ARGUS

Activity Recognition and Object Tracking Based on Multiple Models

PTDC/EEA-CRO/098550/2008

Progress Report

December 23, 2010

Contents

Chapter 1

Theory

1.1 Notation

Symbol	Description
$\mathbf{v}$	Boldface lowercase: Vectors in matrix notation (line or column).
$\mathbf{A}$	Boldface uppercase: Matrices.
$\mathcal{X}$	Caligraphic: Sets, manifolds.
$\mathbb{F}$	Groups, Fields. ($\mathbb{R}$ for reals, $\mathbb{S}^K$ for stochastic matrices $K \times K$).
$\lvert\mathbf{A}\rvert$	Determinant of a matrix.
$\operatorname{Tr}\mathbf{A}$	Trace of a matrix.
$\lVert\mathbf{z}\rVert_{\mathbf{Q}}$	Vector norm $\sqrt{\mathbf{z}^T\mathbf{Q}\mathbf{z}}$.

TO DO: v^i for column vectors in component form and v_i for rows.
A^i_j for matrices in component form.
TO DO: T_k is a vector, should be lowercase.
TO DO: t_k is also used as time index. replace by v to denote velocity vector.
TO DO: write matrix version of (1.12)?

Symbol		Description
t	$\in \mathbb{N}$	Time index used to define a trajectory as an ordered set of points.
x	$\in \mathbb{R}^D$	Position. $D = 2$ for images.
x_t	$\mathbb{N} \to \mathbb{R}^D$	Position at time t.
$\overline{x}$	$:= (x_1, \ldots, x_L)$	Trajectory containing L points.
$\mathcal{X}$	$:= \{\overline{x}^1, \ldots, \overline{x}^S\}$	Set of trajectories.
K	$\in \mathbb{N}$	Number of trajectory models.
k	$\in \{1, \ldots, K\}$	A particular trajectory model.
k_t	$\mathbb{N} \to \{1, \ldots, K\}$	Trajectory model active at time t.
$\overline{k}$	$:= (k_1, \ldots, k_L)$	Sequence followed by the switching variable k_t.
$\mathcal{K}$	$:= \{\overline{k}^1, \ldots, \overline{k}^S\}$	Set of sequences followed by the switching variable.
$T_k(x)$	$\mathbb{N} \times \mathbb{R}^D \to \mathbb{R}^D$	Velocity vector at point x for trajectory model k.
T_k		Vector field of trajectory model k. There are K vector fields.
$\mathbf{t}^n_k$		Velocity vector at node n of trajectory model k.
$\mathcal{T}$		Set of vectors for all nodes and models k.
$b_{ij}(x)$	$\mathbb{N} \times \mathbb{N} \times \mathbb{R}^D \to \mathbb{R}$	Switching probabilities $\Pr\big\{k_t = j \vert k_{t-1} = i, x\big\}$.
$B(x)$	$\mathbb{R}^D \to \mathbb{S}^K$	Stochastic matrix corresponding to $b_{ij}(x)$.
b^n_{ij}	$\mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{R}$	Switching probabilities at the node n.
B^n	$\mathbb{N} \to \mathbb{S}^K$	Stochastic matrix corresponding to b^n_{ij}.
$\mathcal{B}$		
$\phi_n(x)$	$\mathbb{N} \times \mathbb{R}^D \to \mathbb{R}$	Basis function centered at node n (used for interpolation).

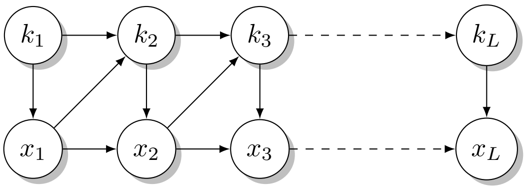

Figure 1.1: Markov diagram showing the state variable x_t and active model k_t updates.

Some variables use both lower and upper indices (*e.g.* b_{ij}^k). The index position is somewhat arbitrary and does not denote the covariant/contravariant tensor component notation. Einstein summation convention is not used. All summations are explicitly indicated.

1.2 Problem statement

The problem under consideration deals with the identification of a set vector fields $T_k(x)$, $k \in \{1, \ldots, K\}$ and associated transition probabilities $b_{ij}(x)$ that best describe a multiple model switched nonlinear system, where the switching mechanism is governed by a state dependent hidden Markov model.

A collection $\mathcal{X}$ of trajectories are initially known. Each trajectory $x^s \in \mathcal{X}$, $s \in \{1, \ldots, S\}$, is an ordered set points $x_t^s \in \mathbb{R}^D$, $t = 1, \ldots, L_s$. Trajectories generally have different lengths L_s.

The space $\mathbb{R}^D$ discretized into a grid composed of N nodes. These nodes can be irregularly spaced. The current work considers nodes fixed on previously defined locations.

The aim is to find a set of K basic trajectories which best approximate the S observed trajectories. Trajectories are allowed to switch arbitrarily according to some transition probabilities. Transition probabilities change in space.

1.3 Trajectory model

Trajectories are ordered sets of points in $\mathbb{R}^D$ indexed by t. The "time" index t is not an universal time but rather an index position. There is no information whether one recorded trajectory occurred before of after others.

Recorded trajectories are assumed to be generated by K different velocity fields switched according to a switching variable $k_t \in \{1, \ldots, K\}$. The switching variable itself is described by a position dependent Markov model.

The vector field and transition probabilities field are used to define a discrete time dynamical system with hybrid state (k_t, x_t), where k_t is a discrete variable denoting the active trajectory at time t, and $x_t \in \mathbb{R}^D$ is a continuous variable indicating the position at time t.

The hybrid dynamic system is updated by

$$\Pr\big\{k_t = j | k_{t-1} = i, x_{t-1}\big\} = b_{ij}(x_{t-1}) \tag{1.1}$$

$$x_t = x_{t-1} + T_{k_t}(x_{t-1}) + w_t \tag{1.2}$$

where $w_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{k_t})$ is a zero-mean multivariable Gaussian with constant (in space) covariance matrix Σ_{k_t} for each vector field, and $b_{ij}(x)$ is the transition probability from vector field T_i to T_j at position x. Equation (1.1) computes the active vector field T_{k_t} used in equation (1.2) to find the updated position x_t. Figure 1.1 shows the information flow of these two equations along time.

1.4 Trajectory distribution

This section shows the computation of the probability of observing a single trajectory $x_{1:L}$ of length L and corresponding switching sequence $k_{1:L}$. For this purpose, the trajectory model developed in section 1.3 is used.

Consider one trajectory $x_{1:L} \triangleq (x_1, \ldots, x_L)$ and the corresponding switching variables $k_{1:L} \triangleq (k_1, \ldots, k_L)$. Their joint probability mass function is

$$\begin{aligned} p(x_{1:L}, k_{1:L}) &= p(x_1, k_1) \prod_{t=2}^{L} p(x_t, k_t | x_{t-1}, k_{t-1}) \\ &= p(x_1, k_1) \prod_{t=2}^{L} p(x_t | k_t, x_{t-1}, k_{t-1}) p(k_t | x_{t-1}, k_{t-1}). \end{aligned} \tag{1.3}$$

For compactness, lets assume $p(x_1, k_1) = p(x_1, k_1 | x_0, k_0)$ for some dummy variables (x_0, k_0). Since x_t is conditionally independent of k_{t-1} given k_t, the following holds:

$$p(x_{1:L}, k_{1:L}) = \prod_{t=1}^{L} p(x_t | k_t, x_{t-1}) p(k_t | x_{t-1}, k_{t-1}). \tag{1.4}$$

Conditional probabilities in (1.4) are given by equations (1.1) and (1.2), therefore

$$p(x_t | k_t, x_{t-1}) = \mathcal{N}\big(x_t \big| \; x_{t-1} + T_{k_t}(x_{t-1}), \Sigma_{k_t}\big) \tag{1.5}$$

$$p(k_t | x_{t-1}, k_{t-1}) = b_{k_{t-1}, k_t}(x_{t-1}). \tag{1.6}$$

The multivariable Gaussian distribution (1.5) is

$$\mathcal{N}\big(x_t \big| \; x_{t-1} + T_{k_t}(x_{t-1}), \Sigma_{k_t}\big) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_{k_t}|}} e^{-\frac{1}{2} \left\| x_t - x_{t-1} - T_{k_t}(x_{t-1}) \right\|^2_{\Sigma^{-1}_{k_t}}}. \tag{1.7}$$

Taking the logarithm of equation (1.4) yields

$$\begin{aligned} \log p(x_{1:L}, k_{1:L}) &= \sum_{t=1}^{L} \log p(x_t | k_t, x_{t-1}) + \sum_{t=1}^{L} \log p(k_t | x_{t-1}, k_{t-1}) \\ &= -\frac{1}{2} \sum_{t=1}^{L} \log (2\pi)^D |\Sigma_{k_t}| - \frac{1}{2} \sum_{t=1}^{L} \left\| x_t - x_{t-1} - T_{k_t}(x_{t-1}) \right\|^2_{\Sigma^{-1}_{k_t}} \\ &\quad + \sum_{t=1}^{L} \log b_{k_{t-1}, k_t}(x_{t-1}). \end{aligned} \tag{1.8}$$

This expressions will be useful latter on.

1.5 Space discretization

Vector fields $T_{k_t}(x_{t-1})$ and transition probabilities $b_{ij}(x_{t-1})$ are defined for any point in $\mathbb{R}^D$. Since we can not describe them in full generality on a finite memory setting, a discretization is going to be performed so that only a finite number of parameters is used to define these fields.

The main idea is to use a predefined grid having nodes located at some coordinates $g_n \in \mathbb{R}^D$, a vector and a transition matrix being associated to each node. The vector and transition matrix fields are then obtained by interpolation.

The grid $g_n \in \mathbb{R}^D$ consists of N fixed nodes. Each node n has K velocity vectors $\mathbf{t}_k^n \in \mathbb{R}^D$, $k \in \{1, \dots, K\}$. The interpolation is performed by N basis functions $\phi_n(x)$ centered at nodes $n \in \{1, \dots, N\}$, such that

$$T_k(x) = \sum_{n=1}^{N} \mathbf{t}_k^n \phi_n(x). \tag{1.9}$$

The previous equation can be written in matrix form as

$$T_k(x) = \mathbf{T}_k \Phi(x), \tag{1.10}$$

where the matrices $\mathbf{T}_k \in \mathbb{R}^{D \times N}$ and $\Phi(x) \in \mathbb{R}^{N \times 1}$ are defined by

$$\mathbf{T}_k \triangleq \begin{bmatrix} \mathbf{t}_k^1 & \cdots & \mathbf{t}_k^N \end{bmatrix}, \qquad \Phi(x) \triangleq \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_N(x) \end{bmatrix}. \tag{1.11}$$

Similarly, switching probabilities $b_{ij}(x_{t-1})$ are interpolated from N matrices b_{ij}^n, defined at the nodes $n \in \{1, \dots, N\}$,

$$b_{ij}(x_{t-1}) = \sum_{n=1}^{N} b_{ij}^n \phi_n(x_{t-1}). \tag{1.12}$$

Rewriting the switching probabilities b_{ij} as a stochastic matrix $\mathbf{B} \triangleq \begin{bmatrix} b_{ij} \end{bmatrix}$ yields

$$\mathbf{B}(x_{t-1}) = \sum_{n=1}^{N} \mathbf{B}^n \phi_n(x_{t-1}). \tag{1.13}$$

The interpolating function $\phi_n(x)$ has to satisfy the constraints

$$0 \leq \phi_n(x) \leq 1 \tag{1.14}$$

$$\sum_{n=1}^{N} \phi_n(x) = 1 \tag{1.15}$$

for all $x \in \mathbb{R}^D$. This constraint ensures that the combination (1.13) does still define a stochastic matrix $\mathbf{B}(x_{t-1})$. Smoothness of $\phi(x)$ is not required.

It may be desirable to enforce additionally the constraint $\phi_n(g_n) = 1$ so that the stochastic matrix $\mathbf{B}^n$ reflects exactly the matrix $\mathbf{B}(g_n)$. If this condition is not satisfied, then it may not be possible to achieve all stochastic matrices at the nodes since $\mathbf{B}^n$ is bounded and the other nodes will be pulling this matrix in other directions. It is not yet clear if this is a real drawback or not.

1.6 Parameter estimation

Parameter estimation aims to find the vector fields and switching probabilities at each node of the grid from a set of sampled trajectories in $\mathbb{R}^D$. The covariance matrix Σ_k for each trajectory model is also estimated FIXME: not yet done.

The model depends on unknown parameters $\theta = (\mathcal{T}, \mathcal{B}, \mathcal{S})$, which include the set of vectors $\mathcal{T} = \{\mathbf{t}_k^n\}$ for all nodes and trajectories, the set transition matrices $\mathcal{B} = \{\mathbf{B}^1, \dots, \mathbf{B}^N\}$ for the nodes, and the set of covariance matrices $\mathcal{S} = \{\Sigma_1, \dots, \Sigma_K\}$ associated with each model k.

1.7 Prior selection

Assuming independence between the vector fields, transition matrices and the noise covariances, the prior $p(\theta)$ becomes

$$p(\theta) = p(\mathcal{T})p(\mathcal{B})p(\mathcal{S}). \tag{1.16}$$

Under this assumption, priors can be built separately. This is done in the following subsections.

1.7.1 Priors for the vector fields

$p(\mathcal{T})$ is built from independent distributions $p(\mathbf{t}_k^n)$ defined over the set of trajectory models k. For each individual model k, velocity vectors $\mathbf{t}_k^n$ are assumed to be dependent across neighbor nodes according to a multivariable Gaussian distribution with covariance matrix $\mathbf{\Lambda}$. Thus,

$$p(\mathcal{T}) = \prod_{k=1}^{K} p(\mathcal{T}_k) = \prod_{k=1}^{K} \mathcal{N}(\mathcal{T}_k|\mathbf{0}, \mathbf{\Lambda}), \tag{1.17}$$

where $\mathcal{T}_k \triangleq \{\mathbf{t}_k^1, \ldots, \mathbf{t}_k^N\}$ denotes the set of vectors from model k, and $\mathbf{\Lambda}$ is the covariance matrix describing the dependencies of vectors between neighbor nodes.

Let $\mathcal{I}$ denote the set of pairs of indices (i, j) containing neighbor nodes. *i.e.*,

$$\mathcal{I} = \{(i, j) \mid i \text{ and } j \text{ are neighbors, and } i \neq j\}. \tag{1.18}$$

Then $p(\mathcal{T}_k)$ is defined as a multivariable Gaussian where vectors, in neighbor nodes, having similar directions and lengths take larger probabilities:

$$p(\mathcal{T}_k) \propto e^{-\frac{1}{2\alpha}\sum_{(i,j)\in\mathcal{I}} \|\mathbf{t}_k^i - \mathbf{t}_k^j\|^2}. \tag{1.19}$$

This approach has a regularization effect, controlled by α, so that nodes where no data is available gather information from their neighbors, thereby introducing smoothness into the vector field.

The covariance matrix $\mathbf{\Lambda}$ is built as follows. Define a matrix $\mathbf{\Delta}$ which operates the vectors $\mathbf{t}_k^i$, $i = 1, \ldots, n$, to obtain the differences between neighbors:

$$\mathbf{T}_k\mathbf{\Delta} = \begin{bmatrix} \mathbf{t}_k^1 & \cdots & \mathbf{t}_k^N \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 1 \\ -1 & 1 & & 0 \\ 0 & -1 & & \vdots \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & -1 \end{bmatrix}}_{\#\mathcal{I}}. \tag{1.20}$$

Then

$$\begin{aligned} \sum_{(i,j)\in\mathcal{I}} \|\mathbf{t}_k^i - \mathbf{t}_k^j\|^2 &= \sum_{(i,j)\in\mathcal{I}} (\mathbf{t}_k^i - \mathbf{t}_k^j)^T(\mathbf{t}_k^i - \mathbf{t}_k^j) \\ &= \operatorname{Tr}\left((\mathbf{T}_k\mathbf{\Delta})^T(\mathbf{T}_k\mathbf{\Delta})\right) \\ &= \operatorname{Tr}\left(\mathbf{T}_k\mathbf{\Delta}\mathbf{\Delta}^T\mathbf{T}_k^T\right). \end{aligned} \tag{1.21}$$

Defining $\mathbf{\Lambda}^{-1} \triangleq \mathbf{\Delta}\mathbf{\Delta}^T$, then (1.19) becomes

$$p(\mathcal{T}_k) \propto e^{-\frac{1}{2\alpha}\operatorname{Tr}(\mathbf{T}_k\mathbf{\Lambda}^{-1}\mathbf{T}_k^T)}. \tag{1.22}$$

The matrix $\mathbf{\Lambda}^{-1}$ as defined above may fail to be positive definite for some neighbor combinations in $\mathcal{I}$. On such occurrence the prior would become improper. To overcome this difficulty, it can be redefined as

$$\mathbf{\Lambda}^{-1} \triangleq \epsilon\mathbf{I} + \mathbf{\Delta}\mathbf{\Delta}^T \tag{1.23}$$

to include a term ensuring positive definiteness for some small value ϵ.

Alternative: In the previous formulation, the neighbors are either neighbors or not neighbors of a given node. A more smooth definition can be made, where neighborhood is weighted by some matrix W. We shall consider the following formulation. A node n has a set of neighbors $\mathcal{V}_n$. These neighbors include a special member $\mathbf{t}_k^0$ which works as a base pulling the estimated vector towards it. The multivariable Gaussian can be written as

$$p(\mathcal{T}_k) \propto e^{-\frac{1}{2\alpha}\sum_{n=1}^{N}\sum_{i\in\mathcal{V}_n} w_n^i \|\mathbf{t}_k^n - \mathbf{t}_k^i\|^2} \tag{1.24}$$

Discussion: The neighbors used to build the Gaussian can be selected by an indicator function I_{ij} which equals one when i and j are neighbors and zero otherwise. Then we can write

$$\sum_{i,j} I_{ij}(\mathbf{t}_k^i - \mathbf{t}_k^j)^T(\mathbf{t}_k^i - \mathbf{t}_k^j). \tag{1.25}$$

The indicator function could be replaced by a smooth function related to the distance between the nodes i and j, so that closer nodes tend to impose more regularization than distant ones. Example: $I_{ij} = e^{-d_{ij}}$, where d_{ij} is the node distance in $\mathbb{R}^D$.

1.7.2 Priors for stochastic matrices

The simplest approach is to build $p(\mathcal{B})$ as a constant density over the set of stochastic matrices $\mathbb{S}^K$:

$$p(b_{ij}^n) \propto 1, \tag{1.26}$$

i.e. an uniform probability density function $b_{ij}^n \sim \text{unif}(\mathbb{S}^K)$. Components b_{ij}^n of a stochastic matrix are just one possible coordinate system. Other parameterizations are possible, for example exponential coordinates.

Equation (1.26) specifies an uniform density on a specific parameterization. Unfortunately, this distribution is not invariant under coordinate transformations, meaning that a uniform density on different coordinates yields a different prior.

The Jeffreys prior defines a distribution which is invariant under coordinate transformations. This prior is built from the Fisher information matrix $\mathbf{G}$ as follows (see appendix A.1):

$$p(b_{ij}^n) \propto \sqrt{\det \mathbf{G}_i^n} = \frac{1}{\prod_{k=1}^{K}\sqrt{b_{ik}^n}}. \tag{1.27}$$

This distribution has higher density for probabilities near zero.

1.7.3 Priors for noise covariance matrices

$p(\mathcal{S})$ is built from independent distributions $p(\mathbf{\Sigma}_k)$ over the set of trajectory models k. Since covariances are constant for all the nodes of the same trajectory model k, then

$$p(\mathcal{S}) = \prod_{k=1}^{K} p(\mathbf{\Sigma}_k). \tag{1.28}$$

The definition of a prior to the covariance matrix is analogous to the prior for stochastic matrices. The covariance matrix is a set of parameters used to define a Gaussian distribution. The problem is to assign a probability density function to these parameters.

First a parameterization has to be selected from many possibilities like, *e.g.*, Cholesky factorization by lower triangular matrices with positive diagonal components

$$\mathbf{\Sigma} = \mathbf{L}^T\mathbf{L}. \tag{1.29}$$

Then, a probability density function has to be defined over the triangular matrix $\mathbf{L}$. This leads to the same kind of problems as the stochastic matrix, namely of invariance under coordinate transformations, *i.e.*, under different parameterizations of the Gaussian density function.

The following possibilities should be analised:

1. Assume a Dirac delta centered at some $\mathbf{\Sigma}$ meaning that no estimation is to be performed.

2. Do not specify the prior $p(\mathbf{\Sigma}_k)$ at all, and then use a Maximum Likelihood criteria for this particular parameter.

3. Find an invariant prior distribution for $\mathbf{\Sigma}$. FIXME: Jeffrey?

1.8 Expectation-Maximization algorithm

The model is estimated from a set of S trajectories $\mathcal{X} = \{x^1, \ldots, x^S\}$, where trajectories $x^s = (x_1^s, \ldots, x_{L_s}^s)$ can have different lengths L_s. The corresponding switching trajectories $\mathcal{K} = \{k^1, \ldots, k^S\}$ are made from unobserved latent variables $k^s = \{k_1^s, \ldots, k_{L_s}^s\}$ with the same length.

The MAP estimate $\hat{\theta}$ is defined as

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} p(\theta|\mathcal{X}) \\ &= \arg\max_{\theta} p(\mathcal{X}, \theta) \\ &= \arg\max_{\theta} p(\mathcal{X}|\theta)p(\theta) \\ &= \arg\max_{\theta} p(\theta) \sum_{\mathcal{K}} p(\mathcal{X}, \mathcal{K}|\theta)\end{aligned} \tag{1.30}$$

The marginalization over $\mathcal{K}$ present in the last equality requires a summation of $O(K^{LS})$ terms, which is computationally unfeasible. We will therefore maximize the one before the last using the Expectation-Maximization (EM) method. For this purpose, the complete joint distribution $p(\mathcal{X}, \mathcal{K}, \theta)$ is used.

Taking the logarithm of the complete joint distribution $p(\mathcal{X}, \mathcal{K}, \theta)$, we get

$$\begin{aligned}\log p(\mathcal{X}, \mathcal{K}, \theta) &= \log p(\mathcal{X}, \mathcal{K}|\theta) + \log p(\theta) \\ &= \log p(\mathcal{K}|\mathcal{X}, \theta) + \log p(\mathcal{X}|\theta) + \log p(\theta).\end{aligned} \tag{1.31}$$

Now suppose an initial guess $\hat{\theta}$ is available. The expected value $E[\,\cdot\,|\mathcal{X}, \hat{\theta}]$ with respect to $p(\mathcal{K}|\mathcal{X}, \hat{\theta})$ is

$$\begin{aligned}&\overbrace{E\big[\log p(\mathcal{X}, \mathcal{K}, \theta)|\mathcal{X}, \hat{\theta}\big]}^{U(\theta,\hat{\theta})} = \\ &\qquad = \underbrace{E\big[\log p(\mathcal{K}|\mathcal{X}, \theta)|\mathcal{X}, \hat{\theta}\big]}_{V(\theta,\hat{\theta})} + \underbrace{E\big[\log p(\mathcal{X}|\theta)|\mathcal{X}, \hat{\theta}\big]}_{\log p(\mathcal{X}|\theta)} + \underbrace{E\big[\log p(\theta)|\mathcal{X}, \hat{\theta}\big]}_{\log p(\theta)}.\end{aligned} \tag{1.32}$$

Then,

$$\begin{aligned}\arg\max_{\theta} p(\mathcal{X}|\theta)p(\theta) &= \arg\max_{\theta} \big(\log p(\mathcal{X}|\theta) + \log p(\theta)\big) \\ &= \arg\max_{\theta} \big(U(\theta, \hat{\theta}) - V(\theta, \hat{\theta})\big).\end{aligned} \tag{1.33}$$

Since $V(\theta, \hat{\theta}) \leq V(\hat{\theta}, \hat{\theta})$ for any choice of θ, then increasing the value of $U(\theta, \hat{\theta})$ will also increase the difference $U(\theta, \hat{\theta}) - V(\theta, \hat{\theta})$. The EM algorithm takes advantage of this property. It basically consists in the computation of $U(\theta, \hat{\theta})$, named the E-step, and then its maximization in the M-step. The alternation of these two steps will eventually converge to a local maximum.

1.8.1 The E-step

The E-step of the EM algorithm aims to find

$$\begin{aligned} U(\theta,\hat{\theta}) &= E\big[\log p(\mathcal{X},\mathcal{K},\theta)|\mathcal{X},\hat{\theta}\big] \\ &= E\big[\log p(\mathcal{X},\mathcal{K}|\theta)|\mathcal{X},\hat{\theta}\big] + \log p(\theta). \end{aligned} \tag{1.34}$$

The term $\log p(\theta)$ at the right contains the prior information. It is given by

$$\begin{aligned} \log p(\theta) &= \log p(\mathcal{T}) + \log p(\mathcal{B}) + \log p(\mathcal{S}) \\ &= \text{const} - \frac{1}{2}\sum_{k=1}^{K} \operatorname{Tr}(\mathbf{T}_k \mathbf{\Lambda}^{-1} \mathbf{T}_k^T) \\ &\quad + \sum_{n=1}^{N} \log p(\mathbf{B}^n) \\ &\quad + \sum_{k=1}^{K} \log p(\mathbf{\Sigma}_k). \end{aligned} \tag{1.35}$$

To compute the expectation in (1.34), the expression $\log p(\mathcal{X},\mathcal{K}|\theta)$ is evaluated first. Equation (1.8) is applied to each individual trajectory for this porpuse:

$$\begin{aligned} \log p(\mathcal{X},\mathcal{K}|\theta) &= \sum_{s=1}^{S} \log p(x^s, k^s|\theta) \\ &= -\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{L_s} \log\left((2\pi)^D |\mathbf{\Sigma}_{k_t^s}|\right) \\ &\quad - \frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{L_s} \left\| x_t^s - x_{t-1}^s - T_{k_t^s}(x_{t-1}^s)\right\|^2_{\mathbf{\Sigma}^{-1}_{k_t^s}} \\ &\quad + \sum_{s=1}^{S}\sum_{t=1}^{L_s} \log b_{k_{t-1}^s, k_t^s}(x_{t-1}^s). \end{aligned} \tag{1.36}$$

Then, the expectation $E\big[\log p(\mathcal{X},\mathcal{K}|\theta)\big|\mathcal{X},\hat{\theta}\big]$ can be performed for each of the three parcels individually.

- For the first parcel we have:

  FIXME: compute the expected value. this is required for optimization of Σ.

- For the second parcel we have:

$$\begin{aligned} &E\Big[-\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{L_s}\left\| x_t^s - x_{t-1}^s - T_{k_t^s}(x_{t-1}^s)\right\|^2_{\mathbf{\Sigma}^{-1}_{k_t^s}}\Big|\mathcal{X},\hat{\theta}\Big] = \\ &\qquad = -\frac{1}{2}\sum_{\mathcal{K}} p(\mathcal{K}|\mathcal{X},\hat{\theta}) \sum_{s=1}^{S}\sum_{t=1}^{L_s} \left\| x_t^s - x_{t-1}^s - T_{k_t^s}(x_{t-1}^s)\right\|^2_{\mathbf{\Sigma}^{-1}_{k_t^s}} \\ &\qquad = -\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{k_t^s=1}^{K} p(k_t^s|\mathcal{X},\hat{\theta}) \left\| x_t^s - x_{t-1}^s - T_{k_t^s}(x_{t-1}^s)\right\|^2_{\mathbf{\Sigma}^{-1}_{k_t^s}} \\ &\qquad = -\frac{1}{2}\sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{k=1}^{K} w_k^s(t) \left\| x_t^s - x_{t-1}^s - \sum_{n=1}^{N} \phi_n(x_{t-1}^s) T_k^n \right\|^2_{\mathbf{\Sigma}_k^{-1}}, \end{aligned} \tag{1.37}$$

  where the probabilities $w_j^s(t) \triangleq \Pr\big\{k_t^s = j|\mathcal{X},\hat{\theta}\big\}$ have to be computed separately.

- For the third parcel we have:

$$
\begin{aligned}
E\Big[\sum_{s=1}^{S}\sum_{t=1}^{L_s}\log b_{k^s_{t-1},k^s_t}(x^s_{t-1})\Big|\mathcal{X},\hat{\theta}\Big] = \\
&= \sum_{\mathcal{K}} p(\mathcal{K}|\mathcal{X},\hat{\theta})\sum_{s=1}^{S}\sum_{t=1}^{L_s}\log b_{k^s_{t-1},k^s_t}(x^s_{t-1}) \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{\mathcal{K}} p(\mathcal{K}|\mathcal{X},\hat{\theta})\log b_{k^s_{t-1},k^s_t}(x^s_{t-1}) \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{\mathcal{K}} p(k^s_{t-1},k^s_t|\mathcal{X},\hat{\theta})p\big(\mathcal{K}-\{k^s_{t-1},k^s_t\}\mid k^s_{t-1},k^s_t,\mathcal{X},\hat{\theta}\big)\cdot \\
&\quad \cdot\log b_{k^s_{t-1},k^s_t}(x^s_{t-1}) \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{k^s_{t-1},k^s_t=1}^{K} p(k^s_{t-1},k^s_t|x^s,\hat{\theta})\log b_{k^s_{t-1},k^s_t}(x^s_{t-1}) \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{i,j=1}^{K} w^s_{ij}(t)\log b_{i,j}(x^s_{t-1}) \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{i,j=1}^{K} w^s_{ij}(t)\log\Big(\sum_{n=1}^{N} b^n_{ij}\phi_n(x^s_{t-1})\Big),
\end{aligned}
\tag{1.38}
$$

where

$$w^s_{ij}(t)\triangleq\Pr\big\{k^s_{t-1}=i,k^s_t=j|x^s,\hat{\theta}\big\}. \tag{1.39}$$

The computation of the components $w^s_j(t)$ and $w^s_{ij}(t)$ is done using the forward-backward algorithm described in the following section.

1.8.2 Forward-backward algorithm

The forward-backward algorithm allows the computation of $w^s_{ij}(t)$ and $w^s_i(t)$.

The forward part works as follows: Consider the forward variable

$$\alpha^s_t(i)\triangleq\Pr\big\{x^s_{1:t},k^s_t=i|\hat{\theta}\big\}. \tag{1.40}$$

For any time $t\in\{1,\ldots,L_s\}$, the computation of $\alpha^s_t(i)$ can be done with the following algorithm:

1. Initialization:

$$
\begin{aligned}
\alpha^s_1(i) &= \Pr\big\{x^s_1,k^s_1=i|\hat{\theta}\big\} \\
&= \Pr\big\{x^s_1|k^s_1=i,\hat{\theta}\big\}\Pr\big\{k^s_1=i|\hat{\theta}\big\}
\end{aligned}
\tag{1.41}
$$

Since we do not have information from times $t<1$, which would allow us to compute the probability of observing the first state, we set the first probability to one. Then the initialization depends only on the probability of the active model at $t=1$:

$$\alpha^s_1(i)=\Pr\big\{k^s_1=i|\hat{\theta}\big\}. \tag{1.42}$$

2. Induction:

$$\begin{aligned}\alpha_t^s(i) &= \Pr\big\{x_t^s|x_{t-1}^s, k_t^s = i, \hat{\theta}\big\}\Pr\big\{x_{1:t-1}^s, k_t^s = i|\hat{\theta}\big\} \\ &= \Pr\big\{x_t^s|x_{t-1}^s, k_t^s = i, \hat{\theta}\big\}\sum_{j=1}^{K}\Pr\big\{k_t^s = i|k_{t-1}^s = j, x_{t-1}^s, \hat{\theta}\big\}\Pr\big\{x_{1:t-1}^s, k_{t-1}^s = j|\hat{\theta}\big\} \\ &= \mathcal{N}\big(x_t^s\big|x_{t-1}^s + T_i(x_{t-1}^s), \mathbf{\Sigma}_{k_t^s}\big)\sum_{j=1}^{K} b_{ij}(x_{t-1}^s)\alpha_{t-1}^s(j)\end{aligned} \tag{1.43}$$

The backward part of the algorithm works similarly, but from the other end backwards in time. It is as follows: Consider the backward variable

$$\beta_t^s(i) \triangleq \Pr\big\{x_{t+1:L_s}^s|k_t^s = i, x_t, \hat{\theta}\big\}. \tag{1.44}$$

It can be computed using the following algorithm:

1. Initialization:

$$\beta_{L_s}^s(i) = 1 \tag{1.45}$$

2. Induction: Since x_{t+1} is conditionally independent of the remaining trajectory $x_{t+2:L_s}$ given x_t and k_{t+1}, then $\beta_t^s(i)$ is given by

$$\begin{aligned}\beta_t^s(i) &= \sum_{j=1}^{K}\Pr\big\{x_{t+1:L_s}^s|k_{t+1}^s = j, x_t^s\big\}\ \Pr\big\{k_{t+1}^s = j|k_t^s = i, x_t\big\} \\ &= \sum_{j=1}^{K}\Pr\big\{x_{t+2:L_s}^s|k_{t+1}^s = j, x_{t+1}^s\big\}\ \Pr\big\{x_{t+1}^s|k_{t+1}^s = j, x_t\big\}\ \Pr\big\{k_{t+1}^s = j|k_t^s = i, x_t\big\} \\ &= \sum_{j=1}^{K}\beta_{t+1}^s(j)\ \mathcal{N}\big(x_{t+1}^s\big|x_t^s + T_j(x_t), \mathbf{\Sigma}_j\big)\ b_{ij}(x_t).\end{aligned} \tag{1.46}$$

The variables $\alpha_t^s(i)$ and $\beta_t^s(i)$ can be combined together to produce the weights $w_i^s(t)$ and $w_{ij}^s(t)$ as shown in the following equations:

$$\begin{aligned}w_i^s(t) &= \Pr\big\{k_t^s = i|\bar{x}^s, \hat{\theta}\big\} \\ &= \frac{\Pr\big\{x^s, k_t^s = i|\hat{\theta}\big\}}{\Pr\big\{x^s|\hat{\theta}\big\}} \\ &= \frac{\Pr\big\{x_{1:t}^s, k_t^s = i|\hat{\theta}\big\}\ \Pr\big\{x_{t+1:L_s}^s|k_t^s = i, x_t, \hat{\theta}\big\}}{\sum_{j=1}^{K}\Pr\big\{x^s, k_t^s = j|\hat{\theta}\big\}} \\ &= \frac{\alpha_t^s(i)\beta_t^s(i)}{\sum_{j=1}^{K}\alpha_t^s(j)\beta_t^s(j)}.\end{aligned} \tag{1.47}$$

$$\begin{aligned}w_{ij}^s(t) &= \Pr\big\{k_{t-1}^s = i, k_t^s = j|x^s, \hat{\theta}\big\} \\ &= \Pr\big\{k_{t-1}^s = i, k_t^s = j, x^s|\hat{\theta}\big\}\ /\ \Pr\big\{x^s|\hat{\theta}\big\} \\ &= \Pr\big\{x_{1:t-1}^s, k_{t-1}^s = i\big|\hat{\theta}\big\}\ \Pr\big\{k_t^s = j\big|k_{t-1}^s = i, x_{t-1}^s, \hat{\theta}\big\}\ \Pr\big\{x_t^s\big|x_{t-1}^s, k_t^s = j, \hat{\theta}\big\} \\ &\quad \cdot \Pr\big\{x_{t+1:L_s}^s\big|k_t^s = j, x_t^s, \hat{\theta}\big\}\ /\ \Pr\big\{\bar{x}^s\big|\hat{\theta}\big\} \\ &= \frac{\alpha_{t-1}^s(i)\cdot b_{ij}(x_{t-1}^s)\cdot\mathcal{N}\big(x_t^s\big|x_{t-1}^s + T_j(x_{t-1}^s), \mathbf{\Sigma}_j\big)\cdot\beta_t^s(j)}{\Pr\big\{\bar{x}^s\big|\hat{\theta}\big\}}\end{aligned} \tag{1.48}$$

where the normalization constant $\Pr\big\{x^s\big|\hat{\theta}\big\}$ is obtained by summing the numerator over all $1 \le i, j \le K$.

1.8.3 The M-step

The maximization step of the EM algorithm will attempt to maximize $U(\theta, \hat{\theta})$. Since the maximization is performed with respect to the vector field, transition probabilities and state disturbance covariance, we can compute partial derivatives of $U(\theta, \hat{\theta})$ individually.

Maximization with respect to the vector field

Differentiating $U(\theta, \hat{\theta})$ with respect to $\mathbf{T}_\alpha$ yields

$$\frac{\partial U}{\partial \mathbf{T}_\alpha} = \frac{\partial}{\partial \mathbf{T}_\alpha} E\big[\log p(\mathcal{X}, \mathcal{K}|\theta)\big|\mathcal{X}, \hat{\theta}\big] + \frac{\partial}{\partial \mathbf{T}_\alpha} \log p(\theta). \tag{1.49}$$

The derivative of the first term amounts to

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{T}_\alpha} E\big[\log p(\mathcal{X}, \mathcal{K}|\theta)\big|\mathcal{X}, \hat{\theta}\big] = \\
&= -\frac{1}{2}\frac{\partial}{\partial \mathbf{T}_\alpha} \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{k=1}^{K} w_k^s(t) \big\| x_t^s - x_{t-1}^s - \mathbf{T}_k \Phi(x_{t-1}^s)\big\|^2_{\mathbf{\Sigma}_k^{-1}} \\
&= \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_\alpha^s(t)\Phi(x_{t-1}^s)\Big[x_t^s - x_{t-1}^s - \mathbf{T}_\alpha \Phi(x_{t-1}^s)\Big]^T \mathbf{\Sigma}_\alpha^{-1} \\
&= -\mathsf{A}_\alpha\, \mathbf{T}_\alpha^T \mathbf{\Sigma}_\alpha^{-1} + \mathsf{B}_\alpha,
\end{aligned} \tag{1.50}$$

where $\mathsf{A}_\alpha \in \mathbb{R}^{N\times N}$ and $\mathsf{B}_\alpha \in \mathbb{R}^{N\times D}$ are given by

$$\mathsf{A}_\alpha = \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_\alpha^s(t)\Phi(x_{t-1}^s)\Phi(x_{t-1}^s)^T, \tag{1.51}$$

$$\mathsf{B}_\alpha = \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_\alpha^s(t)\Phi(x_{t-1}^s)(x_t^s - x_{t-1}^s)^T \mathbf{\Sigma}_\alpha^{-1}. \tag{1.52}$$

The second term is

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{T}_\alpha} \log p(\theta) &= \frac{\partial}{\partial \mathbf{T}_\alpha}\Big(-\frac{1}{2}\sum_{k=1}^{K} \operatorname{Tr}\left(\mathbf{T}_k \mathbf{\Lambda}^{-1}\mathbf{T}_k^T\right)\Big) \\
&= -\mathbf{\Lambda}^{-1}\mathbf{T}_\alpha^T.
\end{aligned} \tag{1.53}$$

The stationarity condition $\partial U/\partial \mathbf{T}_\alpha = 0$ is then

$$\mathbf{\Lambda}^{-1}\mathbf{T}_\alpha^T + \mathsf{A}_\alpha \mathbf{T}_\alpha^T \mathbf{\Sigma}_\alpha^{-1} = \mathsf{B}_\alpha. \tag{1.54}$$

A general solution of the equation above can be obtained from the following equation:

$$\operatorname{vec} \mathbf{T}_\alpha^T = \left(\mathbf{I}_{D\times D} \otimes \mathbf{\Lambda}^{-1} + \mathbf{\Sigma}_\alpha^{-1} \otimes \mathsf{A}_\alpha\right)^{-1} \operatorname{vec} \mathsf{B}_\alpha. \tag{1.55}$$

Here, the symbol $\otimes$ denotes the Kronecker product and the vec operator creates a column vector by stacking the columns of the original matrix. Unfortunately, this solution is computationally unfeasible as the dimension of linear equation is too high, requiring the construction of a $DN \times DN$ sized matrix. To circumvent this practical limitation two approaches can be followed:

1. Constraining the covariance matrix to a diagonal

$$\mathbf{\Sigma}_\alpha^{-1} = \frac{1}{\sigma_\alpha^2}\mathbf{I}_{D\times D}, \tag{1.56}$$

then equation (1.54) becomes a simpler linear equation

$$\Big(\mathbf{\Lambda}^{-1} + \frac{1}{\sigma_\alpha^2}\mathsf{A}_\alpha\Big)\mathbf{T}_\alpha^T = \mathsf{B}_\alpha, \tag{1.57}$$

whose solution is given by

$$\mathbf{T}_\alpha^T = \Big(\mathbf{\Lambda}^{-1} + \frac{1}{\sigma_\alpha^2}\mathsf{A}_\alpha\Big)^{-1}\mathsf{B}_\alpha. \tag{1.58}$$

2. If the regularization matrix $\mathbf{\Lambda}^{-1}$ is nonsingular, then premultiplying (1.54) by $\mathbf{\Lambda}$ turns it into a linear matrix equation of the Sylvester type

$$(-\mathbf{\Lambda}\mathsf{A}_\alpha)\mathbf{T}_\alpha^T(\mathbf{\Sigma}_\alpha^{-1}) - \mathbf{T}_\alpha^T + (\mathbf{\Lambda}\mathsf{B}_\alpha) = \mathbf{0}, \tag{1.59}$$

whose solution can be computed efficiently using *e.g.* the `dlyap()` Matlab function.

The second possibility is followed here.

Maximization with respect to the transition probabilities

The derivative of $U(\theta, \hat{\theta})$ with respect to the transition probabilities $b_{\alpha\beta}^\gamma$ is

$$\frac{\partial U}{\partial b_{\alpha\beta}^\gamma} = \frac{\partial}{\partial b_{\alpha\beta}^\gamma} E\big[\log p(\mathcal{X}, \mathcal{K}|\theta)\big|\mathcal{X}, \hat{\theta}\big] + \frac{\partial}{\partial b_{\alpha\beta}^\gamma}\log p(\theta). \tag{1.60}$$

For the expectation we have from (1.38) that

$$\begin{aligned}\frac{\partial}{\partial b_{\alpha\beta}^\gamma} E\big[\log p(\mathcal{X}, \mathcal{K}|\theta)\big|\mathcal{X}, \hat{\theta}\big] &= \frac{\partial}{\partial b_{\alpha\beta}^\gamma} \sum_{s=1}^{S}\sum_{t=1}^{L_s}\sum_{i,j=1}^{K} w_{ij}^s(t)\log\Big(\sum_{n=1}^{N} b_{ij}^n \phi_n(x_{t-1}^s)\Big) \\ &= \sum_{s=1}^{S}\sum_{t=1}^{L_s} w_{\alpha\beta}^s(t)\frac{1}{\sum_{n=1}^N b_{\alpha\beta}^n\phi_n(x_{t-1}^s)}\phi_\gamma(x_{t-1}^s).\end{aligned} \tag{1.61}$$

If the Jeffreys prior is used, then we have (see appendix A.2):

$$\begin{aligned}\frac{\partial}{\partial b_{\alpha\beta}^\gamma}\log p(\theta) &= \frac{\partial}{\partial b_{\alpha\beta}^\gamma}\log\frac{1}{\prod_{j=1}^K\sqrt{b_{ij}^n}} \\ &= \frac{1}{2}\left(\frac{1}{b_{i\beta}^n} - \frac{1}{b_{iK}^n}\right)\delta_\alpha^i\delta_\gamma^n, \quad \beta = 1, \ldots, K-1.\end{aligned} \tag{1.62}$$

Optimization is done using the natural gradient algorithm. The natural gradient ∇U is defined as the solution of the equation

$$\langle\nabla U, w\rangle = \mathrm{d}U(w) \tag{1.63}$$

for all vectors w. The differential $\mathrm{d}U$ is the covector whose components are given by the partial derivatives obtained in (1.61) and (1.62), and the inner product $\langle\cdot,\cdot\rangle$ is the one defined by the Fisher information metric.

FIXME: ESTOU AQUI!!!

Unfortunately, the equation $\partial E/\partial b_{\alpha,\beta}^\gamma$ does not seem to have an explicit solution, thus requiring an iterative algorithm to solve it.

These partial derivatives define the differential $\mathrm{d}E$ of the function to optimize. The computation of the natural gradient is done as follows:

$$\tilde{\nabla}E = B^n \odot \nabla' E - (B^n \cdot \nabla' E)B^n \tag{1.64}$$

1.9 Model fitness

In this section we compute the probability that the data $\mathcal{X}$ had been generated by a given model parameters θ, *i.e.* the likelihood of θ given $\mathcal{X}$. The likelihood allows the comparison of different estimated models for the best candidate.

1.9.1 Single vector field

We start by considering a single vector field. In this case the active field is not required since there is only one possibility. We assume the data to be generated by

$$x_t = x_{t-1} + T(x_{t-1}) + w_t. \tag{1.65}$$

This equation can be written in a probabilistic way by

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t|x_{t-1} + T(x_{t-1}), \mathbf{\Sigma}). \tag{1.66}$$

Then, the probability of observing a given trajectory x is

$$\begin{aligned} p(x|\theta) &= p(x_1, \dots, x_L|x_0, \theta) \\ &= \prod_{t=1}^{L} p(x_t|x_{t-1}, \theta). \end{aligned} \tag{1.67}$$

Taking the logarithm yields

$$\begin{aligned} \log p(x|\theta) &= \sum_{t=1}^{L} \log p(x_t|x_{t-1}, \theta) \\ &= -\frac{1}{2}\sum_{t=1}^{L} \big(x_t - x_{t-1} - T(x_{t-1})\big)^T \mathbf{\Sigma}^{-1} \big(x_t - x_{t-1} - T(x_{t-1})\big) + \\ &\quad + L \log \frac{1}{\sqrt{(2\pi)^d \det \mathbf{\Sigma}}} \end{aligned} \tag{1.68}$$

where the second term is a constant depending on the noise covariance matrix $\mathbf{\Sigma}$ and the trajectory length L.

Equation (1.68) shows that the likelihood can be computed as the distance between the data related velocity vectors $v_t \triangleq x_t - x_{t-1}$ and the estimated vector field $T(x_{t-1})$, using $\mathbf{\Sigma}^{-1}$ as the metric matrix. I.e.,

$$\log p(x|\theta) = -\frac{Ld}{2} \log(2\pi \det \mathbf{\Sigma}) - \frac{1}{2} \sum_{t=1}^{L} \|v_t - T(x_{t-1})\|_{\mathbf{\Sigma}^{-1}} \tag{1.69}$$

1.9.2 Multiple vector fields, multiple trajectories

When multiple vector fields are available, the hidden variables k_t^s for each trajectory s are not known. Trying to obtain the likelihood $p(\mathcal{X}|\theta)$ yields

$$p(\mathcal{X}|\theta) = \sum_{\mathcal{K}} p(\mathcal{X}, \mathcal{K}|\theta) \tag{1.70}$$

which is unfeasible. Fortunately, it is possible to use the forward part of the forward-backward algorithm to perform this computation efficiently.

Consider the forward variable

$$\alpha_t^s(i) \triangleq \Pr\{x_{1:t}^s, k_t^s = i|\hat{\theta}\}. \tag{1.71}$$

Then, the likelihood (1.70) can be computed by taking the marginalization with respect to only the last hidden state k:

$$p(\mathcal{X}|\theta) = \sum_{i=1}^{K} \Pr\left\{x^s_{1:L_s}, k^s_{L_s} = i\middle|\theta\right\} = \sum_{i=1}^{K} \alpha^s_{L_s}(i). \tag{1.72}$$

The forward variable $\alpha^s_{L_s}$ is computed as follows:

1. Initialization:
$$\alpha^s_1(i) = \Pr\left\{k^s_1 = i\middle|\theta\right\}. \tag{1.73}$$

2. Induction
$$\alpha^s_t(i) = \mathcal{N}\left(x^s_t|x^s_{t-1} + T_i(x^s_{t-1}), \Sigma_{k^s_t}\right) \sum_{j=1}^{K} b_{ij}(x^s_{t-1})\alpha^s_{t-1}(j). \tag{1.74}$$

   See section 1.8.2 for details.

   To consider multiple trajectories $s = 1, \ldots, S$, we assume that they are independently generated. Then, their joint likelihood is simply the product of their individual likelihoods (and the log-likelihood is just the sum).

Appendix A

Detailed computations

A.1 Jeffrey's prior

Consider a categorical distribution (N-sided biased die) with probabilities $b = \big[b_1, \dots, b_K\big]$. When estimating the probabilities b_i the prior distribution $p(b)$ is often required. It is a probability density function on the simplex defined by the probability constraints $0 \leq b_i \leq 1$ and $\sum_{j=1}^{K} b_j = 1$. Because of the later constraint, there are only $K-1$ independent parameters b_i. The parameters $b_1, \dots, b_{K-1}$ will be considered independent, while the dependent probability is obtained by $b_K = 1 - \sum_{j=1}^{K-1} b_j$.

The Jeffreys prior is the probability density function defined by

$$p(b_i) \propto \sqrt{\det \mathbf{G}} \tag{A.1}$$

where $\mathbf{G}(b) = \big[g_{ij}(b)\big]$ is the Fisher information matrix defined by

$$g_{ij} \triangleq E\Big[\frac{\partial \log b_k}{\partial b_i}\frac{\partial \log b_k}{\partial b_j}\Big], \quad 1 \leq i, j \leq K-1, \quad 1 \leq k \leq K, \tag{A.2}$$

which in matrix notation yields

$$\mathbf{G} = \begin{bmatrix} \frac{1}{b_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{1}{b_{K-1}} \end{bmatrix} + \frac{1}{1 - \sum_{i=1}^{K-1} b_i} \mathbf{1}\mathbf{1}^T. \tag{A.3}$$

Then, multiplying $\mathbf{G}$ on the right by the identity $\mathbf{I} = \mathbf{D}\mathbf{D}^{-1}$, with $\mathbf{D} \triangleq \operatorname{diag}(b_1, \dots, b_{K-1})$, yields

$$\begin{aligned} \det \mathbf{G} &= \det(\mathbf{G}\mathbf{D}\mathbf{D}^{-1}) = \det(\mathbf{G}\mathbf{D}) \det(\mathbf{D}^{-1}) \\ &= \det(\mathbf{G}\mathbf{D}) \frac{1}{\prod_{i=1}^{K-1} b_i} \\ &= \det\Big(\mathbf{I} + \frac{1}{1 - \sum_{i=1}^{K-1} b_i} \mathbf{1} \begin{bmatrix} b_1 & \cdots & b_{K-1} \end{bmatrix}\Big) \frac{1}{\prod_{i=1}^{K-1} b_i} \\ &= \Big(1 + \frac{1}{1 - \sum_{i=1}^{K-1} b_i} \begin{bmatrix} b_1 & \cdots & b_{K-1} \end{bmatrix} \mathbf{1}\Big) \frac{1}{\prod_{i=1}^{K-1} b_i} \\ &= \Big(1 + \frac{\sum_{i=1}^{K-1} b_i}{1 - \sum_{i=1}^{K-1} b_i}\Big) \frac{1}{\prod_{i=1}^{K-1} b_i} \\ &= \frac{1}{\Big(1 - \sum_{i=1}^{K-1} b_i\Big) \prod_{i=1}^{K-1} b_i} \\ &= \frac{1}{\prod_{i=1}^{K} b_i}. \end{aligned} \tag{A.4}$$

Then, the Jeffreys prior for the categorical distribution is given by

$$p(b) \propto \sqrt{\det \mathbf{G}} = \frac{1}{\prod_{i=1}^{K} \sqrt{b_i}}. \tag{A.5}$$

A.2 Derivative of the Jeffrey's prior

The same categorical distribution $b = \begin{bmatrix} b_1 & \cdots & b_K \end{bmatrix}$ used in the previous section is used here. The derivative of the logarithm of the Jeffrey's prior is computed next:

$$\begin{aligned}
\frac{\partial \log p(b)}{\partial b_\beta} &= \frac{\partial}{\partial b_\beta} \log \frac{1}{\prod_{j=1}^{K} \sqrt{b_j}} \\
&= \left(\prod_{j=1}^{K} \sqrt{b_j} \right) \left(- \frac{1}{\left(\prod_{j=1}^{K} \sqrt{b_j} \right)^2} \frac{\partial}{\partial b_\beta} \prod_{j=1}^{K} \sqrt{b_j} \right) \\
&= - \frac{1}{\prod_{j=1}^{K} \sqrt{b_j}} \frac{\partial}{\partial b_\beta} \left[\sqrt{1 - \sum_{j=1}^{K-1} b_j} \prod_{j=1}^{K-1} \sqrt{b_j} \right] \\
&\dots \\
&= \frac{1}{2} \left(\frac{1}{b_\beta} - \frac{1}{b_K} \right), \qquad \beta = 1, \dots, K-1.
\end{aligned} \tag{A.6}$$

or in matrix notation

$$\frac{\partial \log p(b)}{\partial b} = \frac{1}{2} \left(\begin{bmatrix} \frac{1}{b_1} & \cdots & \frac{1}{b_{K-1}} \end{bmatrix} - \frac{1}{b_K} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \right). \tag{A.7}$$