Clustering a medieval social network by SOM using a kernel based distance measure

Nathalie Villa-Vialaneix[(1)] Romain Boulet[(2)]
http://www.nathalievilla.org

[(1)]LSP, Toulouse - nvilla@cict.fr

[(2)]MIP, Toulouse

ESANN, Wednesday 25 april 2007

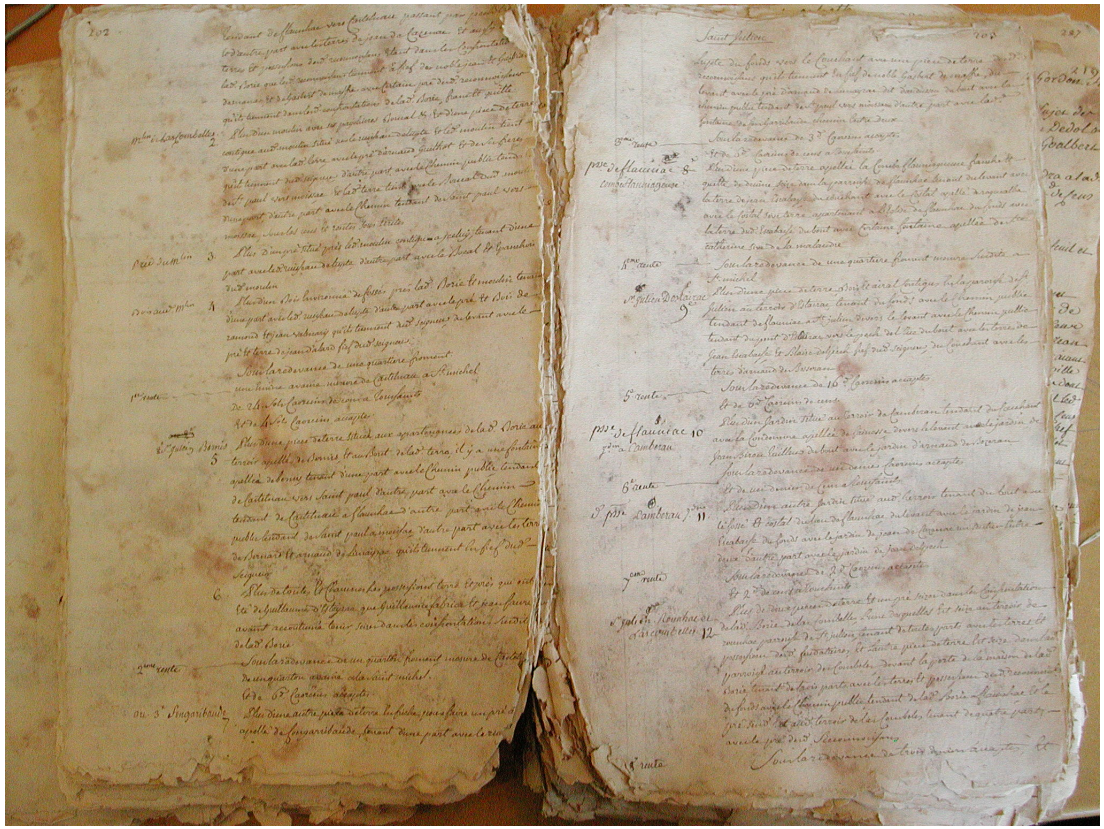
Table of contents

Table of contents

Exploring a big historic database

Data

10 000 agrarian contracts from four seignories of South West of France established between 1240 and 1520.



Exploring a big historic database

Data

10 000 agrarian contracts from four seignories of South West of France established between 1240 and 1520.

Historian's questions:

- family or geographical social links ?
- central people having a main social role ?
- ...

Exploring a big historic database

Data

10 000 agrarian contracts from four seignories of South West of France established between 1240 and 1520.

Historian's questions:

- family or geographical social links ?
- central people having a main social role ?
- ...

⇒ **Data mining** is required.

Exploring a big historic database

Data

10 000 agrarian contracts from four seignories of South West of France established between 1240 and 1520.

Historian's questions:

- family or geographical social links ?
- central people having a main social role ?
- ...

⇒ **Data mining** is required.

First step

Restriction to a part of this database: the Castelnau-Montratier Seignory before the Hundred Years' War.

A graph clustering problem

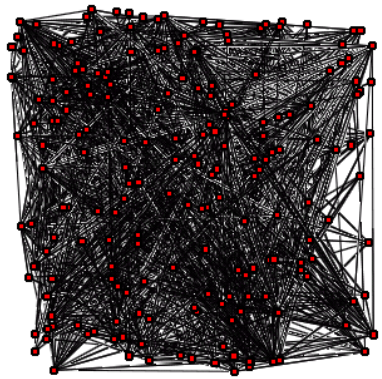
From the database, building a **weighted graph**:

- with 226 **vertices** $x_1, \ldots, x_n$:= peasants found in the contracts;

A graph clustering problem

From the database, building a **weighted graph**:

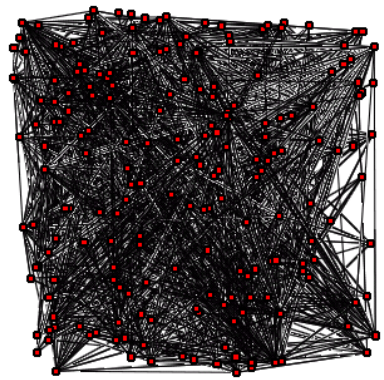
- with 226 **vertices** $x_1, \ldots, x_n$:= peasants found in the contracts;
- with **weights** $(w_{i,j})_{i,j=1,\ldots,n}$:= $\sharp$\{contracts where x_i and x_j are mentionned\}.



A graph clustering problem

From the database, building a **weighted graph**:

- with 226 **vertices** $x_1, \ldots, x_n$:= peasants found in the contracts;
- with **weights** $(w_{i,j})_{i,j=1,\ldots,n}$:= $\sharp$\{contracts where x_i and x_j are mentionned\}.



Clustering the vertices to understand the social links between peasants.

Previous work

In **[Boulet & Jouve, 2007]**, algebraic study of the non-weighted graph:

- **small world** graph $\Rightarrow$ **small number of edges between two vertices**: diameter = 5 and 90% of the couples of vertices have less than 3 edges between them;

Previous work

In **[Boulet & Jouve, 2007]**, algebraic study of the non-weighted graph:

- **small world** graph $\Rightarrow$ **small number of edges between two vertices**: diameter = 5 and 90% of the couples of vertices have less than 3 edges between them;
- classification into **communities** (cliques having the same neighbors) and **rich club** (set of vertices having high degree and a high density):

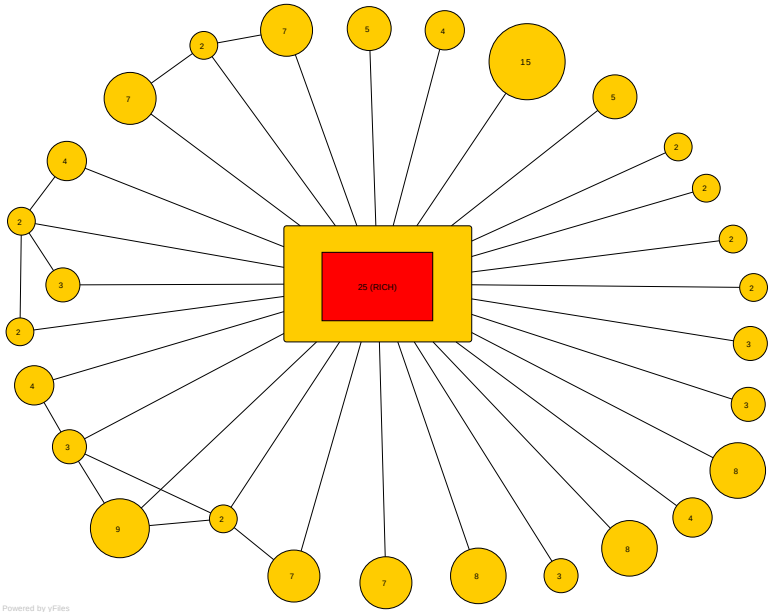


Table of contents

Dissimilarity SOM

[Kohohen & Somervuo, 1998], **[El Golli *et al.*, 2006]**: batch SOM for data $x_1, \ldots, x_n$ known through a dissimilarity measure δ:

1. **Initialization**: for all neurons of the grid, $j = 1, \ldots, M$, select a prototype (randomly), m_j in the dataset;
2. **Assignment**: for all data $i = 1, \ldots, n$, x_i is assigned to the neuron $f(x_i)$ such that
$$f(x_i) = \arg \min_{j=1,\ldots,M} \delta(x_i, m_j)$$
3. **Representation**: for all neuron $j = 1, \ldots, M$, m_j is re-computed:
$$m_j = \arg \min_{x \in (x_k)_{k=1,\ldots,n}} \sum_{i=1}^{n} h(f(x_i), j)\delta(x_i, x)$$
where h is a decreasing function of the distance between 2 neurons of the grid.

Dissimilarity SOM

[Kohohen & Somervuo, 1998], **[El Golli *et al.*, 2006]**: batch SOM for data $x_1, \ldots, x_n$ known through a dissimilarity measure δ:

1. **Initialization**: for all neurons of the grid, $j = 1, \ldots, M$, select a prototype (randomly), m_j in the dataset;
2. **Assignment**: for all data $i = 1, \ldots, n$, x_i is assigned to the neuron $f(x_i)$ such that
$$f(x_i) = \arg \min_{j=1,\ldots,M} \delta(x_i, m_j)$$
3. **Representation**: for all neuron $j = 1, \ldots, M$, m_j is re-computed:
$$m_j = \arg \min_{x \in (x_k)_{k=1,\ldots,n}} \sum_{i=1}^{n} h(f(x_i), j)\delta(x_i, x)$$
where h is a decreasing function of the distance between 2 neurons of the grid.

$\Rightarrow$ Choosing an appropriate δ.

Laplacian and diffusion kernel

Definitions

For a graph with vertices $V = \{x_1, \ldots, x_n\}$ having positive weights $(w_{i,j})_{i,j=1,\ldots,n}$ such that, for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and $d_i = \sum_{j=1}^{n} w_{i,j}$,

- **Laplacian**: $L = (L_{i,j})_{i,j=1,\ldots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{cases};$$

Laplacian and diffusion kernel

Definitions

For a graph with vertices $V = \{x_1, \ldots, x_n\}$ having positive weights $(w_{i,j})_{i,j=1,\ldots,n}$ such that, for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and $d_i = \sum_{j=1}^{n} w_{i,j}$,

- **Laplacian**: $L = (L_{i,j})_{i,j=1,\ldots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{cases};$$

- **Regularization: the diffusion matrix**: for $\beta > 0$, $K^{\beta} = e^{-\beta L}$.

Laplacian and diffusion kernel

Definitions

For a graph with vertices $V = \{x_1, \ldots, x_n\}$ having positive weights $(w_{i,j})_{i,j=1,\ldots,n}$ such that, for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and $d_i = \sum_{j=1}^{n} w_{i,j}$,

- **Laplacian**: $L = (L_{i,j})_{i,j=1,\ldots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{cases};$$

- **Regularization: the diffusion matrix**: for $\beta > 0$, $K^\beta = e^{-\beta L}$.

$\Rightarrow$

$$\begin{array}{rccc} k^\beta : & V \times V & \to & \mathbb{R} \\ & (x_i, x_j) & \to & K^\beta_{i,j} \end{array}$$

is the **diffusion kernel** (or heat kernel).

Properties

1. **Diffusion on the graph**: $k^{\beta}(x_i, x_j) \simeq$ quantity of energy accumulated in x_j after a given time if energy is injected in x_i at time 0 and if diffusion is done along the edges.
   $\beta \simeq$ intensity of diffusion;

Properties

1. **Diffusion on the graph**: $k^\beta(x_i, x_j) \simeq$ quantity of energy accumulated in x_j after a given time if energy is injected in x_i at time 0 and if diffusion is done along the edges.
   $\beta \simeq$ intensity of diffusion;
2. **Regularization operator**: for $u \in \mathbb{R}^n \sim V$, $u^T K^\beta u$ is higher for vectors u that vary a lot over "close" vertices of the graph.
   $\beta \simeq$ intensity of regularization (for small β, direct neighbors are more important);

Properties

1. **Diffusion on the graph**: $k^\beta(x_i, x_j) \simeq$ quantity of energy accumulated in x_j after a given time if energy is injected in x_i at time 0 and if diffusion is done along the edges.
   $\beta \simeq$ intensity of diffusion;
2. **Regularization operator**: for $u \in \mathbb{R}^n \sim V$, $u^T K^\beta u$ is higher for vectors u that vary a lot over "close" vertices of the graph.
   $\beta \simeq$ intensity of regularization (for small β, direct neighbors are more important);
3. **Norm of a Hilbert space**: k^β is symetric and positive $\Rightarrow \exists$ Hilbert space $(\mathcal{H}, \langle ., . \rangle)$ and $\phi : V \to \mathcal{H}$ such that

$$k^\beta(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

Properties

1. **Diffusion on the graph**: $k^\beta(x_i, x_j) \simeq$ quantity of energy accumulated in x_j after a given time if energy is injected in x_i at time 0 and if diffusion is done along the edges.
   $\beta \simeq$ intensity of diffusion;
2. **Regularization operator**: for $u \in \mathbb{R}^n \sim V$, $u^T K^\beta u$ is higher for vectors u that vary a lot over "close" vertices of the graph.
   $\beta \simeq$ intensity of regularization (for small β, direct neighbors are more important);
3. **Norm of a Hilbert space**: k^β is symetric and positive $\Rightarrow \exists$ Hilbert space $(\mathcal{H}, \langle ., . \rangle)$ and $\phi : V \to \mathcal{H}$ such that

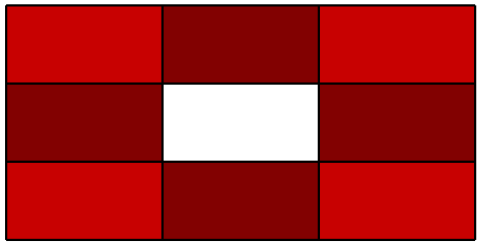
$$k^\beta(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

$\Rightarrow \delta(x_i, x_j) = \sqrt{k^\beta(x_i, x_i) + k^\beta(x_j, x_j) - 2k^\beta(x_i, x_j)}$ is a distance on $\mathcal{H} \rightsquigarrow$ dissimilarity between vertices of the graph.

Table of contents

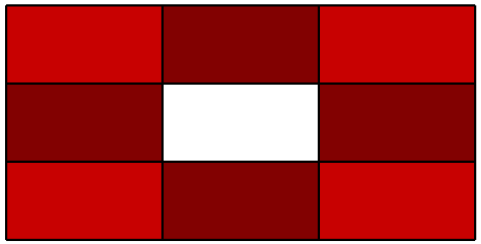
Methodology

1. **SOM**: 3×3 rectangular grid;



Methodology

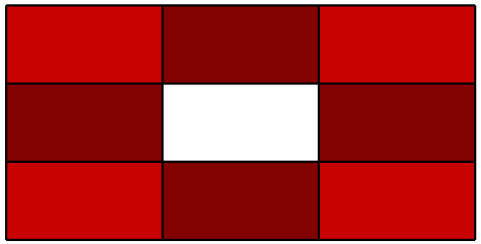
1. **SOM**: 3×3 rectangular grid;



2. **Dissimilarity measure**: $\beta = 0.05, 0.1, 0.2$;

Methodology

1. **SOM**: 3×3 rectangular grid;



2. **Dissimilarity measure**: $\beta = 0.05, 0.1, 0.2$;
3. **Initialization**: for each β,
   - 50 random initialization;
   - algorithm ran until stabilization;
   - best classification minimizes

$$\mathcal{E} = \sum_{j=1}^{M} \sum_{i=1}^{n} h(f(x_i), j)\delta(x_i, m_j).$$

Results

31 35 69 70 79 118 155 156 159 161 175 176 198 220 246 249 272 278 281 284 285 291 316 352 *354* 384 389 396 405 432 433 434 435 442 447 449	66 94 95 96 124 125 126 132 138 142 149 240 241 245 375 *398* 418 419	25 61 64 74 *144* 201 236 247 255 412
8 10 78 119 *127* 128 131 134 135 153 181 183 184 193 200 222 237 251 258 264 276 377 386 390 394 416 420	133 *139* 150 227 262 415	9 67 129 130 *136* 137 140 141 146 148 151 152 217 260 263 413
11 17 26 27 *28* 32 68 80 154 182 189 190 191 196 205 206 208 235 239 256 270 277 280 350 355 370 388 400 402 411 414	6 30 *33* 37 72 120 197 209 226 269 273 274 282 357 379 399 421	2 3 22 23 24 29 34 36 51 52 71 73 75 145 147 192 199 202 203 204 207 216 218 219 221 223 224 *225* 228 229 230 231 232 233 238 248 250 252 257 259 265 275 279 283 315 356 358 366 367 369 371 376 378 383 395 401 403 404 409 410 417 440 443 444

$\beta = 0.05$

6 28 30 *33* 37 68 72 80 120 182 190 197 209 226 256 269 270 273 274 282 357 379 399 414 421	3 61 64 74 191 201 204 220 228 230 232 233 247 255 257 259 272 280 283 376 378 396 *412*	11 25 26 27 75 *144* 154 227 239
8 10 78 95 96 119 125 126 *127* 128 131 132 134 135 142 149 153 181 183 184 193 200 237 241 245 251 258 264 276 377 386 390 394 398 416 419 420	66 94 124 133 138 *139* 143 150 240 262 375 418	9 67 *129* 130 136 137 140 141 146 148 151 152 217 260 263 413
31 35 69 70 79 118 155 156 159 161 175 176 198 222 246 249 278 281 284 285 291 316 352 354 384 389 405 432 433 434 435 442 447 449	17 24 145 189 196 208 235 277 370 402 411 *415*	2 22 23 29 34 36 51 52 71 73 147 192 199 202 203 205 206 207 216 218 219 221 223 224 *225* 229 231 238 248 250 252 265 275 279 315 350 355 356 358 366 367 369 371 383 388 395 400 401 403 404 410 417 440 443 444

$\beta = 0.1$

35 66 69 70 94 118 124 143 150 156 161 175 198 222 240 246 249 *262* 281 284 285 375 384 389 405 418 433 434 447 449	9 67 130 140 141 *148* 151 217 260	3 28 33 61 64 74 176 191 204 206 220 228 230 232 233 236 247 255 257 259 272 273 283 316 376 378 388 *412*
8 10 95 96 119 125 126 127 128 131 134 135 142 149 153 181 193 200 237 245 258 *264* 276 377 386 390 398 416 419 420	129 *136* 137 146 201 263 355 413	6 *30* 32 37 68 72 80 120 182 190 197 205 226 256 269 270 274 280 282 357 379 399 414 421
75 78 79 *132* 155 159 227 241 278 291 352 396 432 435	11 17 25 26 27 31 133 139 144 *152* 154 235 239 277 354 370 400 402 404 415 442	2 22 23 24 29 34 36 51 52 71 73 145 147 183 184 189 192 196 199 202 203 207 208 209 216 218 219 221 223 224 *225* 229 231 238 248 250 251 252 265 275 279 315 350 356 358 366 367 369 371 383 394 395 401 403 409 410 411 417 440 443 444

$\beta = 0.2$

Results

31 35 69 70 79 118 155 156 159 161 175 176 198 220 246 249 272 278 281 284 285 291 316 352 *354* 384 389 396 405 432 433 434 435 442 447 449	66 94 95 96 124 125 126 132 138 142 149 240 241 245 375 *398* 418 419	25 61 64 74 *144* 201 236 247 255 412
8 10 78 119 *127* 128 131 134 135 153 181 183 184 193 200 222 237 251 258 264 276 377 386 390 394 416 420	133 *139* 150 227 262 415	9 67 129 130 *136* 137 140 141 146 148 151 152 217 260 263 413
11 17 26 27 *28* 32 68 80 154 182 189 190 191 196 205 206 208 235 239 256 270 277 280 350 355 370 388 400 402 411 414	6 30 *33* 37 72 120 197 209 226 269 273 274 282 357 379 399 421	2 3 22 23 24 29 34 36 51 52 71 73 75 145 147 192 199 202 203 204 207 216 218 219 221 223 224 *225* 228 229 230 231 232 233 238 248 250 252 257 259 265 275 279 283 315 356 358 366 367 369 371 376 378 383 395 401 403 404 409 410 417 440 443 444

$\beta = 0.05$

6 28 30 *33* 37 68 72 80 120 182 190 197 209 226 256 269 270 273 274 282 357 379 399 414 421	3 61 64 74 191 201 204 220 228 230 232 233 247 255 257 259 272 280 283 376 378 396 *412*	11 25 26 27 75 *144* 154 227 239
8 10 78 95 96 119 125 126 *127* 128 131 132 134 135 142 149 153 181 183 184 193 200 237 241 245 251 258 264 276 377 386 390 394 398 416 419 420	66 94 124 133 138 *139* 143 150 240 262 375 418	9 67 *129* 130 136 137 140 141 146 148 151 152 217 260 263 413
31 35 69 70 79 118 155 156 159 161 175 176 198 222 246 249 278 281 284 285 291 316 352 354 384 389 405 432 433 434 435 442 447 449	17 24 145 189 196 208 235 277 370 402 411 *415*	2 22 23 29 34 36 51 52 71 73 147 192 199 202 203 205 206 207 216 218 219 221 223 224 *225* 229 231 238 248 250 252 265 275 279 315 350 355 356 358 366 367 369 371 383 388 395 400 401 403 404 410 417 440 443 444

$\beta = 0.1$

35 66 69 70 94 118 124 143 150 156 161 175 198 222 240 246 249 *262* 281 284 285 375 384 389 405 418 433 434 447 449	9 67 130 140 141 *148* 151 217 260	3 28 33 61 64 74 176 191 204 206 220 228 230 232 233 236 247 255 257 259 272 273 283 316 376 378 388 *412*
8 10 95 96 119 125 126 127 128 131 134 135 142 149 153 181 193 200 237 245 258 *264* 276 377 386 390 398 416 419 420	129 *136* 137 146 201 263 355 413	6 *30* 32 37 68 72 80 120 182 190 197 205 226 256 269 270 274 280 282 357 379 399 414 421
75 78 79 *132* 155 159 227 241 278 291 352 396 432 435	11 17 25 26 27 31 133 139 144 *152* 154 235 239 277 354 370 400 402 404 415 442	2 22 23 24 29 34 36 51 52 71 73 145 147 183 184 189 192 196 199 202 203 207 208 209 216 218 219 221 223 224 *225* 229 231 238 248 250 251 252 265 275 279 315 350 356 358 366 367 369 371 383 394 395 401 403 409 410 411 417 440 443 444

$\beta = 0.2$

Results

31 35 69 70 79 118 155 156 159 161 175 176 198 220 246 249 272 278 281 284 285 291 316 352 *354* 384 389 396 405 432 433 434 435 442 447 449	66 94 95 96 124 125 126 132 138 142 149 240 241 245 375 *398* 418 419	25 61 64 74 *144* 201 236 247 255 412
8 10 78 119 *127* 128 131 134 135 153 181 183 184 193 200 222 237 251 258 264 276 377 386 390 394 416 420	133 *139* 150 227 262 415	9 67 129 130 *136* 137 140 141 146 148 151 152 217 260 263 413
11 17 26 27 *28* 32 68 80 154 182 189 190 191 196 205 206 208 235 239 256 270 277 280 350 355 370 388 400 402 411 414	6 30 *33* 37 72 120 197 209 226 269 273 274 282 357 379 399 421	2 3 22 23 24 29 34 36 51 52 71 73 75 145 147 192 199 202 203 204 207 216 218 219 221 223 224 *225* 228 229 230 231 232 233 238 248 250 252 257 259 265 275 279 283 315 356 358 366 367 369 371 376 378 383 395 401 403 404 409 410 417 440 443 444

$\beta = 0.05$

6 28 30 *33* 37 68 72 80 120 182 190 197 209 226 256 269 270 273 274 282 357 379 399 414 421	3 61 64 74 191 201 204 220 228 230 232 233 247 255 257 259 272 280 283 376 378 396 *412*	11 25 26 27 75 *144* 154 227 239
8 10 78 95 96 119 125 126 *127* 128 131 132 134 135 142 149 153 181 183 184 193 200 237 241 245 251 258 264 276 377 386 390 394 398 416 419 420	66 94 124 133 138 *139* 143 150 240 262 375 418	9 67 *129* 130 136 137 140 141 146 148 151 152 217 260 263 413
31 35 69 70 79 118 155 156 159 161 175 176 198 222 246 249 278 281 284 285 291 316 352 354 384 389 405 432 433 434 435 442 447 449	17 24 145 189 196 208 235 277 370 402 411 *415*	2 22 23 29 34 36 51 52 71 73 147 192 199 202 203 205 206 207 216 218 219 221 223 224 *225* 229 231 238 248 250 252 265 275 279 315 350 355 356 358 366 367 369 371 383 388 395 400 401 403 404 410 417 440 443 444

$\beta = 0.1$

35 66 69 70 94 118 124 143 150 156 161 175 198 222 240 246 249 *262* 281 284 285 375 384 389 405 418 433 434 447 449	9 67 130 140 141 *148* 151 217 260	3 28 33 61 64 74 176 191 204 206 220 228 230 232 233 236 247 255 257 259 272 273 283 316 376 378 388 *412*
8 10 95 96 119 125 126 127 128 131 134 135 142 149 153 181 193 200 237 245 258 *264* 276 377 386 390 398 416 419 420	129 *136* 137 146 201 263 355 413	6 *30* 32 37 68 72 80 120 182 190 197 205 226 256 269 270 274 280 282 357 379 399 414 421
75 78 79 *132* 155 159 227 241 278 291 352 396 432 435	11 17 25 26 27 31 133 139 144 *152* 154 235 239 277 354 370 400 402 404 415 442	2 22 23 24 29 34 36 51 52 71 73 145 147 183 184 189 192 196 199 202 203 207 208 209 216 218 219 221 223 224 *225* 229 231 238 248 250 251 252 265 275 279 315 350 356 358 366 367 369 371 383 394 395 401 403 409 410 411 417 440 443 444

$\beta = 0.2$

31 35 69 70 79 118 155 156 159 161 175 176 198 246 249 278 281 284 285 291 316 352 354 384 389 405 432 433 434 435 442 447 449		3 61 64 74 191 204 220 228 230 232 233 236 247 255 257 259 272 283 376 378 *412*
8 10 119 *127* 128 131 134 135 153 181 193 200 237 258 264 276 377 386 390 416 420	133 *139* 150 262	9 67 129 130 136 137 140 141 146 148 151 152 217 260 263 413
	6 30 37 72 120 197 226 269 274 282 357 379 399 421	2 22 23 29 34 36 51 52 71 73 147 192 199 202 203 207 216 218 219 221 223 224 *225* 229 231 238 248 250 252 265 275 279 315 356 358 366 367 369 371 383 395 401 403 410 417 440 443 444

Synthesis map

Comparison with previous work

Class 6		Class 7
Class 4	Class 3	Class 2
	Class 5	Class 1

Comparison with previous work

Class 6		Class 7
Class 4	Class 3	Class 2
	Class 5	Class 1

Similarities with historian's prior knowledge

- Class 1 & Class 2 : homogeneous geographical settings (Castrum de Flaugnac & Castrum de La Graulière);
- But "Combelcau" family is not in Class 1 although part of them live in Castrum de Flaugnac;
- "Combelcau" family constitutes a large part of Class 2.

$\Rightarrow$ Family links are more important than geographical ones.

Comparison with previous work

Class 6		Class 7
Class 4	Class 3	Class 2
	Class 5	Class 1

Similarities with algebraic work on the non-weighted graph

- Each class corresponds to one or several connected communities;
- Class 3 is a part of the “rich club” and “Combelcau” family plays a great role both in Class 3 and in the rich club;
- Star-shaped structure around class 3 is similar to star-shaped structure around the rich club.

Table of contents

Conclusion

Expected developments:

- Application to the whole database;
- Find a criterium to select an optimal β;
- Find an automatic procedure to combine/compare several maps obtained for different β.

References

Boulet, R. & Jouve, B. (2007).
Partitionnement d'un réseau de sociabilité à fort coefficient de clustering.
In *7èmes Journées Francophones "Extraction et Gestion des Connaissances"* 569–574.

El Golli, A., Rossi, F., Conan-Guez, B. & Lechevallier, Y. (2006).
Une adaptation des cartes auto-organisatrices pour des données décrites par un tableau de dissimilarités.
Revue de Statistique Appliquée, **LIV**(3), 33–64.

Kohohen, T. & Somervuo, P. (1998).
Self-Organizing maps of symbol strings.
Neurocomputing, **21**, 19–30.