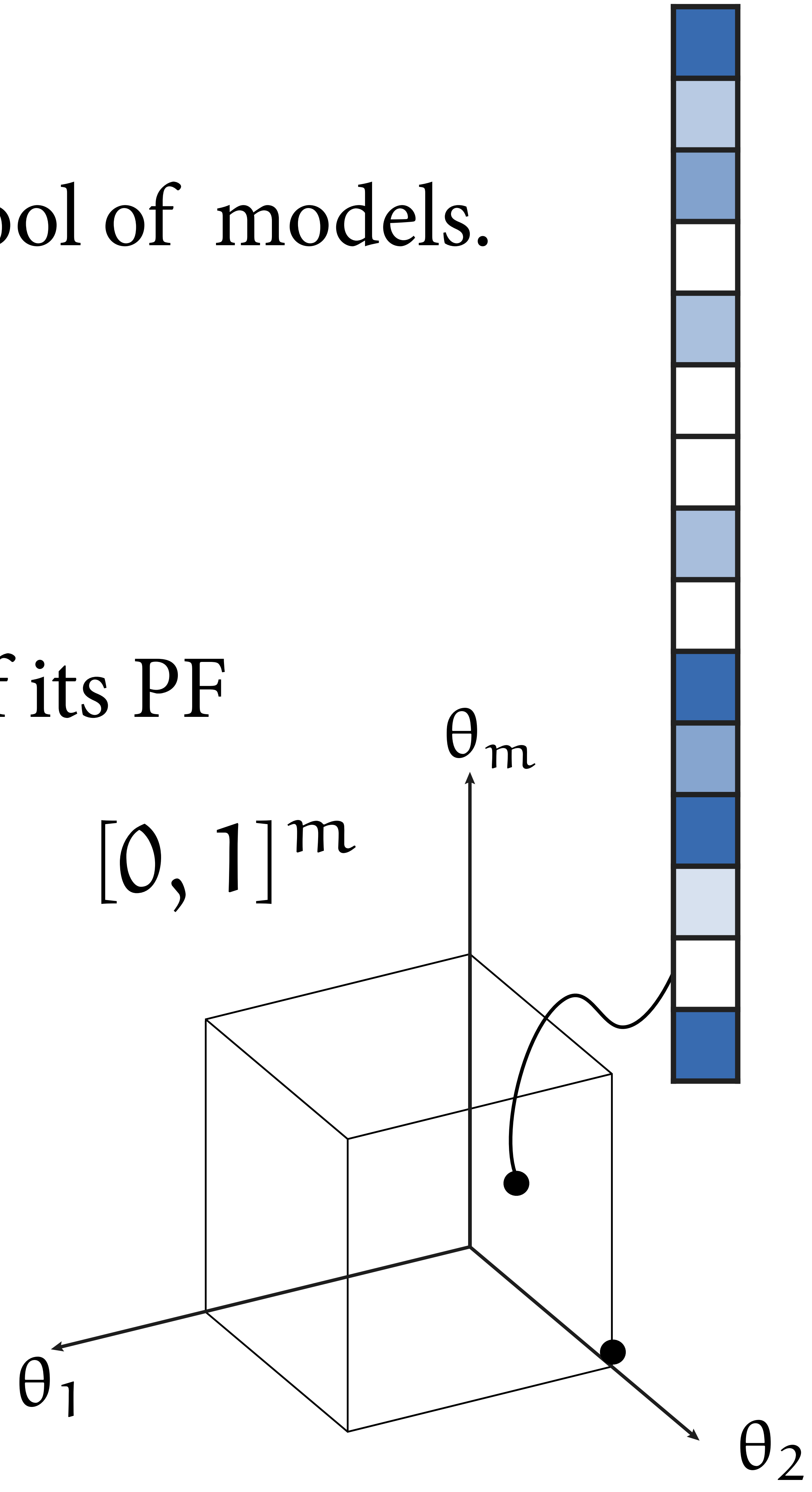
T-linkage relaxation [Magri and Fusiello CVPR 14]

Continuous relaxation of PS.
Points are represented as soft Preference Function (PF) w.r.t. the pool of models.

Cluster representative

Cluster representative is given by the component wise minimum of its PF

$$[PF(U)]_i = \min_{x \in U}([PF(x)]_i)$$



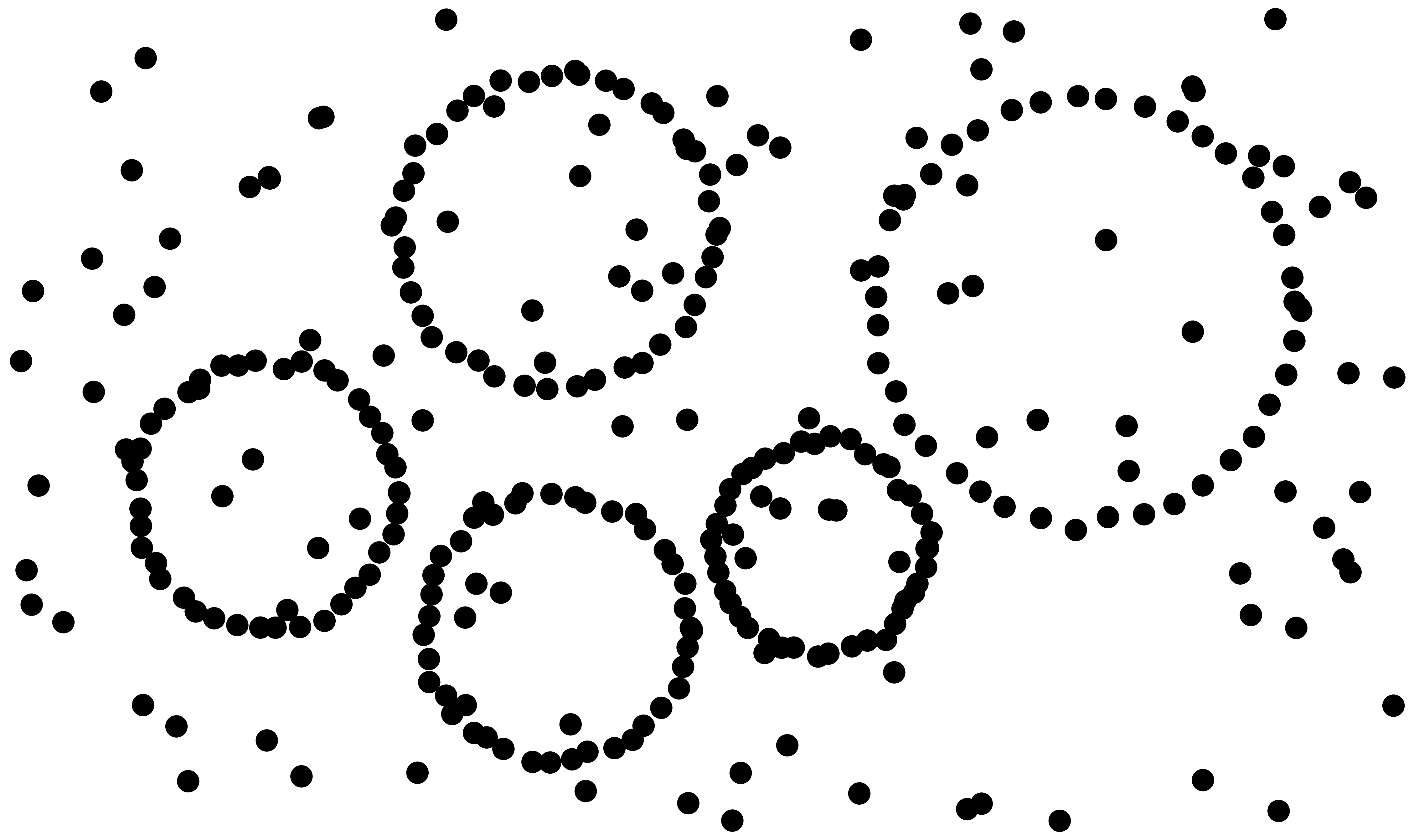
T-linkage relaxation [Magri and Fusiello CVPR 14]

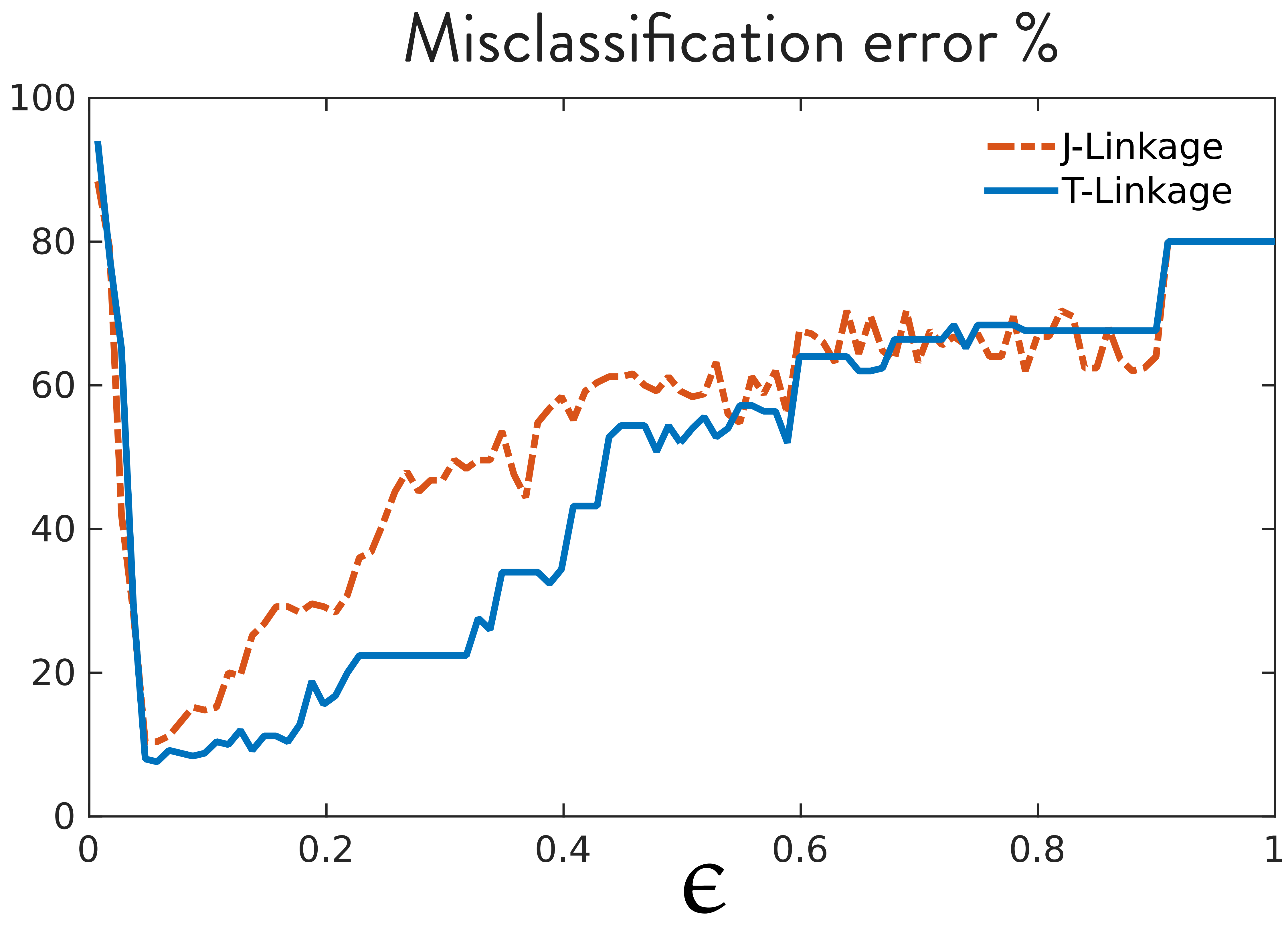
In practice the Preference matrix is no longer binary, but stores soft preferences.

	J-linkage	T-linkage
Representation	$\mathrm{PS} \in \{0, 1\}^m$	$\mathrm{PF} \in [0, 1]^m$
Distance	Jaccard	Tanimoto
Cluster repr.	$\bigcap \mathrm{PS}$	$\min \mathrm{PF}$

T-Linkage results are more accurate and more robust.

T-linkage relaxation [Magri and Fusiello CVPR 14]





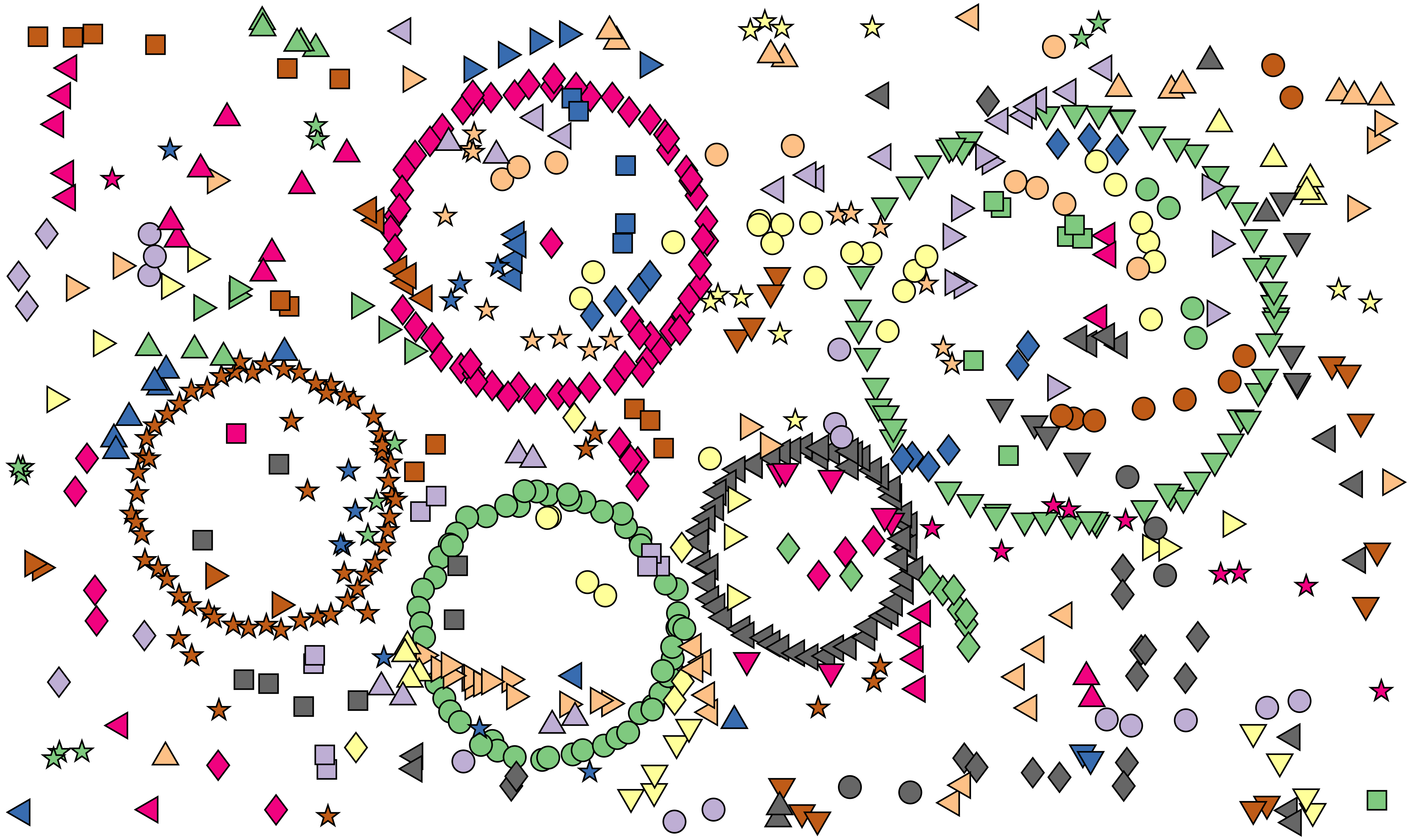
Results are more accurate and more robust.

Dealing with outliers [Magri and Fusiello CVPR 14]

Gestalt theory - Helmholtz principle:

an observed geometric structure is perceptually "meaningful" if its number of occurrences would be very small in a random situation.

Idea:
use statistical validation to prune out structures that are likely to be mere coincidences (bigger structure do not happen by chance)



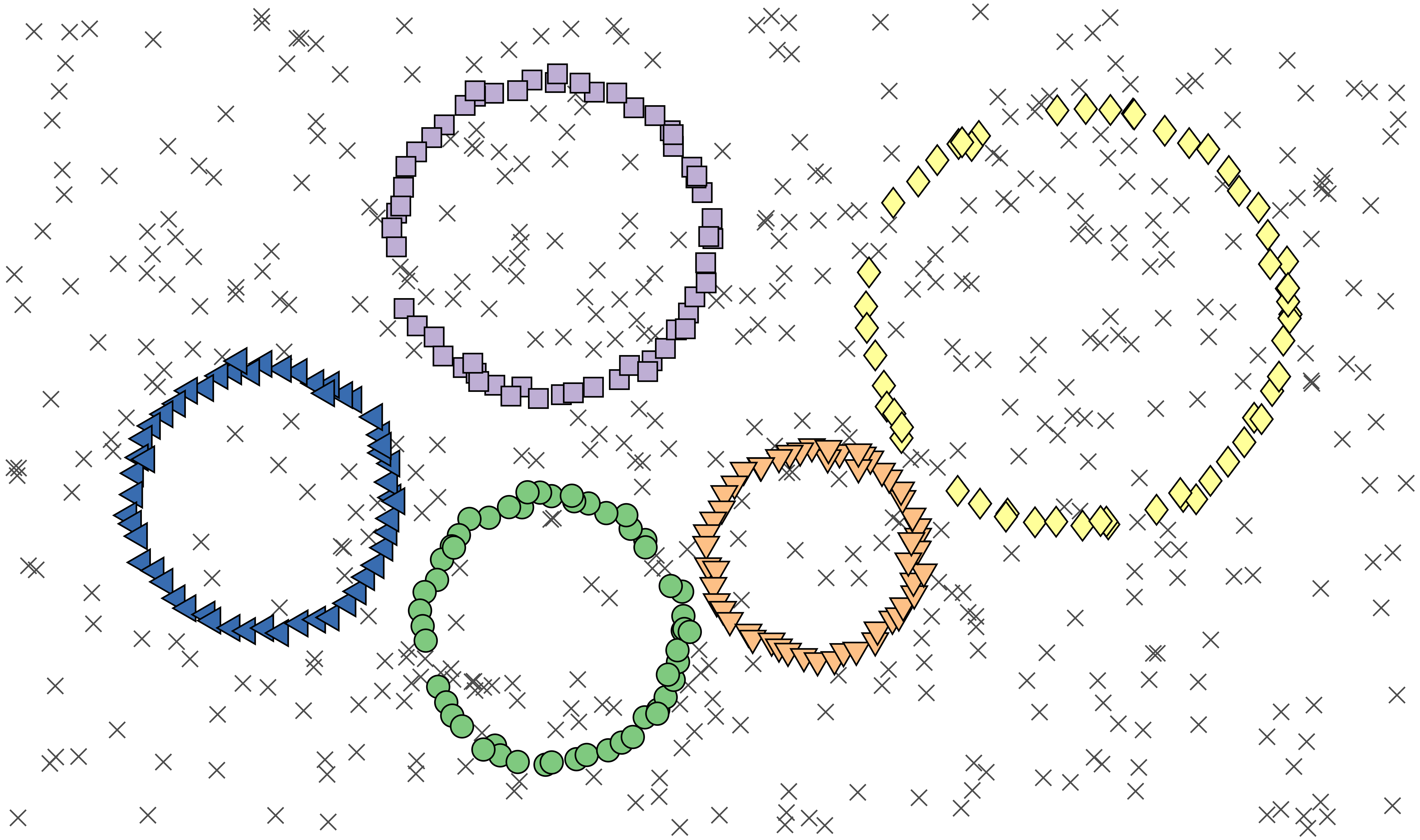
Dealing with outliers [Magri and Fusiello CVPR 14]

Gestalt theory - Helmholtz principle:

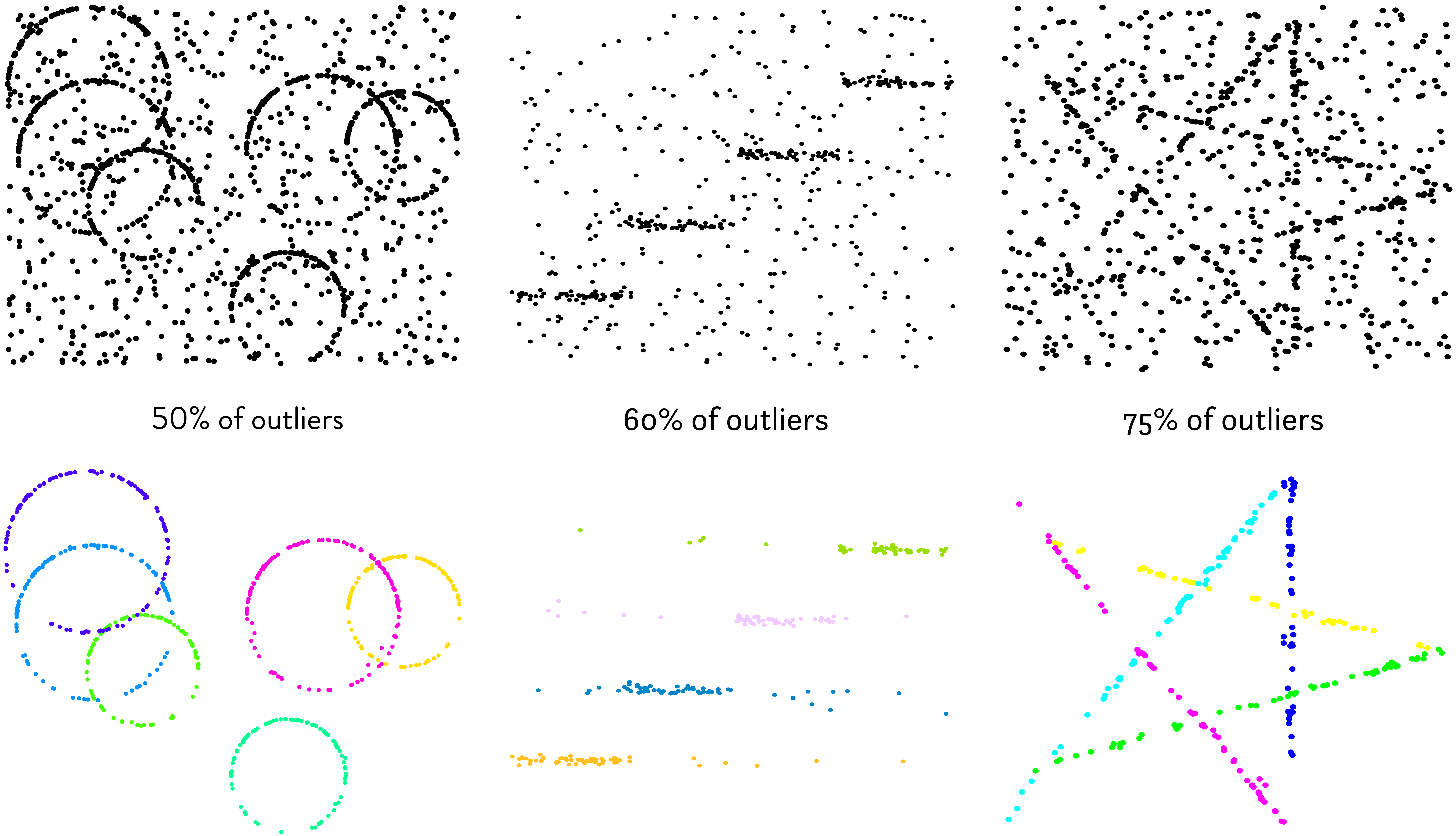
an observed geometric structure is perceptually "meaningful" if its number of occurrences would be very small in a random situation.

Idea:

use statistical validation to prune out structures that are likely to be mere coincidence (bigger structure do not happen by chance)



T-linkage: sample results



Scan2bim sample results [Magri and Fusiello 3DV 18]

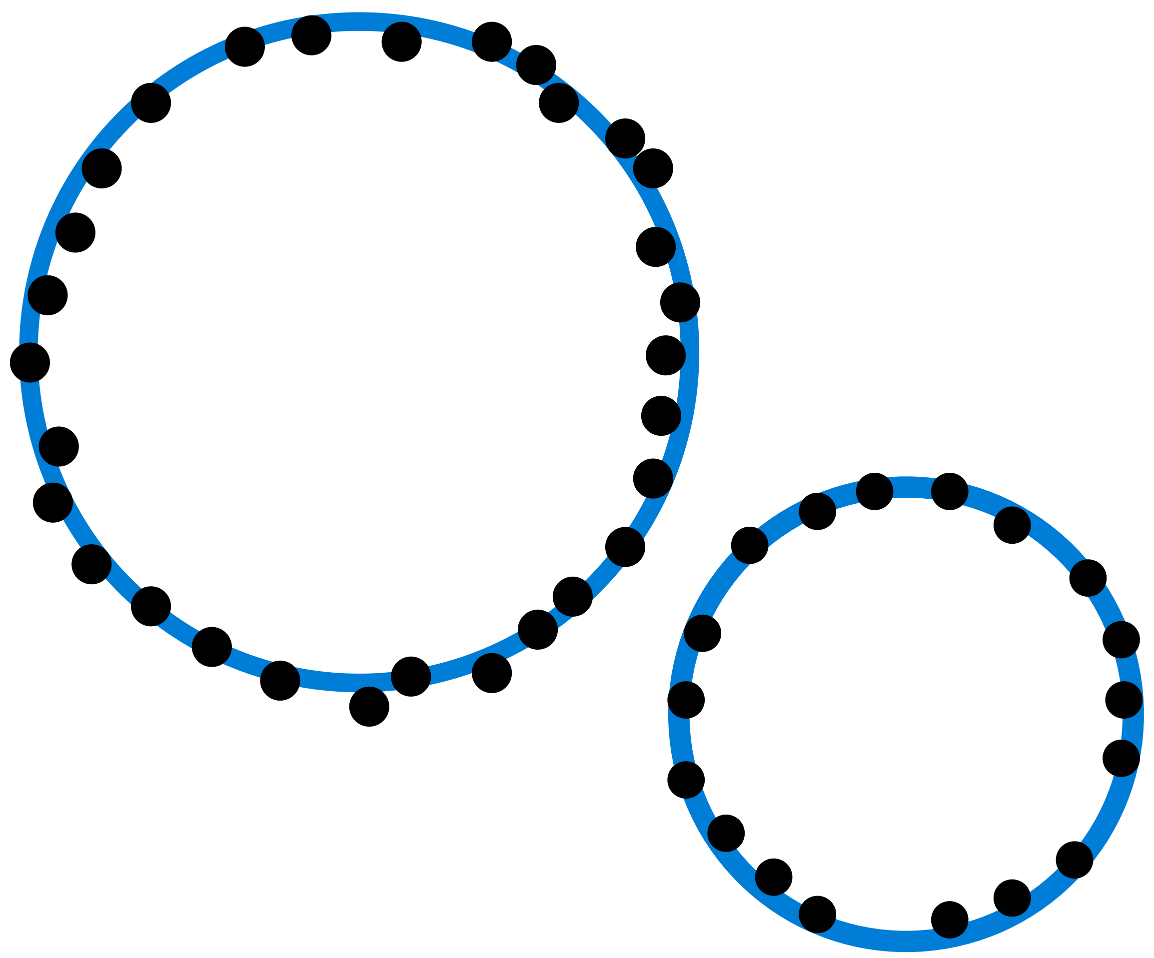
For big dataset an efficient hashing scheme can be used to approximate Jaccard distances.

This idea has been implemented in a blueprint generation tool in 3DF Zephyr:

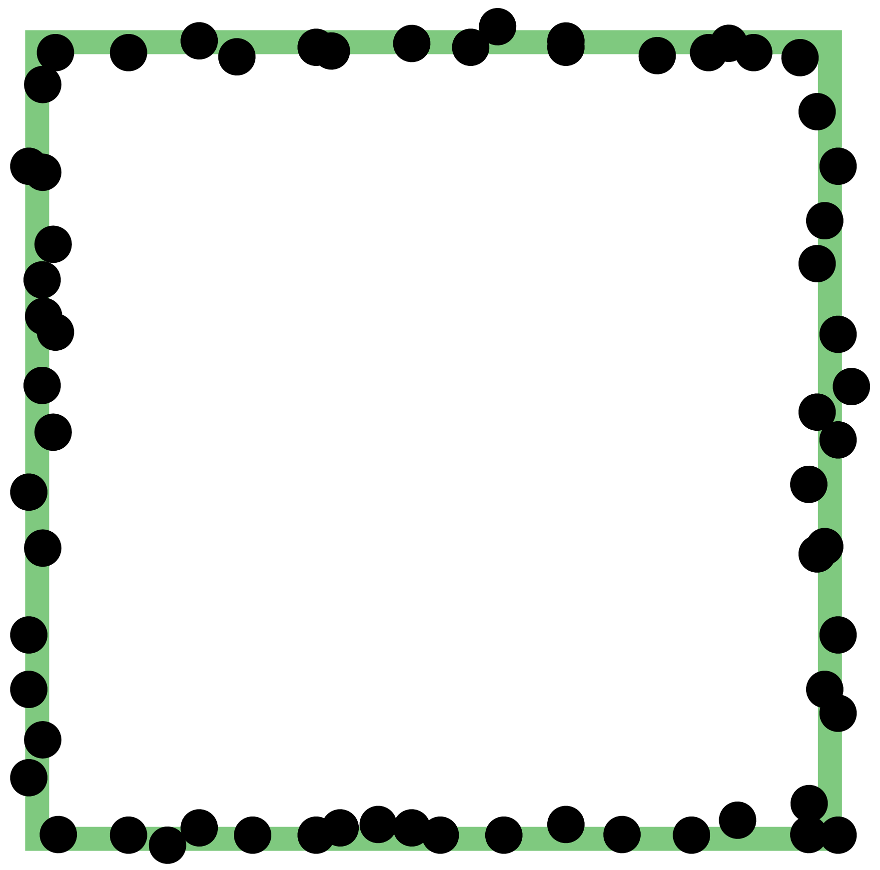
https://youtu.be/uGO2ex_GsaY?t=91

Multi class multi model fitting

Using T-linkage we are able to detect separately



or

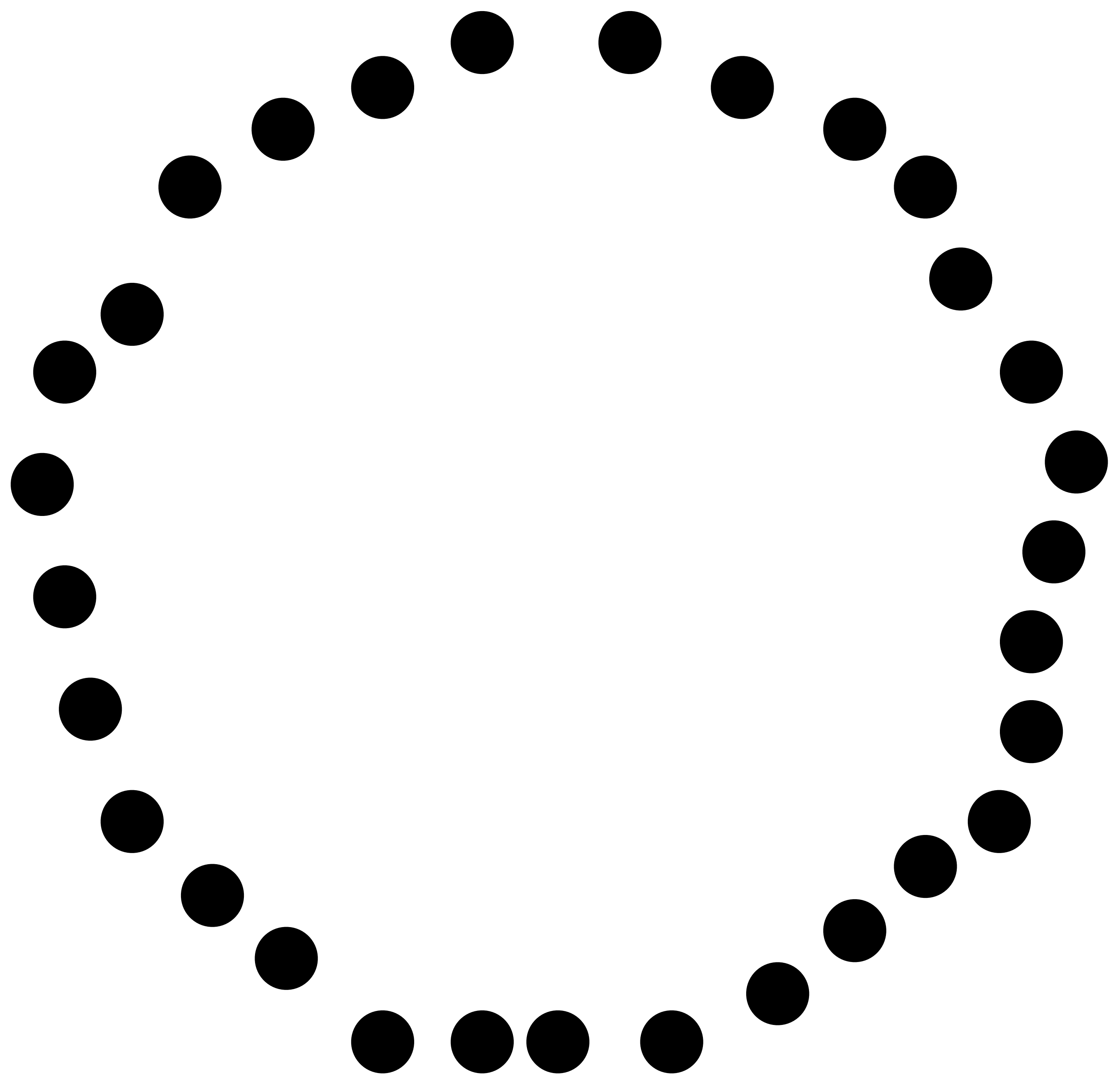


Multiple circles

Multiple lines

Multi class multi model fitting: model selection

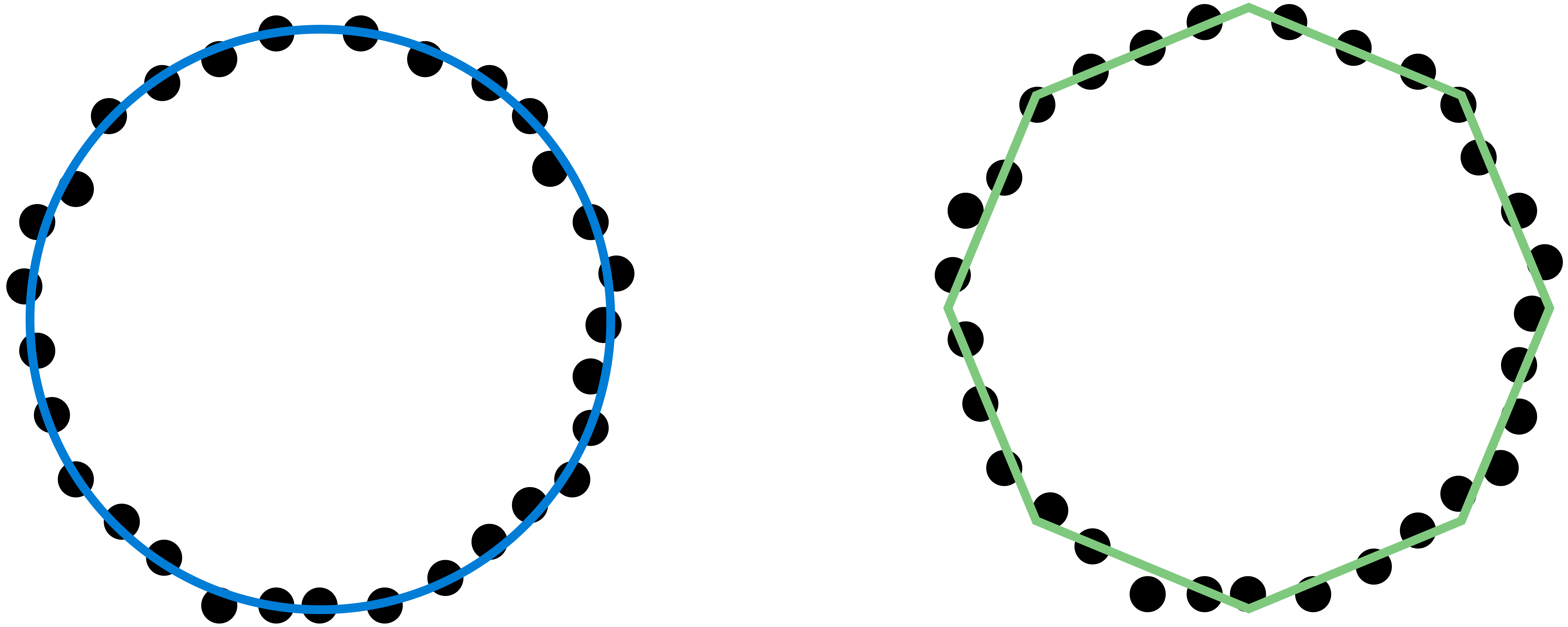
If we want to describes data with lines **&** circles,
at some point we have to face a **model selection problem**:



circle or a polygon?

Multi class multi model fitting

GRIC: geometric robust information criterion



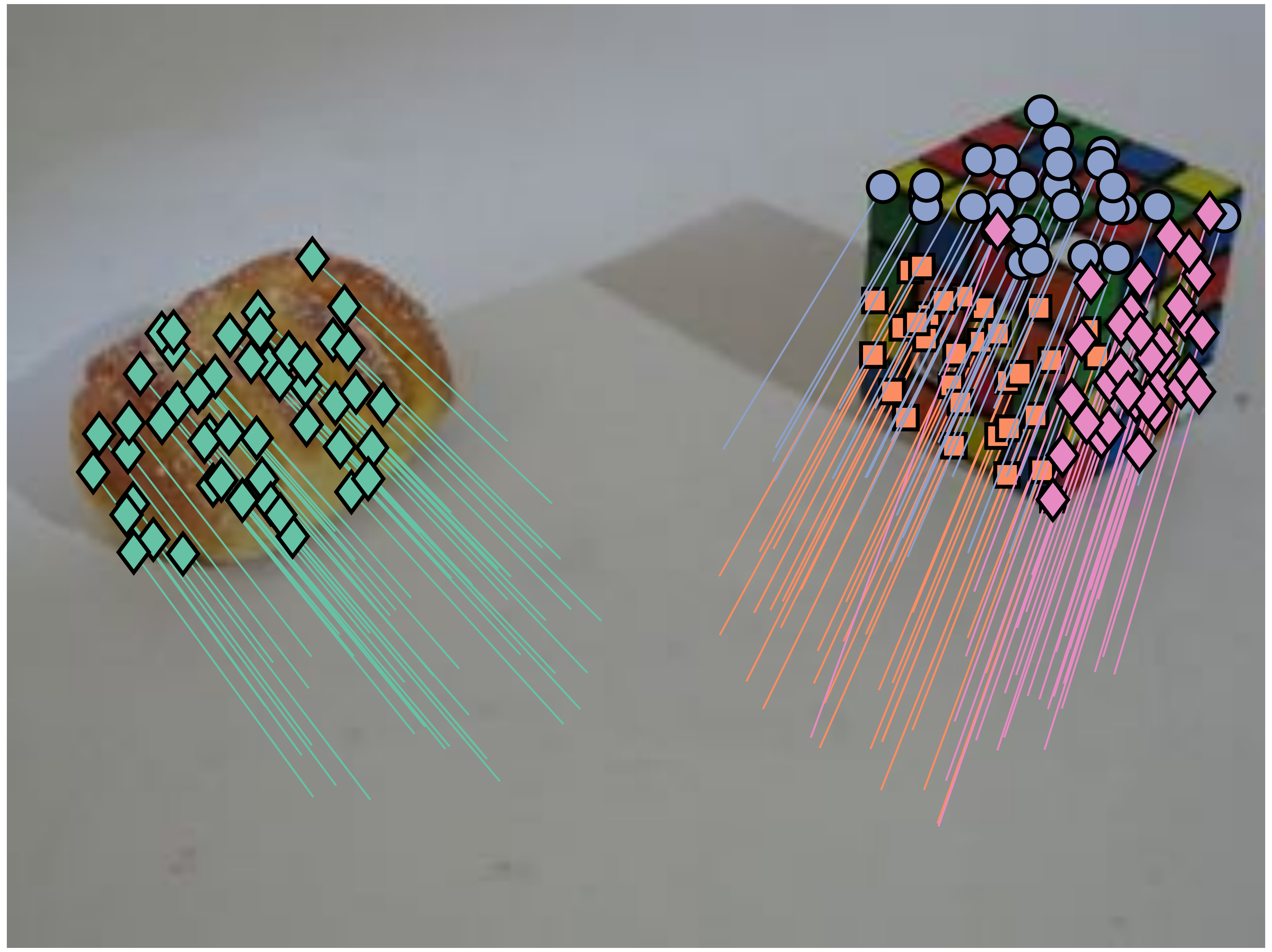
1 complex instance vs many simpler instances

(cirlce/cylinder/fundamental matrix) (lines/palnes/homographies)

Multi class Cascaded T-linkage [Magri and Fusiello CVPR19]

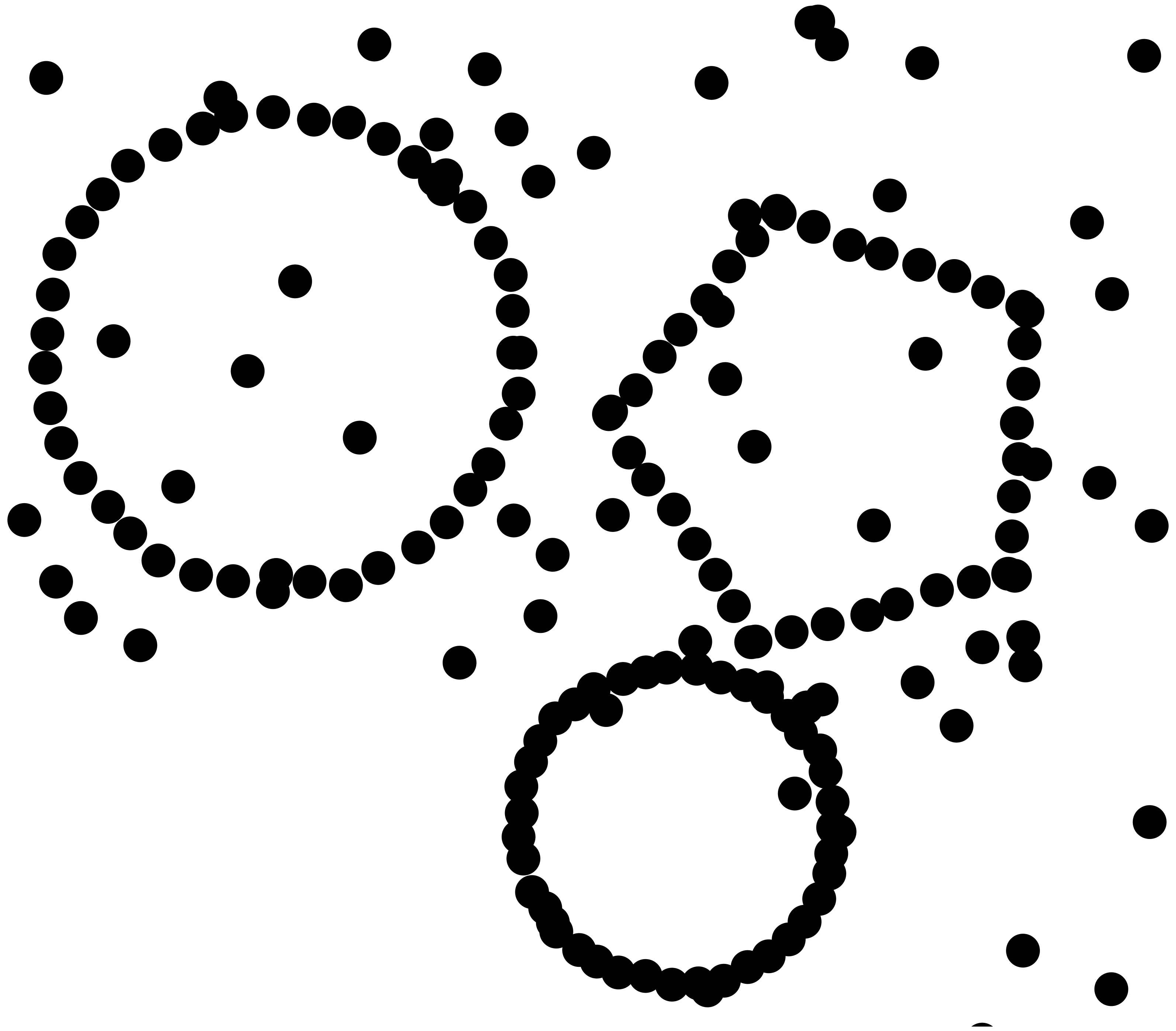
T-linkage in a cascaded fashion combined with GRIC to resolve model selection problem.

Cylinders & planes



Fundamental Matrices & homographies

Multi class cascaded T-linkage [Magri and Fusiello CVPR19]

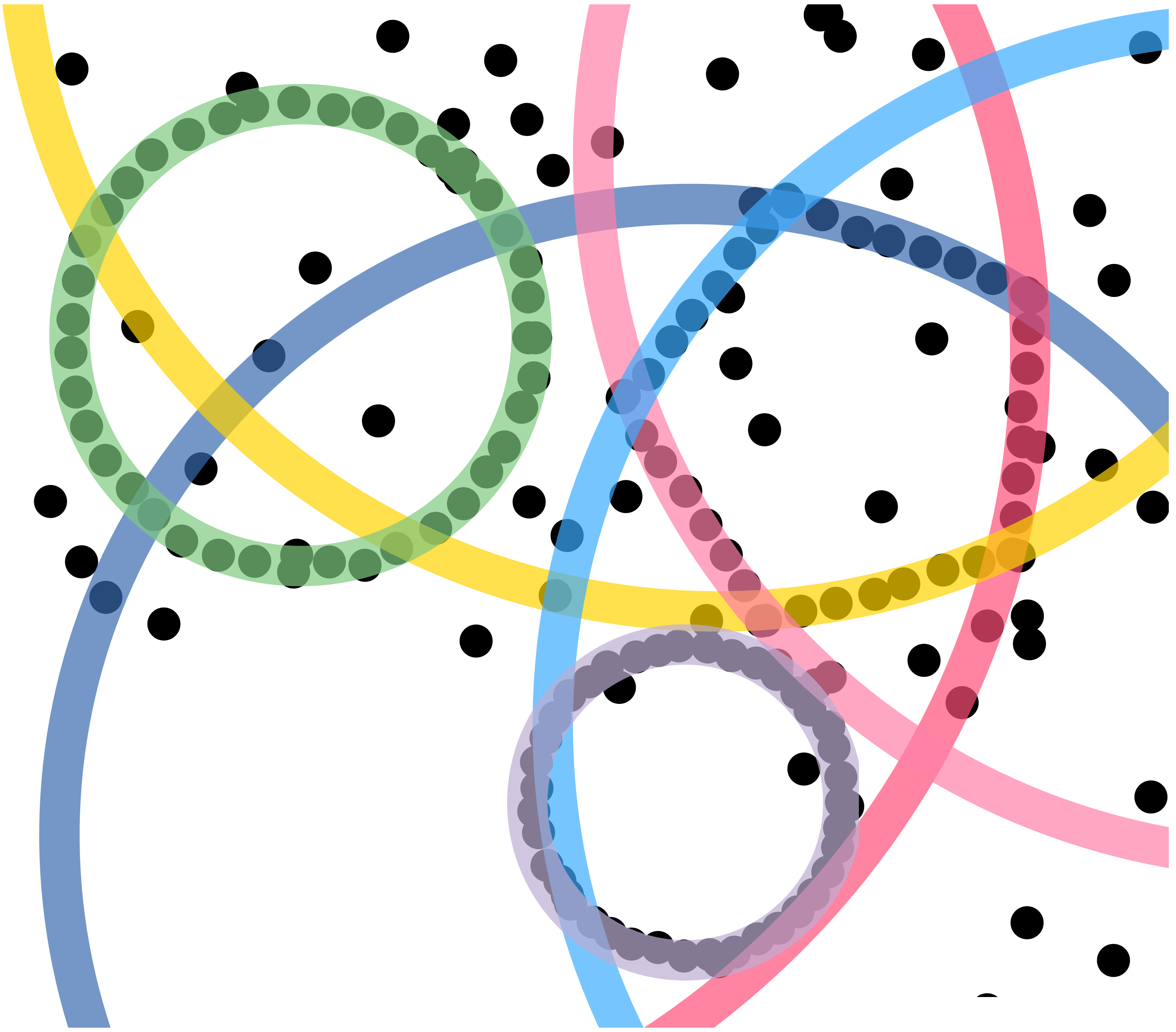


Input: X data, ϵ inlier threshold
Output: Partition in multi-class structures and models

Determine $\mathcal{A}_i$ structures of class Θ_A with T-linkage;
Reject outliers;
for *each structures* $\mathcal{A}_i$ **do**
  Sample compatible models in Θ_B;
  Extract sub-structure(s) B_j with T-linkage;
  Model selection $\mathcal{A}_i$ vs B_j
end
Extract B models from outliers with T-Linkage;
Reject outliers;

Multi class cascaded T-linkage [Magri and Fusiello CVPR19]

Recover complex structures

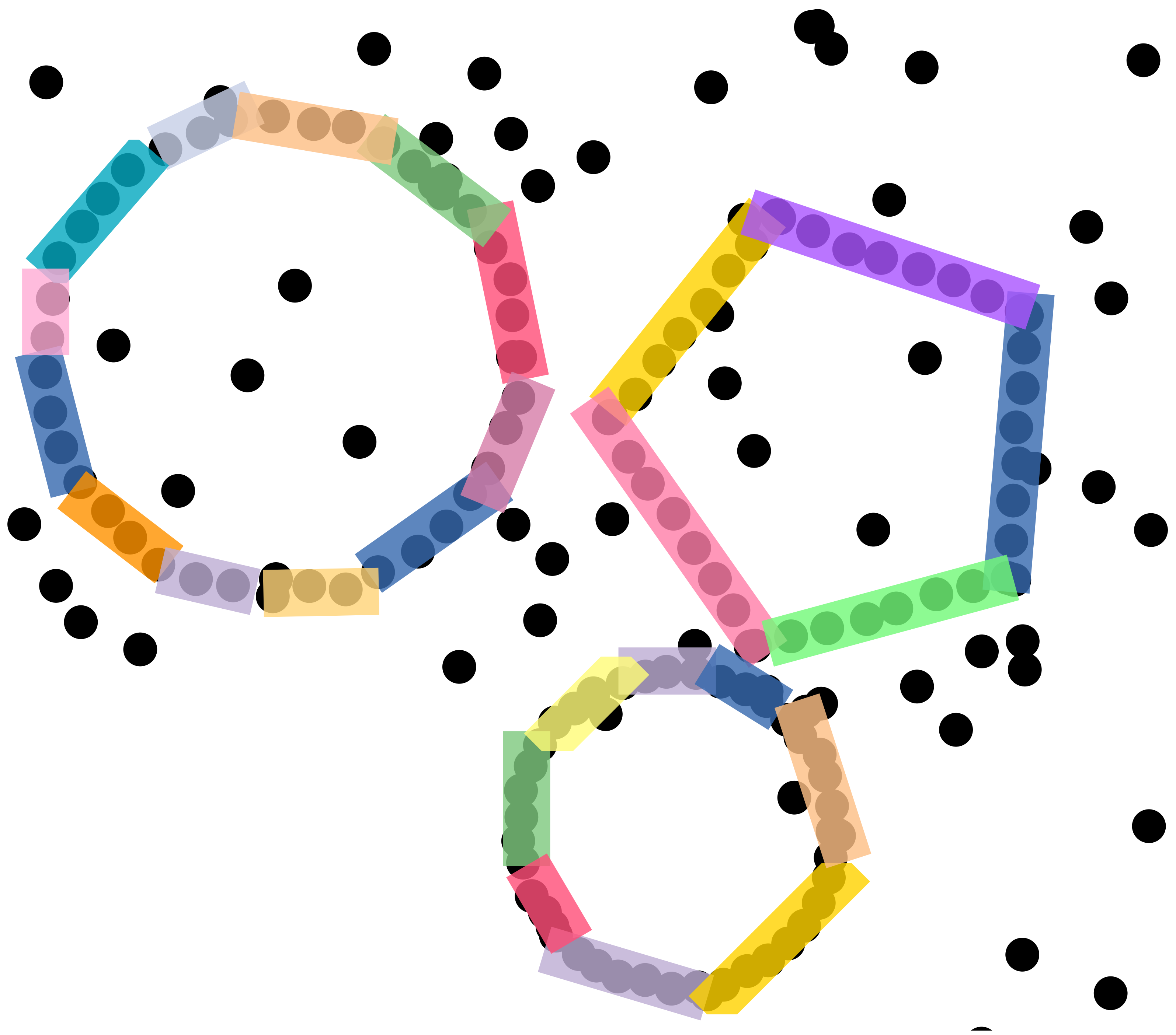


Input: X data, ϵ inlier threshold
Output: Partition in multi-class structures and models

Determine A_i structures of class Θ_A with T-linkage;
Reject outliers;
for *each structures* A_i **do**
  Sample compatible models in Θ_B;
  Extract sub-structure(s) B_j with T-linkage;
  Model selection A_i vs B_j
end
Extract B models from outliers with T-Linkage;
Reject outliers;

Multi class cascaded T-linkage [Magri and Fusiello CVPR19]

Recover nested compatible structures

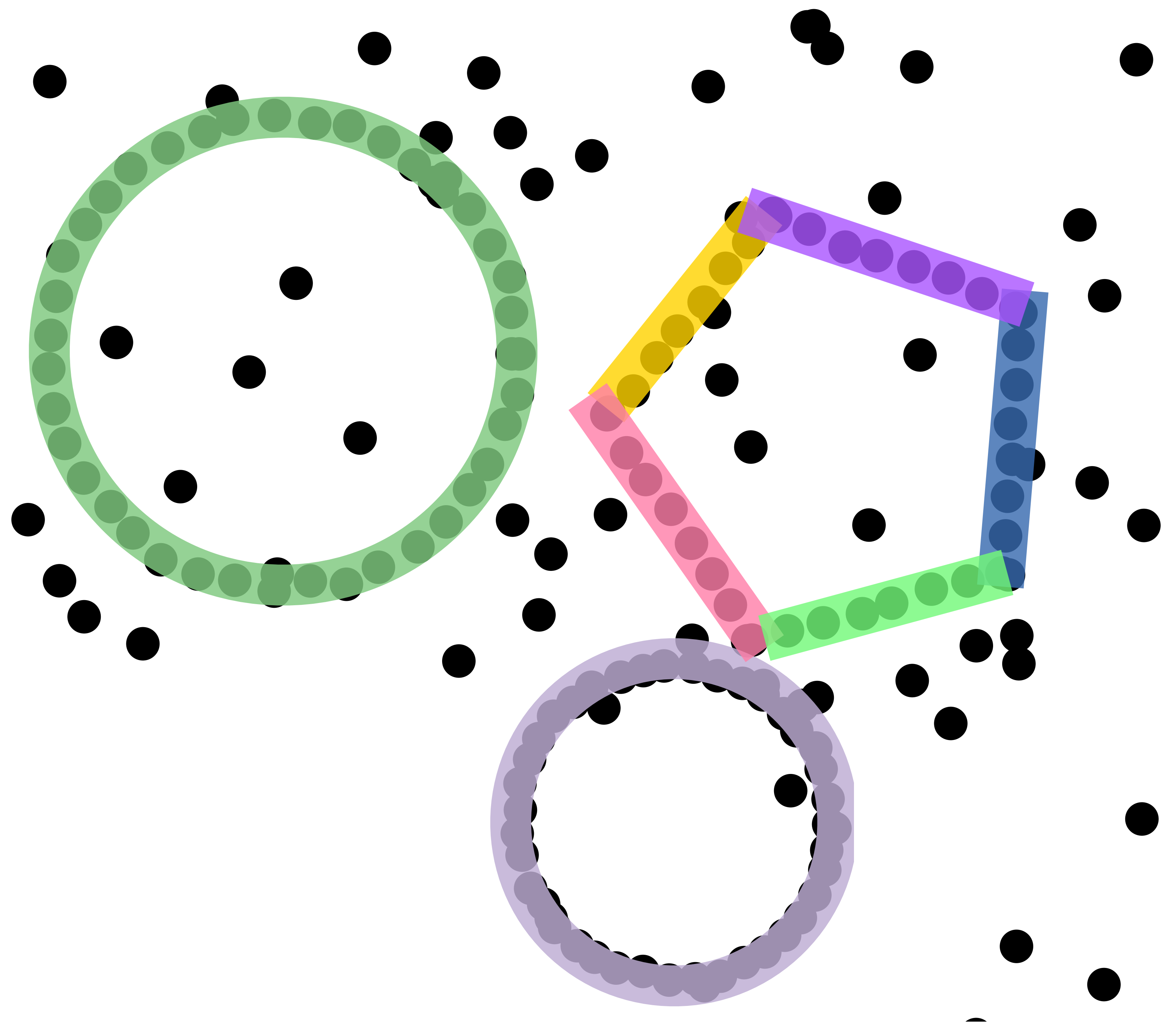


Input: X data, ϵ inlier threshold
Output: Partition in multi-class structures and models

Determine A_i structures of class Θ_A with T-linkage;
Reject outliers;
for *each structures* A_i **do**
    Sample compatible models in Θ_B;
    Extract sub-structure(s) B_j with T-linkage;
    Model selection A_i vs B_j
end
Extract B models from outliers with T-Linkage;
Reject outliers;

Multi class cascaded T-linkage [Magri and Fusiello CVPR19]

Solve 1-vs-many model selection

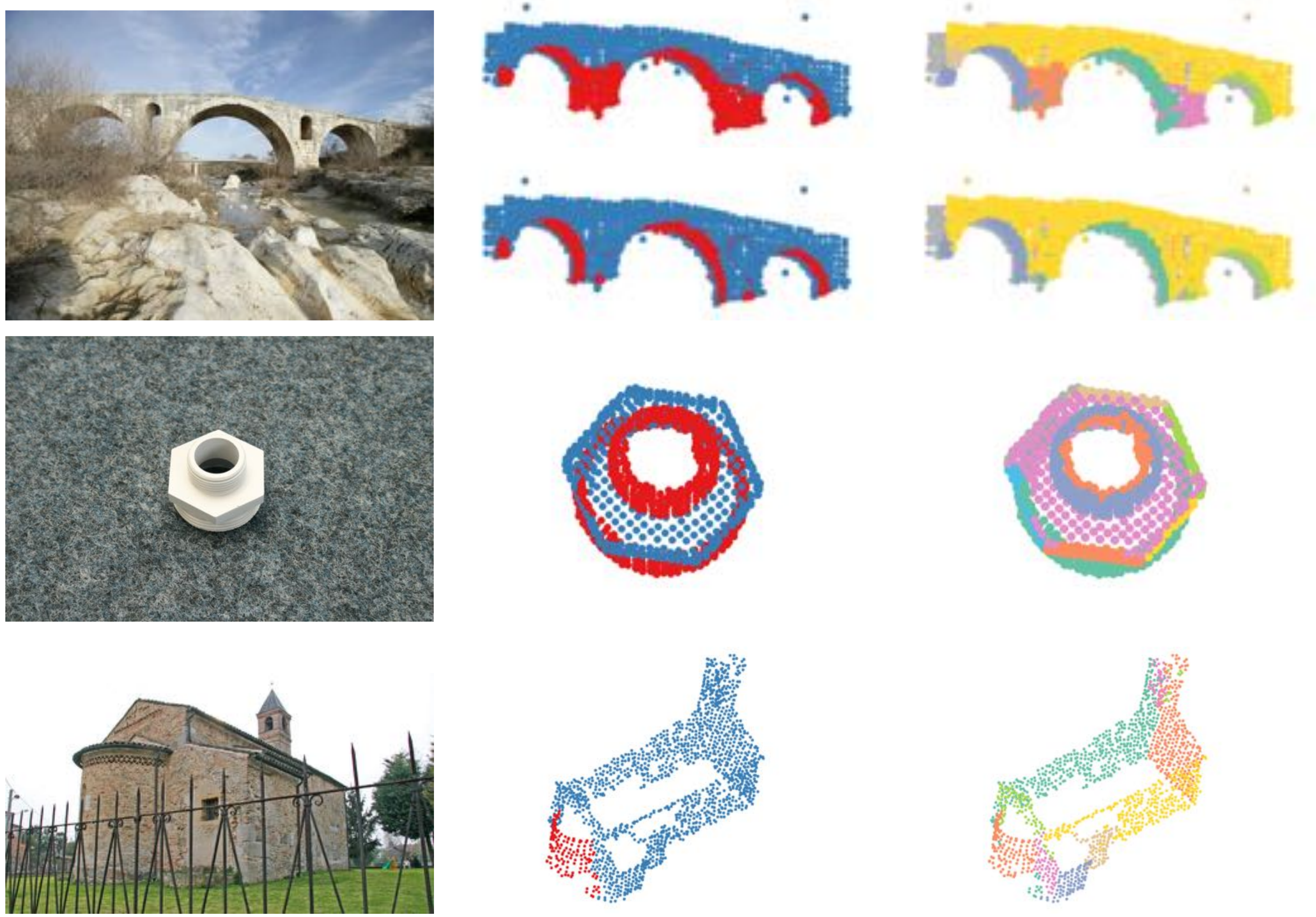


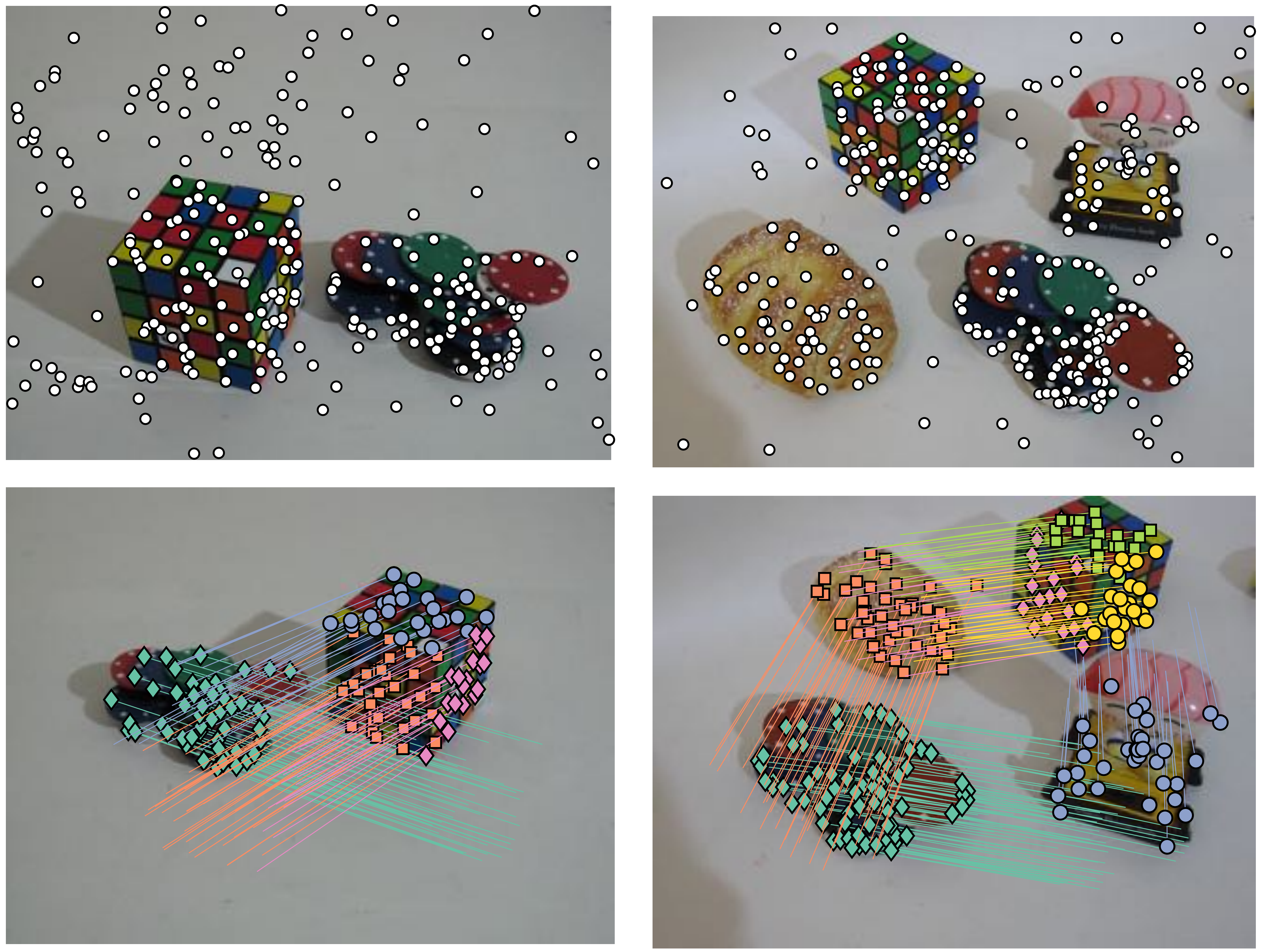
Input: X data, ϵ inlier threshold
Output: Partition in multi-class structures and models

Determine A_i structures of class Θ_A with T-linkage;
Reject outliers;
for *each structures* A_i **do**
    Sample compatible models in Θ_B;
    Extract sub-structure(s) B_j with T-linkage;
    Model selection A_i vs B_j
end
Extract B models from outliers with T-Linkage;
Reject outliers;

Sample results [Magri and Fusiello CVPR19]

Class assignment Model assignement





Conclusions

- We have presented **a** possible solution, among many…

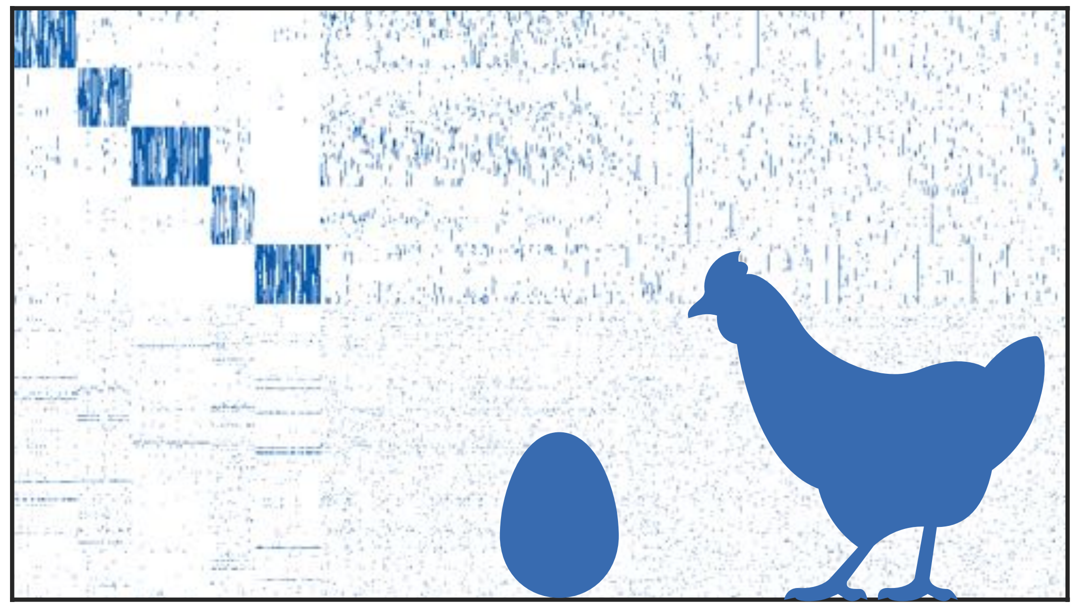
Consensus analysis

- [Xu et Al PatRecLet 90] Randomize Hough Transform
- [Zuliani ICIP 05] Multi Ransac
- [Isack and Boykov CVPR 10] Pearl
- [Magri and Fusiello CVPR 17] Set coverage
- [Barath and Matas ECCV 18] MultiX

Preference analysis

- [Zhang and Koseckà ECCV 06] Residual analysis
- [Toldo and Fusiello ECCV 08] J-linkage
- [Purkait et Al ECCV 14] Hypergraph clustering
- [Magri and Fusiello CVPR 14] T-linkage
- [Magri and Fusiello BMVC 15] Matrix factorization
- [Magri and Fusiello CVPR 19] Cascaded T-linkage

- Clustering provide a powerful and simple formulation, based on few and intelligible parameters



References

The material presented in these slides is mainly based on work done in collaboration with Andrea Fusiello, who is kindly acknowledged here.

J-linkage:

- Toldo, Roberto, and Andrea Fusiello. "Robust multiple structures estimation with j-linkage."ECCV 2008.

T-linkage:

- Magri, Luca, and Andrea Fusiello. "T-linkage: A continuous relaxation of J-linkage for multi-model fitting." *CVPR* 2014.
- Magri, Luca, and Andrea Fusiello. "Multiple structure recovery with t-linkage." *Journal of Visual Communication and Image Representation* 49 (2017).
- Magri, Luca, and Andrea Fusiello. "Fitting Multiple Heterogeneous Models by Multi-Class Cascaded T-Linkage." *CVPR* 2019.

Scan2Bim application:

- Magri, Luca, and Andrea Fusiello. "Reconstruction of interior walls from point cloud data with min-hashed J-linkage." *3DV* 2018.

An improved version appeared in:

- Maset, E., L. Magri, and A. Fusiello. "Improving automatic reconstruction of interior walls from point cloud data." *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).

Sample code can be found, listed under the voice "Multiple-model fitting", at: http://www.diegm.uniud.it/fusiello/index.php/Activities