Bachelor Thesis

PCCA+ and its application to spatial timeseries clustering

Free University of Berlin
Department of Mathematics and Computer Science

Alexander Sikorski
Supervisor: PD Dr. Marcus Weber

March 18, 2015

Contents

1 Introduction

In this thesis we develop an algorithm for clustering spatial timeseries into a prescribed number of clusters, based on their spatial and dynamical properties.

After an introduction to the underlying theoretical background we will review known results about the Perron Cluster Cluster Analysis (PCCA+), which forms the basis for our application. PCCA+ will allow us to identify metastable clusters in Markov Chains, that is a configuration of the system which is likely to persist for a longer time. In the course we will extend the known results by a stochastic interpretation for the propagator matrix, which encodes the time evolution on the clusters.

We will then show a method to turn spatial timeseries into a Markov Chain to obtain a spatial clustering by further application of PCCA+, respecting the dynamic information. Finally we will apply that method to data obtained by tracking human eye fixations while these look at different paintings to detect the depicted objects. This can be seen as a form of object recognition which does not rely on the image data itself but identifies the objects based on the humans recognition reflected in their eye movement.

2 Theoretical background

2.1 Introduction to Markov chains

Let S be any finite set, i.e. $S =: \{s_1, ..., s_N\}$. A Markov chain on S is a stochastic process, consisting of a sequence of random variables $X_i : \Omega \to S$, $i \in \mathbb{N}$ satisfying the Markov property:

$$P(X_{t+1} = x | X_1 = x_1, X_2 = x_2, ..., X_t = x_t) = P(X_{t+1} = x | X_t = x_t) \, \forall t \in \mathbb{N}.$$

It is common to interpret S as the state space of possible outcomes of measurements at timestep t represented by X_t. The Markov property assures, that the transition probabilities to the next timestep x_{t+1} only depend on the current state x_t. This means that the process at time t has no memory of its previous history $(x_1, ..., x_{t-1})$, this is also sometimes called the memoryless property.

We will furthermore assume that the process is autonomous, i.e. not explicitly depending on the time:

$$P(X_{t+1} = x | X_t = y) = P(X_t = x | X_{t-1} = y) \forall t \in \mathbb{N}.$$

This does not really impose a restriction since any non-autonomous process can be turned into an autonomous one: By adding all possible times to the state space S using the cartesian product $S' := \mathbb{N} \times S$ the explicit time-dependence of the process on S can be implicitly subsumed by an autonomous process on S'.

Since S is finite we can, enumerating all states in S, encode the whole process in the right stochastic *transition matrix*

$$P_{ij} := P\left(X_{t+1} = j | X_t = i\right),$$

where row stochastic means that each row has rowsum one and propagation of states is realised by right application of P.

A *stationary distribution* is a row vector π, satisfying

$$\pi P = \pi.$$

Although we only consider a discrete state space in this thesis, the results are extendible to continous state spaces as well. The easiest way is using a set-based discretization, dividing the state space into a finite mesh of subsets. For high dimensional state spaces, as for example met in molecular dynamics, this approach exhibits the curse of dimensionality, as the size of the mesh grows exponentially with the dimensions. Weber developed a meshless version of PCCA+ using a global Galerkin discretization[10] as solution to this problem .

2.2 Clustering of the state space

As the goal of PCCA+ is to reduce the complexity of analysis of the markov chain by a dimension reduction we will now introduce the concept of clustering, that is subsuming different states of the state space to a smaller set of $n \in \mathbb{N}$ clusters $C := \{1, ..., n\}$.

The simplest possibility is assigning each state $k \in S$ to a cluster $i \in C$, which can be encoded by means of the *characteristic vector* $\chi_i \in \{0, 1\}^N$:

$$\chi_{i,k} = \begin{cases} 1, & \text{if state } k \text{ belongs to cluster } i \\ 0, & \text{else} \end{cases}.$$

Due to its discrete nature, this *crisp clustering* approach, used by *Perron Cluster Analysis* (PCCA) of Deuflhard et al. [1], has the disadvantage of not being robust against small pertubations, as continuous changes in P finally result in discontinous changes in the clustering.

Weber and Galliat, Deuflhard and Weber therefore developed a robust version, *Robust Perron Cluster Analyis* (PCCA+), by making use of a *fuzzy clustering* representing each cluster by an *almost characteristic vector*

$$\chi_i \in [0, 1]^n. \tag{2.1}$$

Almost characteristic vectors $\{\chi_i\}_{i=1}^n$ satisfying the partition of unity property

$$\sum_{i=1}^{n} \chi_i = 1 \tag{2.2}$$

are called *membership vectors* as they describe the relative membership of each state to each cluster. We will refer to the matrix collection $\chi := (\chi_i)_{i=1}^n \in \mathbb{R}^{N \times n}$ of the *membership vectors* as a *clustering*, whereas in the field of computational chemistry it is also referred to as *conformations*.

2.3 Galerkin projection of the transition matrix

The coupling Matrix

To represent the dynamics on the reduced/clustered state space in the case of a *crisp clustering* χ, i.e. $\chi_i \in \{0,1\}$, Deuflhard et al. [1] introduced the *coupling matrix*

$$W_{ij} := \frac{\langle \chi_j, P\chi_i \rangle_\pi}{\langle \chi_i, 1 \rangle_\pi} = \frac{\chi_j^T D_\pi P \chi_i}{\pi^T \chi_i},$$

or in matrix notation

$$W := \text{diag}\left(\chi^T \pi\right)^{-1} \chi^T D_\pi P \chi.$$

The entries W_{ij} can thus be interpreted as conditional transition probabilities from cluster i to cluster j, given the starting distribution π.

In the *fuzzy clustering* setting the problem arises, that it is not clear anymore which state belongs to which cluster. It is therefore convenient to interpret the membershipership of state j to cluster χ_i, χ_{ij}, as probability of measuring state j belonging to cluster χ_i. Then W_{ij} denotes the expectation value for measuring cluster χ_j after propagating the density given by χ_i.

Note, however, that even if no real transitions are actually happening in the state space, we still may count transitions between clusters, since we once measure the same state belonging to one and then to another cluster, as demonstrated in example 3.

One of the main motivations for developing PCCA+ was the wish to identify so called *metastable conformations* of molecular systems, e.g. to analyse the effectivity of active pharmaceutical ingredients in Computional Molecular Design (for a overview over this approach see [4]). These *conformations* are *almost invariant aggregates* of states, i.e. *membership vectors* with high self-transition probabilities, guaranteeing that the system resides in these states on longer timescales.

This can be formalized, as proposed by Huisinga [3], by the definition of the *metastability* of *membership vectors* as the trace of the corresponding *coupling matrix*: $\text{tr}\,(W)$. Note, that this does not need to correspond with a high probability of the cluster.

The propagator matrix

Unfortunately, the projection via the *coupling matrix* does not commute with time propagation, and therefore cannot be used for long term analyses of the underlying markov chain.

As remedy Kube and Weber [6] proposed the *coarse propagator matrix*

$$P_C := \left(\chi^T D_\pi \chi\right)^{-1} \chi^T D_\pi P \chi, \tag{2.3}$$

which coincides with the *coupling matrix* W in the *crisp clustering* setting. Assuming that χ is a linear combination of vectors spanning a P-invariant subspace satisfying an invertibilty condition it has the advantage that discretization via χ and time propagation commute, i.e.

$$P\chi = \chi P_C. \tag{2.4}$$

This property ensures that the *coupling matrix* represents the right dynamics of the underlying markov chain on the reduced state space, even for iterative application, i.e. $P^n\chi = \chi P_C^n$.

Theorem 1. *Let $\chi = XA$, $X \in \mathbb{R}^{N\times n}$, $A \in \mathbb{R}^{n\times n}$ satisfying the subspace condition*

$$PX = X\Lambda \tag{2.5}$$

for some $\Lambda \in \mathbb{R}^{n\times n}$ and $C := X^T D_\pi X$ be invertible.

Then the P_C is conjugate to Λ and discretization-propagation commutativity 2.4 holds.

Proof. We calculate

$$\begin{aligned} P_C &= \left(\chi^T D_\pi \chi\right)^{-1} \chi^T D_\pi P \chi \\ &= \left(A^T C A\right)^{-1} A^T C \Lambda A \\ &= A^{-1} C^{-1} A^{-T} A^T C \Lambda A \\ &= A^{-1}\Lambda A, \end{aligned} \tag{2.6}$$

which implies

$$P\chi = PXA = X\Lambda A = XAA^{-1}\Lambda A = \chi P_C.$$

□

This form of the theorem constitutes a small generalization towards its so far published versions, where instead of invertibility $C = Id$ was assumed.

Stochastic interpretation

Due to the matrix inversion the *propagator matrix* can have negative entries, as shown in example 3 below, thus prohibiting a natural stochastic interpretation.

We will therefore shed some light into the connection between the *coupling-* and the *propagator matrix*, making use of the notation of the *restriction-* and *interpolation operators*, introduced by Kube and Weber [6]:

$$\begin{aligned} R : \mathbb{R}^N &\to \mathbb{R}^n,\ x \mapsto x\chi \\ I : \mathbb{R}^n &\to \mathbb{R}^N,\ x \mapsto x\tilde{D_\pi}^{-1}\chi^T D_\pi \end{aligned}$$

with $D_{\tilde\pi} = \mathrm{diag}\,(\tilde\pi)$ and $\tilde\pi = \pi R$, where we apply them from the right in line with the used notation of right-stochastic matrices.

These provide the transformations between the (fine-grained) configuration space and the (coarse-grained) cluster space, being natural in the sense that $IRw = w$, i.e. I reconstructs the fine-grained density, lost by the restriction R, using the fine-grained stationary density.

This allows to reformulate the *coupling-* and *propagator matrix as*

$$\begin{aligned} W &= IPR \\ P_C &= (IR)^{-1} IPR. \end{aligned}$$

Now, consider the situation when setting $P = \mathrm{Id}$ with a fuzzy clustering. Then $W = IR = \tilde{D_\pi}^{-1}\chi^T D_\pi \chi \neq \mathrm{Id}$ as different clusters overlap. The result corresponds to the transitions which are introduced to the coarser system due to the overlap.

As this overlap would be applied on every iteration of W it would lead to increased mixing between the states, resulting in wrong long-term results. P_C grants the desired commutativity 2.4 by factoring out these transitions.

But this also shows us how we can compute a corresponding stochastically interpretable *coupling matrix* for larger times corresponding to n iterations, W_n, from the smaller matrix P_c:

$$W_n := IP^n R = IRP_C^n.$$

2.4 Example Processes

To demonstrate the connections between the different projections we now will show some example systems.

Example 1: The decoupled system

Consider

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In this ideal decoupled system we have two invariant subspaces spanned by the so called *Perron eigenvectors* with eigenvalue 1, the vectors

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

These can be interpreted as a *crisp clustering*, and assuming an equidistributed starting distribution we can compute the, in the *crisp* case coinciding, matrices

$$W = P_C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Example 2: The 3-pot

Next, we will consider the stationary markov chain on three states with a *fuzzy clustering*:

$$P := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \chi = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}, \pi = \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

According to our definitions we now compute

$$P_C := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, W = \frac{1}{6}\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}.$$

We thus observe, that P_C contains the expected stationary dynamics on the reduced state space, while W accounts for the possible transitions of observing the second state once in cluster 1 and once in cluster 2, due to the overlap in the *clustering*.

Example 3: Negative entries

Let us now consider

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \chi = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \pi = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}.$$

Computation of the propagator matrix now leads to negative entries:

$$P_C = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 3 & -1 \end{pmatrix}.$$

Let us first compute the propagation of the normalized density corresponding to χ_2:

$$(0,1) \cdot P = (1,0),$$

i.e. s_2 is propagated to s_1. The corresponding computation on the cluster space is

$$(0,1) \cdot P_C = \frac{1}{2}(3,-1).$$

Here, the negative entry amounts for the overlap of the clusters and is necessary to encode state s_1: As both clusters hold an amount of s_2 we use a linear combination to eliminate this. We thus may interpret P_C acting to the basis of clusters, eliminating the overlap.

We can calculate the corresponding density on the state space by applying the interpolation operator

$$I = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \end{pmatrix}, \frac{1}{2}(3,-1) \cdot I = (1,0).$$

3 PCCA+

In this section we will construct the *Robust Perron Cluster Analyis* algorithm, introduced in [2]. We first will construct the matrix X spanning the required invariant subspace and examine the possible linear transformations A mapping these to a set of membership vectors. Finally we propose different objectives to an optimization problem to specify a "good" solution.

Unlike previous treatises we will not restrict ourselves to reversible processes and provide a more general version of PCCA+ using the Schur decomposition thus reflecting the newest developments in [7].

Note, that we impose a fixed cluster number n. An overview over methods for estimating the cluster number, based on different criteria, is given by Röblitz, Weber[9].

PCCA+ will construct the clusters, described by the *membership vectors*, as a linear combination of eigenvectors. This guarantees that χ spans an invariant subspace, thus leading to preservation of the slow time-scales. By choosing the $n < N$ eigenvectors with the largest eigenvalues one hopes to preserve the principal dynamics of P. A good indicator towards this goal is that eigenvectors with high eigenvalues represent conformations with a high degree of self-mapping. Deuflhard et al. [1] have furthermore shown that the desired metastability is bounded from above by the sum of the chosen eigenvalues, and for ϵ-pertubations of the coupling of uncoupled markov chains also from below by $\sum \lambda_i - O\left(\epsilon^2\right)$, justifying the choice of high eigenvalues.

3.1 Subspace construction

For the construction of the invariant subspace we make use of the *real Schur decomposition*, decomposing a matrix $P = QTQ^{-1}$ into a orthonormal matrix Q, whose columns are called the *Schur vectors,* and an upper quasi-triangular (1-by-1 and 2-by-2 blocks on its diagonal) matrix T, called the *Schur form*. The columns of Q are called the Schur vectors of P. The eigenvalues of P appear on the diagonal of T, where complex conjugate eigenvalues correspond to the 2-by-2 blocks. We then reorder the *Schur form* and recompute the corresponding *Schur vectors* such that the upper left $n \times n$ block of T contains the n largest eigenvalues by a reordering algorithm [5]. Note, that in the case of complex conjugate eigenvalues we have to select or discard the whole 2-by-2 blocks. Define Λ as the upper left $n \times n$ block of T and X as the n first columns of Q. These now satisfy the *subspace condition* for 1 by definition.

3.2 The feasible transformation set

Given the invariant subspace X, we will now examine the set of feasible matrices $F_A \subset \mathbb{R}^{n \times n}$ for the transformation A, leading to actual *membership vectors* $\chi := XA$. As P is stochastic the constant one vector is mapped to itself,

$$P \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix},$$

and thus forms an eigenvector to eigenvalue one, therefore $X_{i,1} = 1$, $i = 1, ..., N$. Thus, one can reformulate the positivity 2.1 and partition of unity 2.2 conditions in terms of

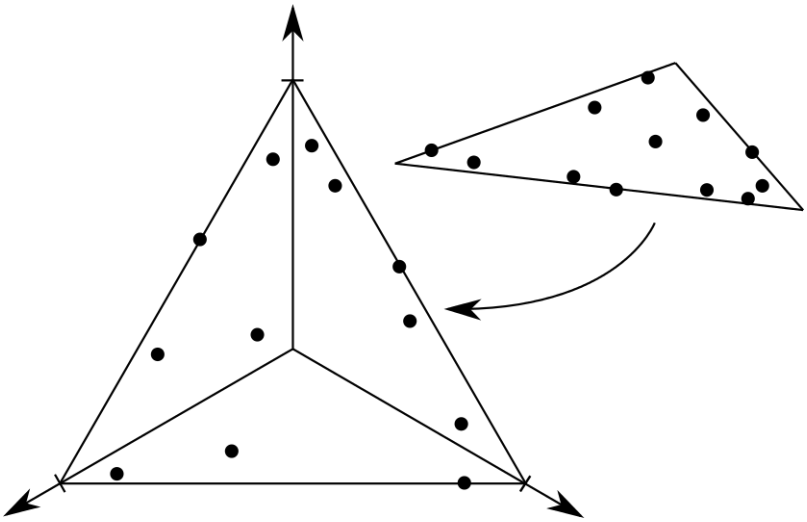

Figure 3.1:
Schematic illustration of the linear transformation mapping the row vectors of the eigenvectors (points on the $z = 1$ hyperplane) onto the standard 2-simplex.

the matrices X and A, leading to the following constraints for A:

$$A_{1,j} \geq -\sum_{k=2}^{n} X_{ik} A_{kj},\ i = 1, ..., N,\ j = 1, ..., n \text{ (positivity)}, \tag{3.1}$$

$$A_{i,1} = \delta_{i,1} - \sum_{j=2}^{n} A_{ij},\ i = 1, ..., n \text{ (partition of unity)} \tag{3.2}$$

Since these constraints are linear in A the set F_A is a convex polytope.

Geometrically one might think of the N rows of the matrix χ as points in the space $\mathbb{R}^n$. The positivity and 2.1 and partition of unity 2.2 conditions force these points to lie on the standard $(n-1)$-simplex Δ. Now, if $\chi = XA$ this means that the matrix A maps the the N rows of the eigenvector matrix X to that simplex. As we have seen the first component of each row is 1, thus all rows lie on the hyperplane with first component 1. They furthermore are contained in a bounded region, and thus we can map them linearly onto Δ via a linear map A.

Thus we directly see that the set F_A is not empty. It furthermore is infinite, as we can always shrink the image further and it will still fit into Δ.

3.3 Optimization

As we have the choice between infinitely many possible solutions for $A \subset F_A$ we will now specify and motivate different optimization objectives, to specify a specific solution by the optimization problem.

Maximal scaling condition

Assuming (*maximality assumption)* that the convex hull co (X) of the rows of X already has the form of an $(n-1)$-simplex, we can now choose A uniquely (up to permutation) to map this exactly onto Δ, which among all the ways of mapping X into Δ gives us the highest distinctiveness between the resulting clusters. This assumption is equivalent to the situation that for each corner there exists a row getting mapped into that corner, i.e.

$$\max_{i=1..N} \chi_{ij} = 1,\ j = 1, ..., n,$$

justifying its name.

As in general the *maximality assumption* is not met, it seems natural to turn it into an optimization problem. This has been done in [2, 10] by imposing maximization of the *maximal scaling condition*

$$I_1(A) := \sum_{j=1}^{n} \max_{i=1..N} \chi_{ij} \leq n_C.$$

Assuming that the *maximality assumption* is almost met, i.e. $\max_{i=1..N} \chi_{ij} \approx 1,\ j = 1, ..., n$, Weber [10] proposes to determine the maximizing indices by the *index mapping algorithm*2, turning this convex optimization problem into a linear one:

$$I_1(A) = \sum_{i,j=1}^{n} X_{ind(X)_j, i} A_{ij}$$

In [10] Weber furthermore shows that $W_{jj} \leq \max_{i=1..N} \chi_{ij}$, which implies that I_1 is an upper bound for the metastability, which thus should be large.

Note, that this objective, ignoring the datapoints not beeing the maxima, cannot distinguish between differences in the interior of the convex hull, leading to possibly non-optimal transformation matrices A, as illustrated in 3.2.

Maximal metastability condition

Another choice might be optimizing directly towards a maximal metastability, as done by Deuflhard and Weber [2, 10]

$$I_2(A) := \text{trace}(W) = \sum_{i=1}^{n} \lambda_i \sum_{j=1}^{n} \frac{A(i,j)^2}{A(1,j)},$$

where they establish the latter equation making use of $\pi_i = A_{1,i}$ ([10], Lemma 3.6).

Crispness objective

Röblitz [9] argues that the stochastic interpretation of W is not valid in the fuzzy setting, due to the overlap. Optimization of the trace of P_C makes no sense as it is similar to

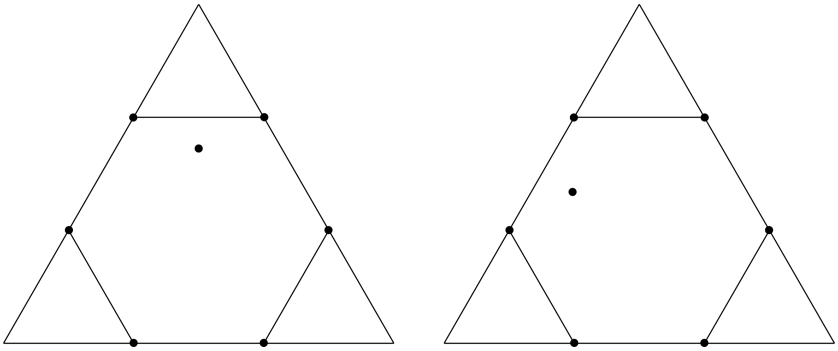

Figure 3.2: Mapping of 7 rows of the eigenvectors (affine hexagon with an interior point) to a 2-simplex. While I_1 cannot differentiate between the two mappings, I_2 will choose the second as it provides a crisper assignment of the interior point.

Λ2.6 and therefore independent of A. She therefore suggests maximization of

$$I_3 := \text{trace}\,(IR) = \sum_{i,j=1}^{n} \frac{(A_{ij})^2}{A_{1,j}}.$$

which is similar to the *maximal metastability condition*, with P replaced by the identity.

Maximizing the trace minimizes the off-diagonal entries of IP leading to the least amount of clustering-induced transitions and therefore to a as crisp as possible clustering.

Unconstrained Optimization

Due to the high number of inequality constraints 3.1 solving these linear or convex problems may still be very time consuming. Following Deuflhard and Weber [2, 10] we will now show how to turn this constrained into an unconstrained optimization problem by enforcing the constraints after each iteration.

Define the set $F_A^{'}$ by the equality constraints

$$\begin{aligned} A_{i,1} &= \delta_{i,1} - \sum_{j=2}^{n} A_{ij},\ i = 1, ..., n \\ A_{1,j} &= -\min_{l=1,...,N} \sum_{i=2}^{n} X_{li} A_{ij},\ j = 1, ..., n. \end{aligned} \tag{3.3}$$

Comparing these equalities to 3.1 one easily checks that $F_A^{'} \subset F_A$.

Now consider the following *feasibilization algorithm* $F : \mathbb{R}^{(n-1)\times(n-1)} \twoheadrightarrow F_A^{'}$, mapping any arbitrary matrix $\left(\tilde{A}_{ij}\right)_{i,j=2,...,n}$ to a feasible transformation matrix A and thus enforcing the desired constraints:

Algorithm 1: Feasibilization algorithm

1. For $i = 2, ..., n$ define $\tilde{A}_{i,1} := -\sum_{j=2}^{n} \tilde{A}_{ij}$
2. For $j = 1, ..., n$ define $\tilde{A}_{1,j} := -\min_{l=1,...,N} \sum_{i=2}^{n} X_{li} \tilde{A}_{ij}$
3. For $i, j = 1, ..., n$ define $A_{ij} := \frac{\tilde{A}_{ij}}{\sum_{j=1}^{k} \tilde{A}_{1,j}}$
4. Return A

Steps 1 and 2 guarantee feasibility of $\tilde{A}$ with respect to 3.3 for $i = 2, ..., n$ respectively $j = 1, ..., n$. As these equalities are linear in A they are invariant under scalar multiplication and step 3 now furthermore assures the equality 3.3 for $i = 1$. Thus F indeed maps to $F_A^{'}$. Furthermore, taking any matrix $A \in F_A^{'}$, dropping the first row and column to get $\tilde{A}$ and computing $F\left(\tilde{A}\right) = A$ we see that F is surjective.

As any objective function I_i, $i = 1, 2, 3$ is convex over F_A it attains its maximum at one of the vertices $v(F_A)$, which are contained in $F_A^{'}$ (for a proof see [10], Lemma 3.5). Thus, we can also optimize the function $F \circ I_i$ over $\mathbb{R}^{(n-1)\times(n-1)}$ and have thereby transformed the constrained optimization problem in n^2 unknows to an unconstrained in $(n-1)^2$ unknows.

As the *feasibilization algorithm* is not differentiable, Deuflhard and Weber [2] propose the use of the nonlinear simplex method of Nelder and Mead [8] as local optimization routine.

Initial guess

Based on Weber and Galliat [11] we outline the *inner simplex algorithm*, determining an initial guess for the matrix A by constructing a simplex surrounding all row-points and then computing the transformation to the standard simplex, thus turning the global into a local optimization problem.

The first step, the *index mapping algorithm,* looks for the indices i_j of the succesively farthest linear independent rows. It starts by choosing the largest row vector as starting point, and then iteratively adds the points with the largest distance to the hyperplane spanned by the chosen points so far:

The *inner simplex algorithm* uses these indices to construct of the n extremal points to construct the matrix A mapping these to the vertices of Δ.

In the case of the *maximality assumption*, X spans a $(n-1)$-simplex, and the *index mapping algorithm* determines its vertices, thus $\text{co}(AX) = \Delta$ and $A \in v(F_A)$ maximizes I_1.

For the general case though Weber [10] (Lemma 3.13, Theorem 3.14) has shown that the following statements are equivalent:

1. The convex hull $\text{co}(X)$ of X is a simplex.

Algorithm 2: Index mapping algorithm

1. Find starting point: $i_1 := \text{argmax}_{j \in C} \|X_{\cdot,j}\|_2$
2. Translate to origin: For $i \in S$ set $X_{i,\cdot} \leftarrow\!\shortmid X_{i,\cdot} - X_{i_1,\cdot}$
3. For $j = 2, ..., n$
   a) Find index of the farthest point: $i_j := \text{argmax}_{j \in C} \|X_{\cdot,j}\|_2$
   b) Projection to hyperplane by Gram-Schmidt process: $X \leftarrow\!\shortmid X - \frac{\mathrm{X} X_{i_j,\cdot}^T \otimes X_{i_j,\cdot}}{\left\|X_{i_j,\cdot}\right\|_2}$
4. Return the indices i_j

Algorithm 3: Inner simplex algorithm

Return $A(X) := (X_{ij})^{-1}_{i=i_1,...,i_n,\, j=1,...,n}$ with $i_1, ..., i_n$ computed by the *index mapping algorithm*

2. The result of the *inner simplex algorithm* is feasible, i.e. $A \in F_A$.
3. $A \in v(F_A)$ and therefore maximizes I_1.

Therefore, the result is not feasible in the generic case. If however the *maximality assumption* almost holds, i.e. the convex hull of X is a small pertubation of a simplex, which according to Weber [10] (3.4.4) is satisfied in many applications, the algorithm still gives a solution near the unperturbed solution. Thus A is near a vertex of the set F_A and thus a good initial guess for a local optimization of the unconstraint optimization.

3.4 The PCCA+ algorithm

Once we have decided on an objective function we can now put together all steps:

Extension to time-continous markov chains

PCCA+ is also applicable to the clustering of time-continuous markov chains (c.f. [6]). In that case the *transition matrix* P gets replaced by a *transition rate matrix* Q, having row-sum zero and nonnegative off-diagonal entries. The transitions after a fixed time can then be computed by

$$P(t) = e^{tQ}.$$

In that case the eigenvectors of P and Q are the same and the eigenvalues of P are the exponential of the corresponding eigenvalues of Q. As the exponential is monotone the eigenvectors with highest absolute eigenvalue, near 1, of P correspond to the eigenvectors with smallest absolute value, near zero, of Q.

Algorithm 4: PCCA+

1. Given P, compute X, Λ as in 3.1
2. Determine the, in general infeasible, initial guess $A_0 := A(X)$ by 3.
3. Perform an iterative local optimization A_0, A_1, ... of the objective function I_1, I_2 or I_3. In each step $A_k \to A_{k+1}$ only update the elements $A_{k,ij}$, $i, j \neq 1$ without constraints, then use algorithm *1* to get a feasible matrix A_k before evaluating the corresponding objective function.
4. Return the optimal transformation A

So by selecting the eigenvectors with smallest eigenvalue of Q we can compute the corresponding clustering for time-continuous markov cains.

4 Application to eyetracking data

This algorithm was applied to experimental eye-tracking data obtained by the department of psychology of the Universität Potsdam, with the goal to detect objects as metastable clusters using just the dynamics of the human eye, i.e. without any data of the image itself, and thus provides a way of interpreting the humans object recognition expressed through the eye movements.

4.1 The experiment and model

A group of test persons was presented different pictures for about 10 seconds, during which an eye-tracker measured their eye-fixations $f_i \in \mathbb{R}^2$ and their respective durations $t_i \in \mathbb{R}$. For subsequent analysis it is necessary to group different areas of the image into *areas of interest (AOE)* which correspond to subjectively identified objects in the corresponding picture.

To apply PCCA+ we need to turn this spatial timeseries into a markov chain.

We model each fixation as a *random choice* on a spatial grid weighted by a gaussian of the distance to the grid points, and then construct a *markov chain* by counting the induced transitions on the grid points. Assuming that humans, when looking at the pictures, dont jump randomly between all recognized objects but remain for some fixations inside one *AOE*, this behaviour should recur as high metastability of a clustering, corresponding to the *AOEs*.

4.2 Implementation

As state space we choose a spatial grid $S := \{s_i\}$, where the natural choice is using all fixation coordinates as grid ($s_i = f_i$) or alternatively use some spatial clustering

algorithm (e.g. k-means) to reduce the computational effort of the following PCCA+ routine.

Introducing a parameter σ, we assign a membership of each fixation to each grid point weighted by a gaussian of the distance between them, i.e. for each fixation f_i and each state s_j:

$$M_{ij} := \frac{e^{\frac{\left|f_i - s_j\right|^2}{2\sigma^2}}}{\sum_j e^{\frac{\left|f_i - s_j\right|^2}{2\sigma^2}}}$$

This assures that nearby fixations "overlap", adopting the metric information contained in the fixation data to the markov process. Thus the parameter σ, scaling the distance between points, can be interpreted as a spatial coupling constant.

We then choose a fixed time step $\Delta\tau$ as grid size for the time discretization, along which we count the transitions between the states weighted with the the corresponding fixation transitions, and row-normalize it to generate a *transition matrix*. In detail, for the transitions from state i to j, we have

$$P_{ij} \quad = \quad \frac{\sum_{s=0} M_{f_s,i} M_{f_{s+1},j}}{\sum_{s=0} M_{f_s,i}},$$

where f_s denotes the current fixation at time $s\Delta\tau$.

Alternatively we can also generate a *rate matrix* corresponding to a time-continuous markov model. In that case we estimate the transition rate from state i to j using the most likelihood estimator of the exponential distribution, the inverse of the expected transition time, which scales inversely to the membership, i.e.:

$$W_{ij} = \left(\frac{\sum_{a \to b} \frac{\tau_{a \to b}}{M_{ai} M_{bj}}}{\sum_{a \to b} 1} \right)^{-1},$$

denoting by $\sum_{a \to b}$ the sum over all fixation transitions, from a to b, and $\tau_{a \to b}$ the corresponding transition time.

Once we have constructed P this way we now compute the invariant eigenspace using the weighted Schur decomposition as in [todo] and pass is to PCCA+, which in return gives us the fuzzy clustering χ.

As a final step we discretize this fuzzy clustering by assigning to each state s_i the cluster c_i with the maximal share:

$$c_i = \text{argmax}_j \chi_{ij}. \tag{4.1}$$

Note, that choosing this discretization of the fuzzy clustering, some clusters may never be assigned, when beeing dominated by other clusters on every grid point.

In case of preclustering via k-means, the cluster assignment of the grid is passed to the corresponding fixations according to the k-means assignments.

The program was implemented in Julia and the source will be published under https://github.com/kliegl/

4.3 Choice of the parameters

The desired number of clusters, n, was chosen near the number of objects recognized by the experimentator. This, of course, is a subjective choice, but the number of clusters in general depends on the desired resolution of the clustering and thus on the further application. For example imagine a picture of a bookshelf with books, here one might recognize either the whole shelf, the books, or their titles as objects.

The time step size $\Delta\tau$ should be chosen as large as possible without skipping too many transitions. If it is chosen too large some fixations will be skipped resulting in loss of information and thus leading to a worse clustering. If on the other hand chosen too small we count one fixation as multiple self-transitions, thus weakening the effect of the real transitions, favouring the spatial over the dynamic information.

The parameter σ introduces the spatial informations and can thus be considered as a weight between dynamic and spatial clustering. While small σ values favour the dynamic informations, this can lead to scattered clusters neglecting the spatial component. Large σ values will lead to more regular and convex clusterings by enforcing a stronger spatial coupling between nearby fixations.

4.4 Results

The following pictures were computed from about 2000 fixations. $\Delta\tau$ was chosen small enough for most fixations to be counted ($\sim$95%), then sigma was raised from low values to reach the desired regularity.

Figure 4.1: Raphael - Sistine Madonna

Figure 4.2: Rembrandt - Landschaft mit den drei Bäumen

Figure 4.3: Hokusai - The Fuji from the mountains of Totomi

5 Discussion

Given a good choice of parameters the algorithm showed to be able to cluster the points to the subjectively identified objects in the picture, backing up the hypothesis that human sight exhibits the metastable behaviour on recognized objects.

This conclusion already implies the current problem of the approach, the parameters. The time step parameter $\Delta\tau$ could be completely eliminated by using a time-continuous

markov chain as underlying transition model, leading to a transition rate matrix. Unfortunately the ad-hoc approach using

$$W_{ij} = \left(\frac{\sum_{a \to b} M_{ai} M_{bj} \tau_{a \to b}}{\sum_{a \to b} M_{ai}} \right)^{-1},$$

denoting by $\sum_{a \to b}$ the sum over all fixation transitions, from a to b, and $\tau_{a \to b}$ the corresponding transition time, lead to worse results. It is not yet clear to the author how to construct the most likelihood estimator for the corresponding process.

The last fixation poses another problem leading to a ..

The results could probably be further improved by enhancing the method by which the fuzzy clustering is turned to a discrete one in 4.1. One possibility here might be weighting the clusters with their size, thus emphasizing their relative shape, or just discarding points with no clear assignment to an extra cluster.

References

[1] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled markov chains. *Linear Algebra and its Applications*, 315:39–59, 2000.

[2] P. Deuflhard and M. Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005.

[3] Wilhelm Huisinga. *Metastability of Markovian systems: A transfer operator based approach in application to molecular dynamics*. PhD thesis, Free University Berlin, 2001.

[4] Marco Sarich Bettina Keller Martin Senne Martin Held John D. Chodea Christof Schütte Frank Noé Jan-Hendrik Prinz, Hao Wu. Markov models of molecular kinetics: Generation and validation. *Journal of Chemical Physics*, 134, 2011.

[5] Daniel Kressner. Block algorithms for reordering standard and generalized schur forms. *ACM Transactions on Mathematical Software*, 2006.

[6] S. Kube and M. Weber. A coarse graining method for the identifiation of transition rates between molecular conformations. *Journal of Chemical Physics*, 126(2), 2007.

[7] K. Fackeldey M. Weber. G-pcca: Spectral clustering for non-reversible markov chains. 2015.

[8] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[9] S. Röblitz and M. Weber. Fuzzy spectral clustering by pcca+: application to markov state models and data classification. *Advances in Data Analysis and Classification*, 7:147–179, 2013.

[10] M. Weber. *Meshless methods in Conformation Dynamics*. PhD thesis, Free University Berlin, 2006.

[11] M. Weber and T. Galliat. Characterization of transition states in conformational dynamics using fuzzy sets. *ZIB-Report*, 02-12, 2002.