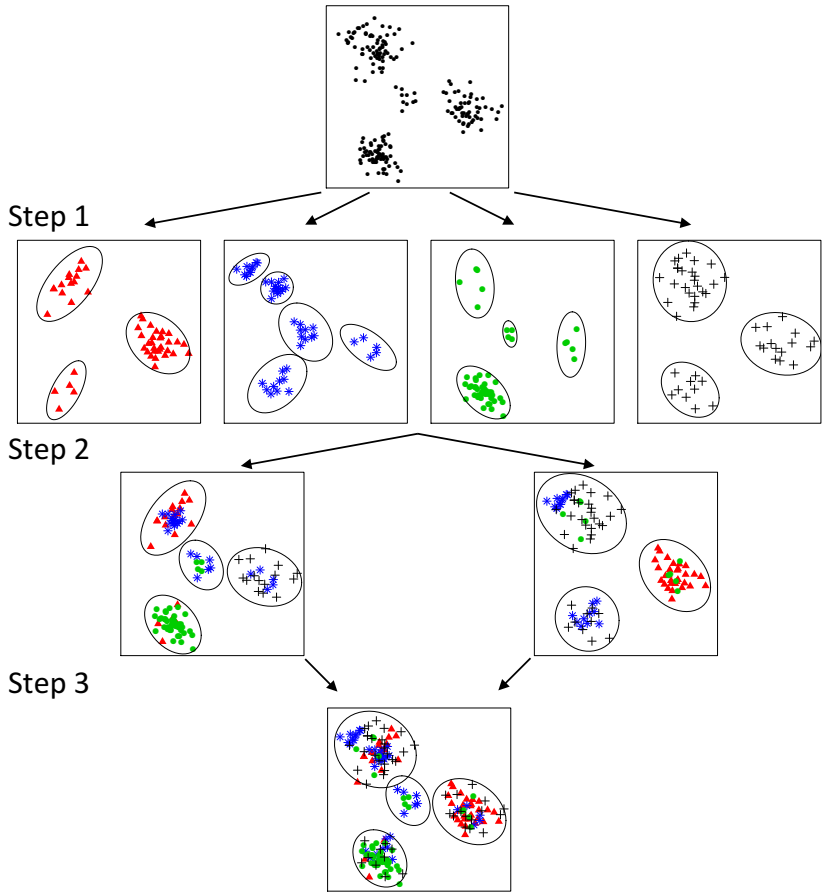
Scalable Bayesian Nonparametric Clustering and Classification

P. Müller, S. Williamson & M. Diesendruck, UT Austin,
D. A. Zuanetti, UF Sao Carlos,
Y. Ni, TX A&M, and Y. Ji, U Chicago



www.math.utexas.edu/users/pmueller/cirm.pdf

Health Records Data

Data: $n = 85,021$ patients

Health Records Data

Data: $n = 85,021$ patients

Variables y_i: fasting blood glucose, white blood cell count, red blood cell count, hemoglobin, platelets, low density lipoproteins, total cholesterol, triglycerides, triketopurine, high density lipoproteins, serum creatinine, serum glutamic oxaloacetic transaminase, total bilirubin, gender, height, weight, blood pressure and waist

Health Records Data

Data: $n = 85,021$ patients

Variables y_i: fasting blood glucose, white blood cell count, red blood cell count, hemoglobin, platelets, low density lipoproteins, total cholesterol, triglycerides, triketopurine, high density lipoproteins, serum creatinine, serum glutamic oxaloacetic transaminase, total bilirubin, gender, height, weight, blood pressure and waist

Outcome: diabetes

Health Records Data

Data: $n = 85,021$ patients

Variables y_i: fasting blood glucose, white blood cell count, red blood cell count, hemoglobin, platelets, low density lipoproteins, total cholesterol, triglycerides, triketopurine, high density lipoproteins, serum creatinine, serum glutamic oxaloacetic transaminase, total bilirubin, gender, height, weight, blood pressure and waist

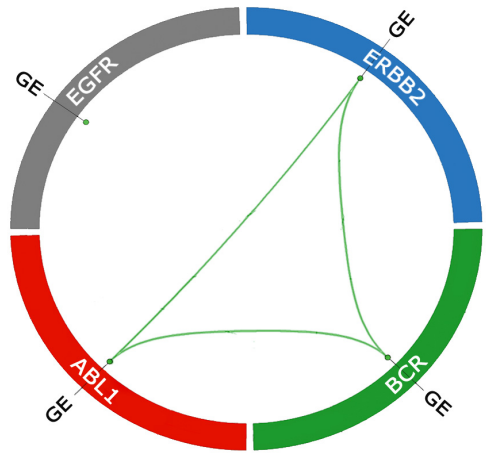
Outcome: diabetes

Goal: clustering, classification & prediction

Zodiac Data

Data: gene-gene interactions of
n = 19,304 genes with all other genes.

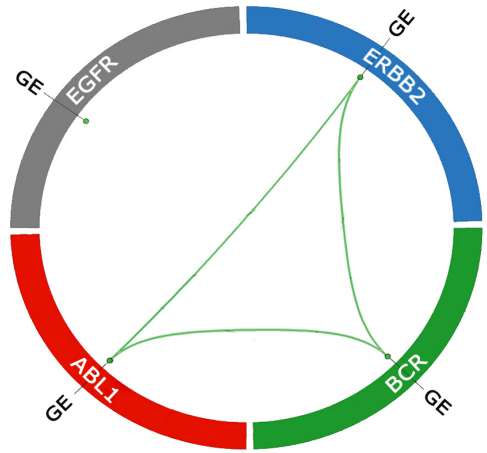
Data on gene-gene interactions from Zodiac (Zhu et al., 2015) with TCGA data.

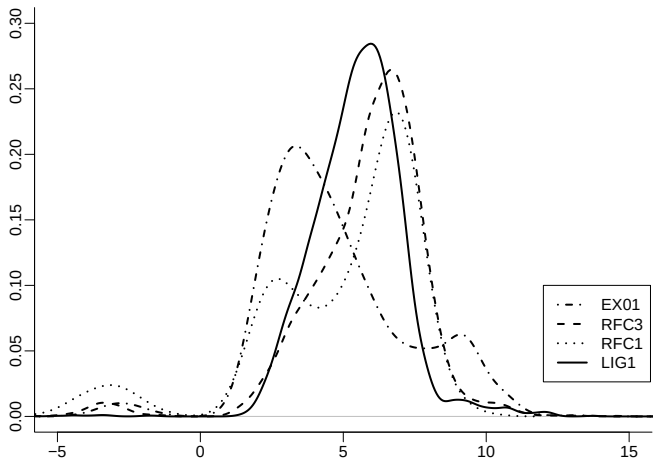


Zodiac Data

Data: gene-gene interactions of
n = 19,304 genes with all other genes.

Data on gene-gene interactions from Zodiac (Zhu et al., 2015) with TCGA data.





Goal: cluster genes by distribution of interactions (with all other $n-1$ genes).

Clustering large data

Problem: clustering large (not "Big") data, $y_i, i = 1, \ldots, n$

Random partition: exchangeable partition of $[n] \equiv \{1, \ldots, n\} \Leftrightarrow$ ties under sampling from discrete random prob measure (Kingman, 1978)

$$\theta_i \sim G \text{ and } G \sim H(G)$$

cluster membership $s_i = j \Leftrightarrow \theta_i = \theta_j^\star$ for jth unique value $\theta_j^\star$; BNP prior $H(G)$, e.g. DP

Clustering large data

Problem: clustering large (not "Big") data, $y_i, i = 1, \ldots, n$

Random partition: exchangeable partition of $[n] \equiv \{1, \ldots, n\} \Leftrightarrow$ ties under sampling from discrete random prob measure (Kingman, 1978)

$$\theta_i \sim G \text{ and } G \sim H(G)$$

cluster membership $s_i = j \Leftrightarrow \theta_i = \theta_j^\star$ for jth unique value $\theta_j^\star$; BNP prior $H(G)$, e.g. DP

$\rightarrow$ "BNP clustering",

Clustering large data

Problem: clustering large (not "Big") data, $y_i, i = 1, \ldots, n$

Random partition: exchangeable partition of $[n] \equiv \{1, \ldots, n\} \Leftrightarrow$ ties under sampling from discrete random prob measure (Kingman, 1978)

$$\theta_i \sim G \text{ and } G \sim H(G)$$

cluster membership $s_i = j \Leftrightarrow \theta_i = \theta_j^\star$ for jth unique value $\theta_j^\star$; BNP prior $H(G)$, e.g. DP

$\rightarrow$ "BNP clustering",

Sampling model: together with $y_i \sim f(y_i \mid \theta_i)$, e.g., normal kernel, BNP mixture:

$$y_i \sim \int f(y_i \mid \theta)\, dG(\theta) \text{ and } G \sim \text{DP},$$

or any other BNP mixture model.

Computation: full posterior simulation becomes challenging with $n > 1000$;

Computation: full posterior simulation becomes challenging with $n > 1000$;

Variational Bayes: DP mixture (Lin, 2013 NIPS; Tank, Foti & Fox, 2015 AISTATS) – on-line learning;

Computation: full posterior simulation becomes challenging with $n > 1000$;

Variational Bayes: DP mixture (Lin, 2013 NIPS; Tank, Foti & Fox, 2015 AISTATS) – on-line learning;

Parallelize algorithm: Williamson, Dubey & Xing (2013, ICML) exploit representation of the DP as normalized Ga process to parallelize inference.

Computation: full posterior simulation becomes challenging with $n > 1000$;

Variational Bayes: DP mixture (Lin, 2013 NIPS; Tank, Foti & Fox, 2015 AISTATS) – on-line learning;

Parallelize algorithm: Williamson, Dubey & Xing (2013, ICML) exploit representation of the DP as normalized Ga process to parallelize inference.

Predictive recursion: Newton, Quintana & Zhang (1998) use approximate predictive recursion, to approximate $p(y_{n+1} \mid \boldsymbol{y})$ under DP mixture model.

Computation: full posterior simulation becomes challenging with $n > 1000$;

Variational Bayes: DP mixture (Lin, 2013 NIPS; Tank, Foti & Fox, 2015 AISTATS) – on-line learning;

Parallelize algorithm: Williamson, Dubey & Xing (2013, ICML) exploit representation of the DP as normalized Ga process to parallelize inference.

Predictive recursion: Newton, Quintana & Zhang (1998) use approximate predictive recursion, to approximate $p(y_{n+1} \mid \boldsymbol{y})$ under DP mixture model.
Similar idea in Wang & Dunson (2011, JCGS) who sequentially build up clusters by assigning $(i+1)$ to a cluster in a partition of $[i]$ (SUGS)

Predictive recursion clustering (PRC)

Predictive recursion clustering: Zuanetti et al., (2018 StatComp); use predictive recursion like Newton et al. (98),

Predictive recursion clustering (PRC)

Predictive recursion clustering: Zuanetti et al., (2018 StatComp);
use predictive recursion like Newton et al. (98),
approximating the posterior predictive
$p(\theta_{i+1} \mid y_1, \ldots, y_{i-1}) \approx g_i(\theta)$:

$$g_i(\theta) = (1 - w_i) g_{i-1}(\theta) + w_i \frac{f(y_i \mid \theta) g_{i-1}(\theta)}{c(y_i, g_{i-1})},$$

Predictive recursion clustering (PRC)

Predictive recursion clustering: Zuanetti et al., (2018 StatComp);
use predictive recursion like Newton et al. (98),
approximating the posterior predictive
$p(\theta_{i+1} \mid y_1, \ldots, y_{i-1}) \approx g_i(\theta)$:

$$g_i(\theta) = (1 - w_i) g_{i-1}(\theta) + w_i \frac{f(y_i \mid \theta) g_{i-1}(\theta)}{c(y_i, g_{i-1})},$$

exact for $i = 1$ (and $w_i = 1/(1 + \alpha)$), and approx beyond.

Predictive recursion clustering (PRC)

Predictive recursion clustering: Zuanetti et al., (2018 StatComp); use predictive recursion like Newton et al. (98), approximating the posterior predictive $p(\theta_{i+1} \mid y_1, \ldots, y_{i-1}) \approx g_i(\theta)$:

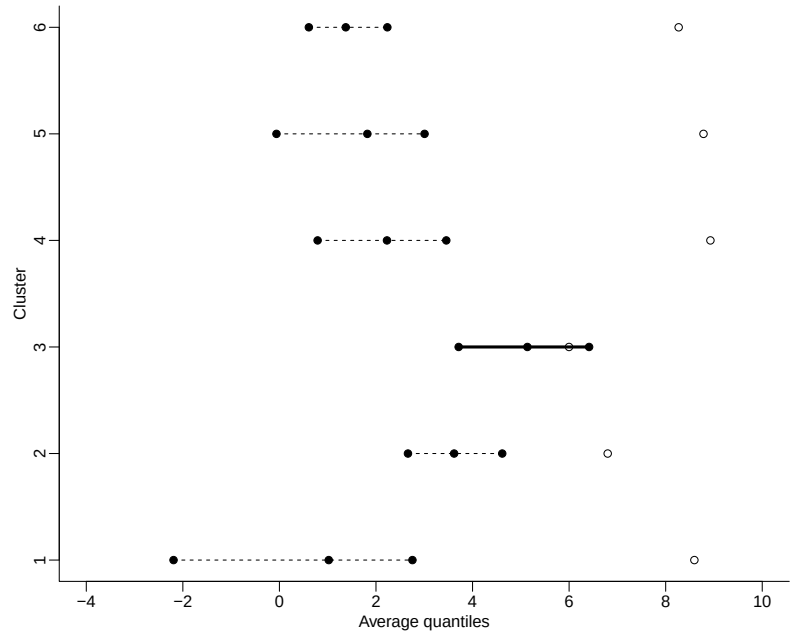
$$g_i(\theta) = (1 - w_i) g_{i-1}(\theta) + w_i \frac{f(y_i \mid \theta) g_{i-1}(\theta)}{c(y_i, g_{i-1})},$$

exact for $i = 1$ (and $w_i = 1/(1+\alpha)$), and approx beyond.

Clustering: g_i builds up as a mixture model, which implicitely defines a random partition (with some computational simplifications, like dropping terms with very small weight, etc.)

GE-GE Interactions

Summarize each gene histogram by Jacobi polynomials $\rightarrow$ clustering of $y_i \in \Re^8$;



PRC clusters. Top (more than 1% of the genes) 6 clusters: average 25%, 50% and 75% quantiles (solid bullets) and average $10 f_{0i}$ (empty circle).

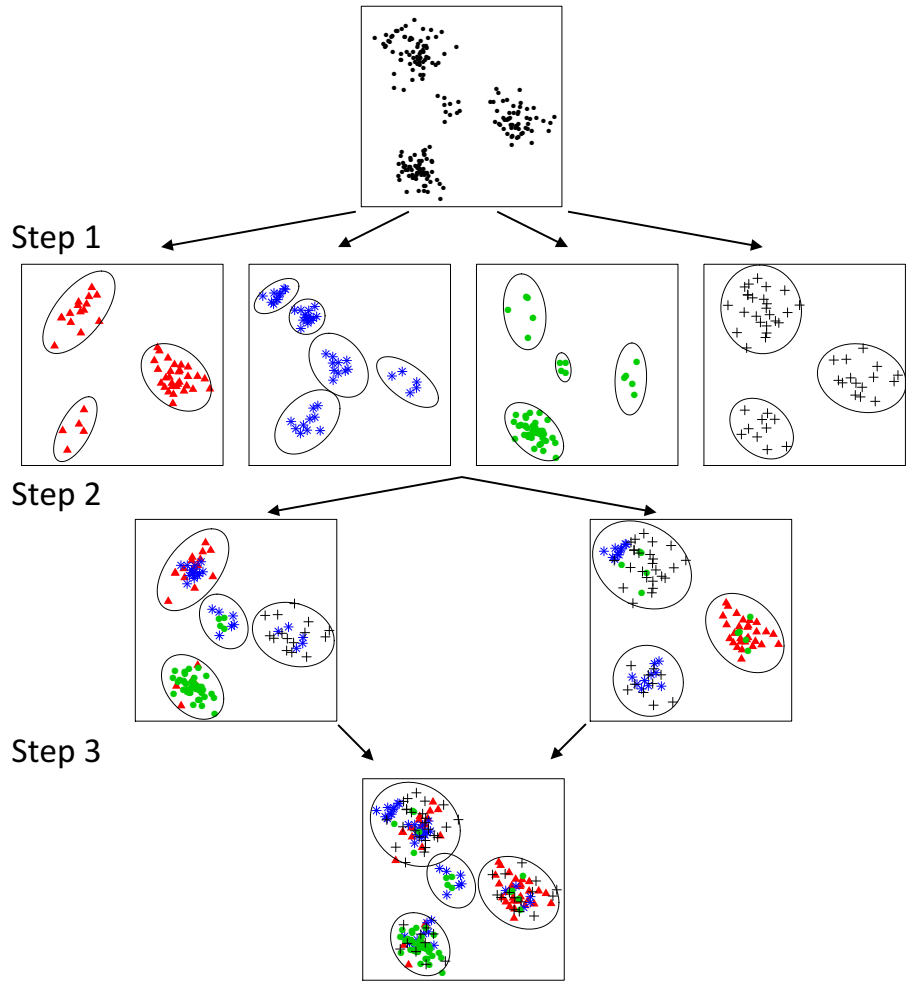
Cluster 3 are the genes of interest.

SIGN algorithm for BNP clustering

1. split data into “shards”;

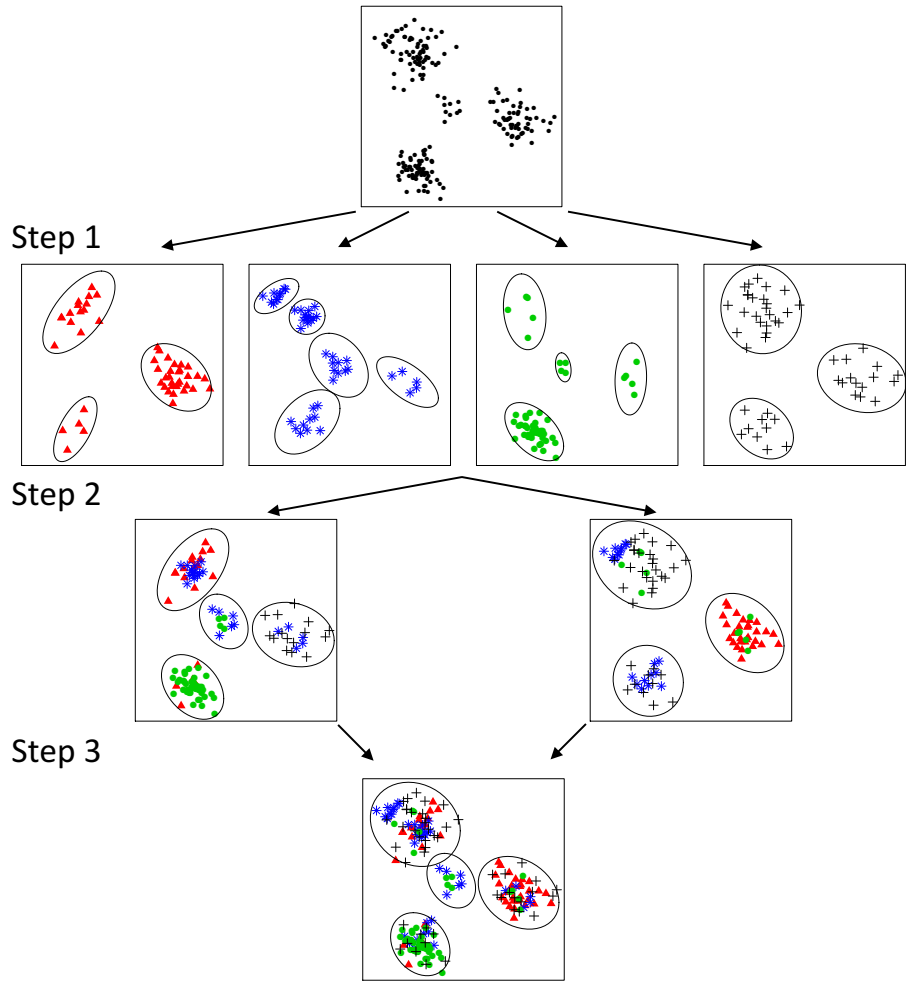
SIGN algorithm for BNP clustering

1. split data into "shards";
2. subset posterior on shards;



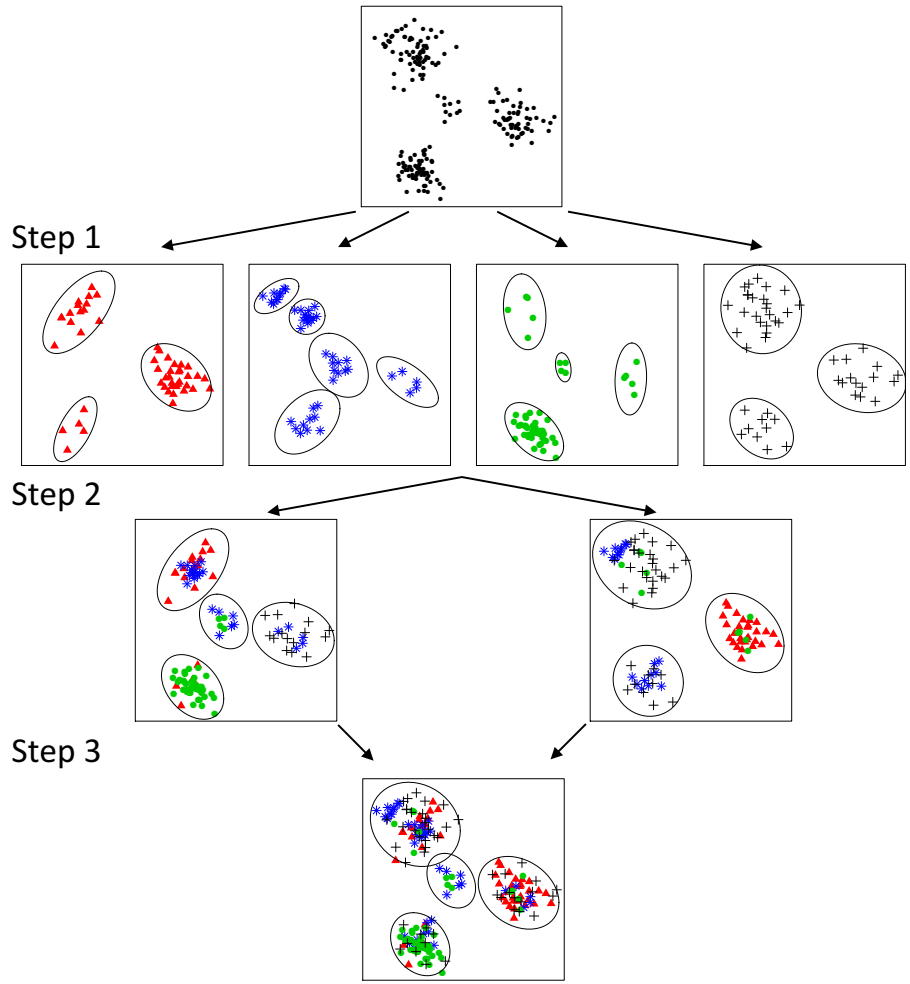
SIGN algorithm for BNP clustering

1. split data into "shards";
2. subset posterior on shards;
3. split clusters into shards $\rightarrow$ Step 2 with clusters as the new units;



SIGN algorithm for BNP clustering

1. split data into "shards";
2. subset posterior on shards;
3. split clusters into shards $\rightarrow$ Step 2 with clusters as the new units;
4. stop when only one shard is left.



Clustering of clusters

In Step $2, 3, \ldots$: clustering of clusters;
similar notion in Argiento et al. (2014) and Malsiner-Walli, Frühwirth-Schnatter & Grün (2017) (for mixture of non-Gaussian dists).

Clustering of clusters

In Step $2, 3, \ldots$: clustering of clusters;
similar notion in Argiento et al. (2014) and Malsiner-Walli, Frühwirth-Schnatter & Grün (2017) (for mixture of non-Gaussian dists).

Notation: let $\tilde{s}_i = j$ if ith cluster (of original units) joins the jth cluster of clusters.

Clustering of clusters

In Step $2, 3, \ldots$: clustering of clusters;
similar notion in Argiento et al. (2014) and Malsiner-Walli, Frühwirth-Schnatter & Grün (2017) (for mixture of non-Gaussian dists).

Notation: let $\tilde{s}_i = j$ if ith cluster (of original units) joins the jth cluster of clusters.

Prior prob: exchangeable prior $\Leftrightarrow p(\rho) = f(n_1, \ldots, n_C)$ for C clusters with cardinalities $n_c \Rightarrow$

$$p(\tilde{s}_i = c \mid \tilde{\boldsymbol{s}}^{-\boldsymbol{i}}) \propto \frac{p(\rho^{+c})}{p(\rho^{-i})}$$

for any BNP prior – easy;

Clustering of clusters

In Step $2, 3, \ldots$: clustering of clusters;
similar notion in Argiento et al. (2014) and Malsiner-Walli, Frühwirth-Schnatter & Grün (2017) (for mixture of non-Gaussian dists).

Notation: let $\tilde{s}_i = j$ if ith cluster (of original units) joins the jth cluster of clusters.

Prior prob: exchangeable prior $\Leftrightarrow p(\rho) = f(n_1, \ldots, n_C)$ for C clusters with cardinalities $n_c \Rightarrow$

$$p(\tilde{s}_i = c \mid \tilde{\boldsymbol{s}}^{-\boldsymbol{i}}) \propto \frac{p(\rho^{+c})}{p(\rho^{-i})}$$

for any BNP prior – easy;

Transdim MCMC: Neal's (2000) Algorithm 8 for new singleton clusters

Approximation

Partition: $s_i = k$ if ith unit in kth cluster;
alternatively use indicators $\delta_{ij} = I(s_i = s_j)$.

Approximation

Partition: $s_i = k$ if ith unit in kth cluster;
alternatively use indicators $\delta_{ij} = I(s_i = s_j)$.

Subset posteriors: Let $[n] = A \cup B$ denote two shards;
$\boldsymbol{\delta}_A = (\delta_{ij},\ i, j \in A)$, same for $\boldsymbol{\delta}_B$, $\boldsymbol{\delta}_{AB} = (\delta_{ij},\ i \in A,\ j \in B)$

Approximation

Partition: $s_i = k$ if ith unit in kth cluster;
alternatively use indicators $\delta_{ij} = I(s_i = s_j)$.

Subset posteriors: Let $[n] = A \cup B$ denote two shards;
$\boldsymbol{\delta}_A = (\delta_{ij},\ i,j \in A)$, same for $\boldsymbol{\delta}_B$, $\boldsymbol{\delta}_{AB} = (\delta_{ij},\ i \in A,\ j \in B)$

$$p(\boldsymbol{\delta} \mid \boldsymbol{y}) \approx$$
$$q(\boldsymbol{\delta} \mid \boldsymbol{y}) \equiv p(\boldsymbol{\delta}_A \mid \boldsymbol{y}_A)\, p(\boldsymbol{\delta}_B \mid \boldsymbol{y}_B)\, p(\boldsymbol{\delta}_{AB} \mid \boldsymbol{\delta}_A, \boldsymbol{\delta}_B, \boldsymbol{y})$$

Approximation

Partition: $s_i = k$ if ith unit in kth cluster;
alternatively use indicators $\delta_{ij} = I(s_i = s_j)$.

Subset posteriors: Let $[n] = A \cup B$ denote two shards;
$\boldsymbol{\delta}_A = (\delta_{ij},\ i,j \in A)$, same for $\boldsymbol{\delta}_B$, $\boldsymbol{\delta}_{AB} = (\delta_{ij},\ i \in A,\ j \in B)$

$$p(\boldsymbol{\delta} \mid \boldsymbol{y}) \approx$$
$$q(\boldsymbol{\delta} \mid \boldsymbol{y}) \equiv p(\boldsymbol{\delta}_A \mid \boldsymbol{y}_A)\, p(\boldsymbol{\delta}_B \mid \boldsymbol{y}_B)\, p(\boldsymbol{\delta}_{AB} \mid \boldsymbol{\delta}_A, \boldsymbol{\delta}_B, \boldsymbol{y})$$

Summary: judge approximation by

$$F_{.1} = \ \% \text{ pairs with } |E_q(\delta_{ij}) - E(\delta_{ij} \mid \boldsymbol{y})| > 0.1$$

Simulations

Simulation I: simple mixture of $C_0 = 5$ (truth) normals; comparison with PY mixture (full MCMC) and DBSCAN (Ester et al., 1996)

	SIGN	SIGN-VI	PYM	DBSCAN
C	4.94 (0.31)	4.90 (0.30)	5.08 (0.27)	3.64 (1.63)
MISC	0.08 (0.03)	0.09 (0.03)	0.04 (0.01)	0.41 (0.11)
MSE	0.01 (0.01)	0.01 (0.01)	0.01 (0.00)	0.49 (0.18)

MISC=misclassification rate; MSE=estimation of cluster-specific means

Simulations

Simulation I: simple mixture of $C_0 = 5$ (truth) normals; comparison with PY mixture (full MCMC) and DBSCAN (Ester et al., 1996)

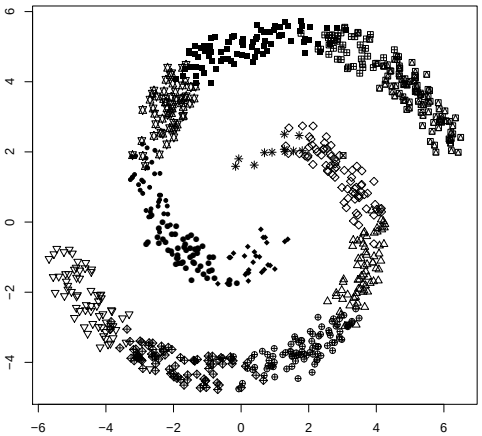
	SIGN	SIGN-VI	PYM	DBSCAN
C	4.94 (0.31)	4.90 (0.30)	5.08 (0.27)	3.64 (1.63)
MISC	0.08 (0.03)	0.09 (0.03)	0.04 (0.01)	0.41 (0.11)
MSE	0.01 (0.01)	0.01 (0.01)	0.01 (0.00)	0.49 (0.18)

MISC=misclassification rate; MSE=estimation of cluster-specific means

Estimated partition at the end of each step: need estimated clusters after each step; use Dahl (2006) summary. Alternatively, variation of information loss (SIGN-VI)

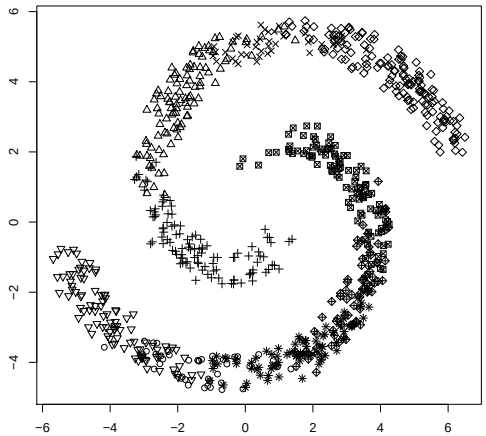
Simulation II: two spirals

full MCMC



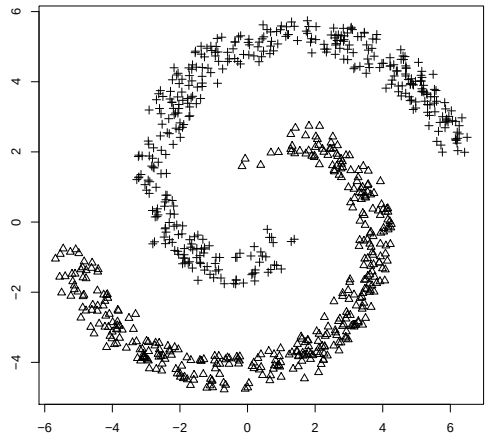
$\hat{C} = 14.10(1.37)$

SIGN



9.28(.93)

DBSCAN



1.98(0.14)

The reduced $\hat{C}$ under the SIGN approximation is typical. CLusters from early steps can be merged, but never split.

Prediction

Density estimation: BNP clustering,

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(\quad y_i \mid \rho, \boldsymbol{\theta}),$$

implies density estimation $p(y_{n+1} \mid \boldsymbol{y})$.

Prediction

Density estimation: BNP clustering,

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(\quad y_i \mid \rho, \boldsymbol{\theta}),$$

implies density estimation $p(y_{n+1} \mid \boldsymbol{y})$.

Regression & prediction: to be useful for regression, need conditioning in $p(\rho \mid \boldsymbol{x})$, on covariates x_i;
$p(y_{n+1} \mid x_{n+1}, \boldsymbol{y}, \boldsymbol{x})$ defindes desired regression.

Prediction

Density estimation: BNP clustering,

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(\quad y_i \mid \rho, \boldsymbol{\theta}),$$

implies density estimation $p(y_{n+1} \mid \boldsymbol{y})$.

Regression & prediction: to be useful for regression, need conditioning in $p(\rho \mid \boldsymbol{x})$, on covariates x_i;
$p(y_{n+1} \mid x_{n+1}, \boldsymbol{y}, \boldsymbol{x})$ defindes desired regression.

Augmented model: augment response to $z_i = (x_i, y_i)$ and proceed as before

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(x_i, y_i \mid \rho, \boldsymbol{\theta}).$$

Predictive $p(x_{n+1}, y_{n+1} \mid \boldsymbol{x}, \boldsymbol{y})$ implies regression; conditional regression or density regression (Park & Dunson, 2010; M & al, 1996).

Prediction

Density estimation: BNP clustering,

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(\quad y_i \mid \rho, \boldsymbol{\theta}),$$

implies density estimation $p(y_{n+1} \mid \boldsymbol{y})$.

Regression & prediction: to be useful for regression, need conditioning in $p(\rho \mid \boldsymbol{x})$, on covariates x_i;
$p(y_{n+1} \mid x_{n+1}, \boldsymbol{y}, \boldsymbol{x})$ defindes desired regression.

Augmented model: augment response to $z_i = (x_i, y_i)$ and proceed as before

$$p(\rho)\, p(\boldsymbol{\theta} \mid \rho)\, p(x_i, y_i \mid \rho, \boldsymbol{\theta}).$$

Predictive $p(x_{n+1}, y_{n+1} \mid \boldsymbol{x}, \boldsymbol{y})$ implies regression; conditional regression or density regression (Park & Dunson, 2010; M & al, 1996).

PPMx: define mroe general $p(\rho \mid \boldsymbol{x})$, avoiding explicit modeling of a covariate distribution

Simulations

Classification: AUC.
Comparison with full MCMC ("PPMx"), BART, random forest (RF), logistic regression (LR) and SVM

	Simulation III	Simulation IV
SIGN	0.808 (0.067)	0.838 (0.067)
PPMx	0.824 (0.060)	0.841 (0.063)
BART	0.755 (0.062)	0.866 (0.050)
RF	0.793 (0.059)	0.838 (0.067)
LR	0.600 (0.091)	0.524 (0.073)
SVM	0.622 (0.077)	0.585 (0.077)

Results

Response y_i: indicator for diabetes

Covariates x_i: white blood cell count (WBC), red blood cell count (RBC), hemoglobin (HGB), platelets (PLT), fasting blood glucose (FBG), low density lipoproteins (LDL), total cholesterol (TC), triglycerides (Trig), triketopurine (Trik), high density lipoproteins (HDL), serum creatinine (SCr), serum glutamic oxaloacetic transaminase (SGOT), and total bilirubin (TB); sex, height, weight, blood pressure, and waist.

Data: $n = 85,021$ patients

SIGN: $M_1 = 250$ shards $\rightarrow$ 1351 local clusters;
$M_2 = 5$ shards $\rightarrow$ 25 regional clusters;
Algorithm stops at step $K = 3$

EHR – Results

AUC for classification by diabetes:

	EHR	Bank
SIGN	0.880	0.825
PPMx	-	-
BART	0.867	0.792
RF	0.869	0.786
LR	0.856	0.781
SVM	0.856	0.761

("Bank" is another data set, on success of telemarketing)

GAN

GAN: Chinese policy requires "China first" publication;
We use a "Generative Adversarial Network" (GAN) (Goodfellow et al. 2014) to generate a hypothetical repeat

GAN

GAN: Chinese policy requires "China first" publication;
We use a "Generative Adversarial Network" (GAN) (Goodfellow et al. 2014) to generate a hypothetical repeat

- One network does density estimation $p(x_i, y_i)$ and predictive simulation of n fake data, $i = n + 1, \ldots, n + n$; pass the augmented data to a second network:

GAN

GAN: Chinese policy requires "China first" publication;
We use a "Generative Adversarial Network" (GAN) (Goodfellow et al. 2014) to generate a hypothetical repeat

- One network does density estimation $p(x_i, y_i)$ and predictive simulation of n fake data, $i = n+1, \ldots, n+n$; pass the augmented data to a second network:
- A second network tries to discriminate original versus fake data.

GAN

GAN: Chinese policy requires "China first" publication;
We use a "Generative Adversarial Network" (GAN) (Goodfellow et al. 2014) to generate a hypothetical repeat

- One network does density estimation $p(x_i, y_i)$ and predictive simulation of n fake data, $i = n + 1, \ldots, n + n$; pass the augmented data to a second network:
- A second network tries to discriminate original versus fake data.
- Iterate until discrimination is impossible.

GAN

GAN: Chinese policy requires "China first" publication;

We use a "Generative Adversarial Network" (GAN) (Goodfellow et al. 2014) to generate a hypothetical repeat

- One network does density estimation $p(x_i, y_i)$ and predictive simulation of n fake data, $i = n + 1, \ldots, n + n$; pass the augmented data to a second network:
- A second network tries to discriminate original versus fake data.
- Iterate until discrimination is impossible.

We comply with Chinese law, but statistical inference is identical

Variations: Overlapping Shards

Simplification: replace clustering of clusters (and beyond) by deterministic match and merge.

Variations: Overlapping Shards

Simplification: replace clustering of clusters (and beyond) by deterministic match and merge.

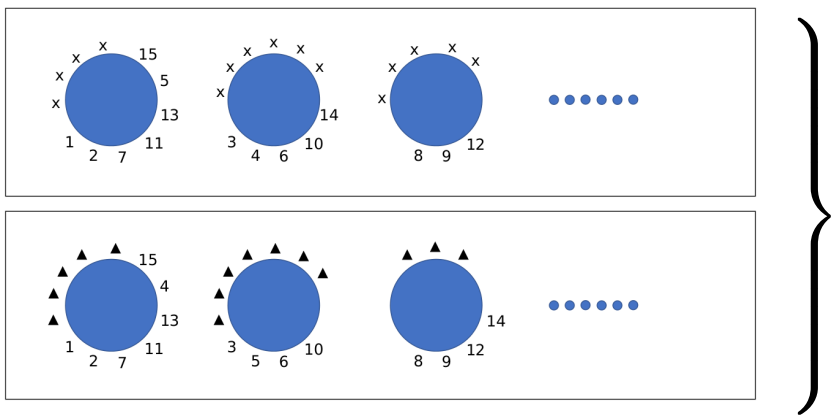
Clustering: Split data into shards *with common overlap*

Variations: Overlapping Shards

Simplification: replace clustering of clusters (and beyond) by deterministic match and merge.

Clustering: Split data into shards *with common overlap*

Consensus: Merge clusters C_1, C_2 with m_{12} common members if $\min\left\{\frac{m_{12}}{|C_1|}, \frac{m_{12}}{|C_1|}\right\} > \lambda$

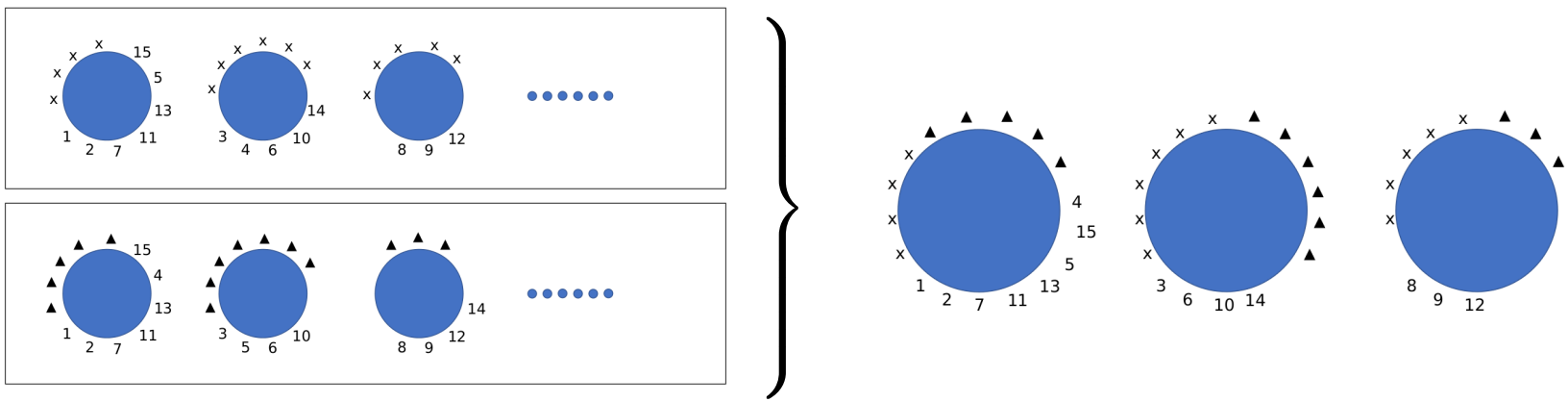


Variations: Overlapping Shards

Simplification: replace clustering of clusters (and beyond) by deterministic match and merge.

Clustering: Split data into shards *with common overlap*

Consensus: Merge clusters C_1, C_2 with m_{12} common members if $\min\left\{\frac{m_{12}}{|C_1|}, \frac{m_{12}}{|C_1|}\right\} > \lambda$

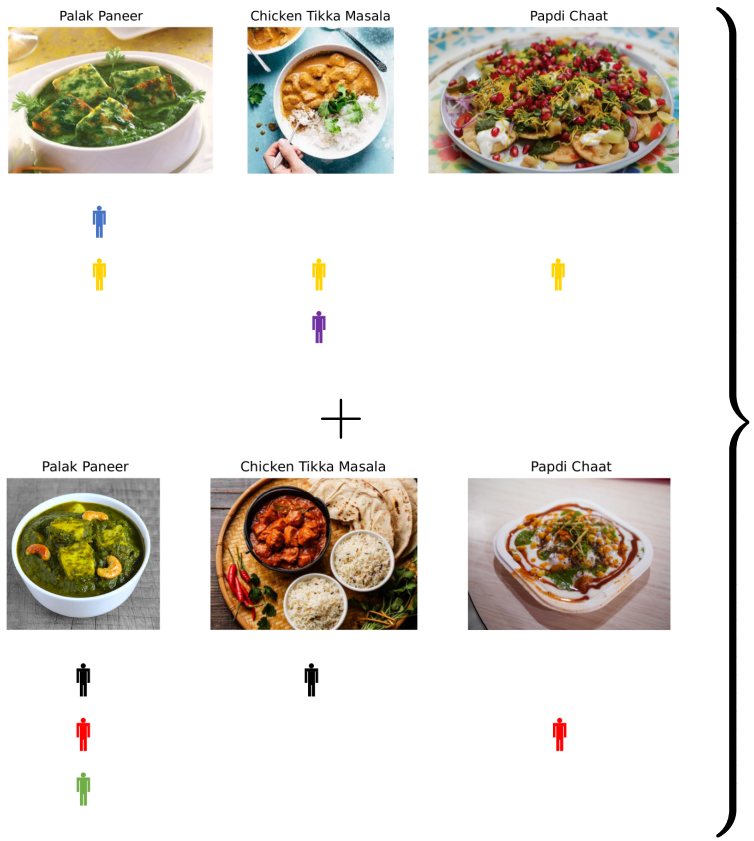


Random subsets (feature allocation)

Feature allocation: Merge features F_1, F_2 if feature-specific parameters are close, $d(F_1, F_2) < \lambda$.

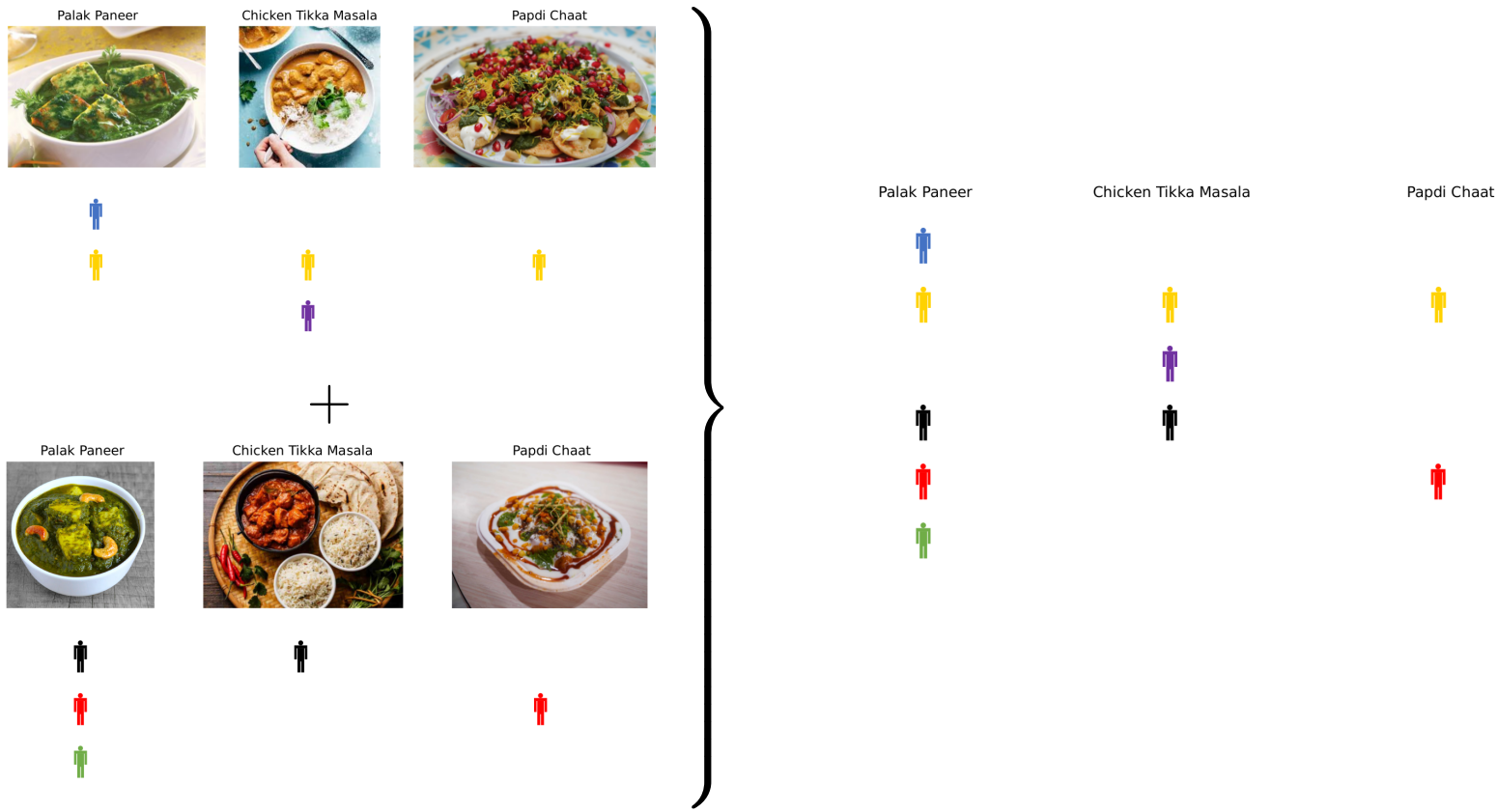
Random subsets (feature allocation)

Feature allocation: Merge features F_1, F_2 if feature-specific parameters are close, $d(F_1, F_2) < \lambda$.



Random subsets (feature allocation)

Feature allocation: Merge features F_1, F_2 if feature-specific parameters are close, $d(F_1, F_2) < \lambda$.



Example: Tumor heterogeneity

- experimental unites $=$ mutations $i = 1, \ldots, n$;

Example: Tumor heterogeneity

- experimental unites = mutations $i = 1, \ldots, n$;
- features = homogeneous subclones $F_j \subseteq [n]$, subsets of mutations;

Example: Tumor heterogeneity

- experimental unites = mutations $i = 1, \ldots, n$;
- features = homogeneous subclones $F_j \subseteq [n]$, subsets of mutations;
- Each subclone is linked with a set of weights, $\boldsymbol{w}_j$, for observed tissue samples,
  use $d(\boldsymbol{w}_j, \boldsymbol{w}_\ell)$ to decide merging

Double feature allocation

Double feature allocation: two sets of experimental units, $i = 1, \ldots, n$ (e.g., patients) and $s = 1, \ldots, S$ (e.g., symptoms); each feature $F_j \subseteq [n]$ (e.g., disease) is associated with a subset $S_j \subseteq [S]$.

Double feature allocation

Double feature allocation: two sets of experimental units, $i = 1, \ldots, n$ (e.g., patients) and $s = 1, \ldots, S$ (e.g., symptoms); each feature $F_j \subseteq [n]$ (e.g., disease) is associated with a subset $S_j \subseteq [S]$.

Clustering of clusters: Same – merge features F_1, F_2 if feature-specific parameters S_1, S_2 are close, $d(S_1, S_2) < \lambda$.

Double feature allocation

Double feature allocation: two sets of experimental units, $i = 1, \ldots, n$ (e.g., patients) and $s = 1, \ldots, S$ (e.g., symptoms); each feature $F_j \subseteq [n]$ (e.g., disease) is associated with a subset $S_j \subseteq [S]$.

Clustering of clusters: Same – merge features F_1, F_2 if feature-specific parameters S_1, S_2 are close, $d(S_1, S_2) < \lambda$.

Example: EHR, features = "*disease*", $F_j \subseteq [n]$, subsets of patients; Each disease is linked to a set S_j of symptoms, $d(S_j, S_\ell) = |S_j \cap S_\ell| / |S_j \cup S_\ell|$.

Summary

- Model-based clustering (feature allocation, double FA) is more flexible than purely algorithmic methods, but computationally challenging for large n (also for large p)
- Several algorithms, using predictive recursion, approximation, parallelization, subset posteriors (consensus MC)
- Approxiamte posterior uncertainties important for decision problems (e.g., phenotype discovery in EHR data)