

# EM Algorithm Updates for Dimensionality Reduction Using Automatic Supervision

Anelia Angelova  
Computer Science Department  
California Institute of Technology  
anelia@caltech.edu

This note derives the E-step and M-step of the EM algorithm [4], for the purposes of *dimensionality reduction with automatic supervision* presented in [1]. The algorithm extends the Mixture of Factor Analyzers (MoFA) framework [6] to allow for the mechanical measurements to act as supervision to the vision-based dimensionality reduction. We first formulate the problem and briefly describe the proposed solution [1]. We then derive the EM-updates in detail and describe how to impose monotonic constraints and apply regularization. Some of the derivations are related to the EM updates for Mixture of Gaussians [2], [5] and for Mixture of Experts [8].

## 1 Problem formulation

Consider the problem of predicting the mobility characteristics  $Z = F(\mathbf{x}, \mathbf{y})$  of the robot in each map cell on the forthcoming terrain using as input the visual information  $\mathbf{x} \in \Omega$  of the cell and some information about the terrain geometry  $\mathbf{y} \in \Phi$ , e.g. local terrain slope ( $\Omega$  is the visual space,  $\Phi$  is the space of terrain slopes). Because of the physical nature of the problem, we can assume that there are a limited number ( $K$ ) of terrain types, that can be encountered, and that on each terrain the robot experiences different behavior (e.g. rover mobility):

$$F(\mathbf{x}, \mathbf{y}) = f_j(\mathbf{y}), \quad \text{if } \mathbf{x} \in \Omega_j \quad (1)$$

where  $\Omega_j \in \Omega$  are different subsets in the visual space,  $\Omega_i \cap \Omega_j = \emptyset, i \neq j$  and  $f_j(\mathbf{y})$  is a (nonlinear) function which represents the rover mobility on the  $j$ -th terrain. In other words, different mobility behaviors occur on different terrain types which are determined by visual information. We use rover slippage as a measure of (lack of) rover mobility. The goal is to learn the function  $Z = F(\mathbf{x}, \mathbf{y})$  from the available training data  $D = \{\mathbf{x}_i, \mathbf{y}_i, z_i\}_{i=1}^N$ . where  $\mathbf{x}_i$  are the visual representations of patches from the observed terrain;  $\mathbf{y}_i$  are the terrain slopes,  $z_i$  are the particular slip measurements when the robot traverses that terrain. Note that the slip and slope measurements  $\mathbf{y}_i, z_i$ , which are assumed to have

come from one of several unknown nonlinear models, act as the only supervision to the whole system which needs to infer the classification in the input space as well as learn the models. We assume that the slip functions have a particular parametric form so that the problem can be reduced to parameter estimation.

In this setup we are going to use the robot’s automatic slip measurements as supervision to the vision-based learning. The intuition is that two visually similar terrains which might not be normally discriminated in the visual space, will be discriminated after introducing the supervision. The motivation for using the slip measurements as supervision, instead of manual labeling of terrain types, is that they are obtained automatically and effortlessly by the robot’s sensors.

The main problem regarding the supervision, however, is that the slip signal to be used as supervision can be of very weak form. In particular, because of the nonlinearity of the slip models  $f_i(\mathbf{y})$ , it is possible that some of the models overlap in parts of their domain. (i.e. for some  $i, j, i \neq j$ ,  $f_i(\mathbf{y}) \equiv f_j(\mathbf{y})$ , for  $\mathbf{y} \in \Phi_0$ , for some  $\Phi_0 \subseteq \Phi$ ). That is, using slip measurements as supervision is not a trivial extension of supervised learning (this is because the slip supervision signals cannot be meaningfully clustered to provide terrain class labels). However, this form of supervision can still provide useful information for discrimination. The intuition is that, although the supervision might not always be useful, some of the examples which provide useful supervision information can propagate it to other examples through their visual similarity.

Furthermore, the input space  $X$ , representing the visual data, is usually of high dimension, which impedes working with it. Instead, we work with a lower dimensional embedding  $U$  of the input space  $X$ . For that purpose we need to learn the embedding  $R: X \rightarrow U$  itself. As the learning of this mapping requires prohibitive amount of data, whenever the input is high dimensional, we assume that it takes a particular form, as in [6]. Namely:

$$\mathbf{x} = \Lambda_j \mathbf{u} + \mathbf{v}_j \quad \text{for } \mathbf{x} \in \Omega_j \tag{2}$$

where  $\mathbf{v}_j$  are normally distributed ( $\mathbf{v}_j \sim \mathcal{N}(\eta_j, \Psi_j)$ ) and  $\Lambda_j$  is a projection matrix. That is, we assume that a locally linear mapping is a good enough approximation within terrain patches that belong to the same terrain class.

## 2 Automatically supervised dimensionality reduction

The problem defined in Section 1 is formulated and solved in a probabilistic framework which performs dimensionality reduction and terrain classification by using automatic supervision and which can cope with both noisy and ambiguous supervision [1]. In this way the automatic supervision influences the selection of appropriate low dimensional visual representations and helps learn the distinction between different terrain classes.

In particular, indicator variables  $L_{ij}$  are introduced which can do the decoupling between the vision part, in which dimensionality reduction (or learning of

a lower dimensional representation) is done, and the mechanical behavior part, in which the slip measurements act as supervision. The indicator variables  $L$  are *latent*, i.e. hidden, and are added to simplify the inference process [2]. They define the class-membership of each training example i.e.  $L_{ij} = 1$  if the  $i^{th}$  training example  $(\mathbf{x}_i, \mathbf{y}_i, z_i)$  has been generated by the  $j^{th}$  nonlinear slip model and belongs to the  $j^{th}$  terrain class. As an additional step, a dimensionality reduction of the visual part of the data is done, so now the supervision can affect the parameters related to the dimensionality reduction too. This essentially means preferring projections which fit the data well, and therefore also the supervision. Now, given the labeling of the example is known, the slip supervision measurements and the visual information are independent. The complete likelihood factors as follows:

$$P(X, U, Y, Z, L|\Theta) = \underbrace{P(X|U, L, \Theta)P(U|L, \Theta)}_{\text{Vision part, dim. red.}} \underbrace{P(Y, Z|L, \Theta)}_{\text{Autom. supervision}} \underbrace{P(L|\Theta)}_{\text{Prior}}$$

where  $\Theta = \{\mu_j, \Sigma_j, \Lambda_j, \eta_j, \Psi_j, \theta_j, \sigma_j, \pi_j\}_{j=1}^K$  contains all the parameters that need to be estimated in the system. The parameters  $\mu_j$  and  $\Sigma_j$  are the means and the covariance matrices of the lower dimensional representation  $U$  for each class,  $\Lambda_j$  are the projection matrices,  $\eta_j$  and  $\Psi_j$  are the means and covariances of the input data (see Equation (2)),  $\theta_j$  are the parameters of the nonlinear fit of the slip data,  $\sigma_j$  are their covariances (here they are the standard deviations, as the final measurement is one dimensional), and  $\pi_j$  are the prior probabilities of each class. The graphical model corresponding to the problem is shown in Figure 1. Maximum likelihood estimation is used to learn the parameters of each class and the classification boundaries between them. The proposed model allows the automatically obtained mechanical supervision to affect both the dimensionality reduction and the clustering process, thus improving a purely unsupervised learning for the purposes of the task at hand. Note that here the lower dimensional representation  $U$  is hidden and that the supervision part can influence the visual learning and the lower dimensional projections through the latent variables  $L_{ij}$ .

With the help of the hidden variables  $L$ , the complete log likelihood function ( $CL$ ) for the whole data can be written as:

$$CL(X, U, Y, Z, L|\Theta) = \sum_{i=1}^N \sum_{j=1}^K L_{ij} [\log P(\mathbf{x}_i|\mathbf{u}_i, L_{ij} = 1, \Lambda_j, \eta_j, \Psi_j) + \log P(\mathbf{u}_i|L_{ij} = 1, \mu_j, \Sigma_j) + \log P(\mathbf{y}_i, z_i|L_{ij} = 1, \theta_j, \sigma_j) + \log \pi_j]$$

## 2.1 Obtaining the lower dimensional representation

The Mixture of Factor Analyzers (MoFA) formulation [6] is used to model the distribution of the lower dimensional representation  $U$  and the initial visual

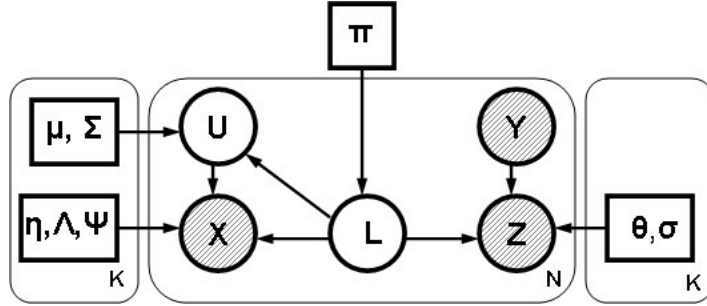


Figure 1: Graphical model of automatically supervised dimensionality reduction in which mechanical measurements obtained from the robot are used as supervision. The observed random variables are displayed in shaded circles.

space  $X$ . As previously stated in Equation 2, it is assumed that  $\{X|U, C = j\} \sim \mathcal{N}(\Lambda_j U + \eta_j, \Psi_j)$  and  $U \sim \mathcal{N}(\mu_j, \Sigma_j)$ . In other words, the joint probability of  $X$  and  $U$  is assumed to be modeled as a mixture of  $K$  local linear projections, or factors [6]. Note that the variables  $U$  are latent. After introducing the auxiliary variables  $L_{ij}$  we can write the probability of a data point  $\mathbf{x}_i$  belonging to a terrain class  $j$ , given the latent representation  $\mathbf{u}_i$ , and the probability of the latent representation  $\mathbf{u}_i$ , given the class  $j$  as:

$$P(\mathbf{x}_i|\mathbf{u}_i, L_{ij} = 1) = \frac{e^{-\frac{1}{2}(\mathbf{x}_i - \Lambda_j \mathbf{u}_i - \eta_j)^T \Psi_j^{-1} (\mathbf{x}_i - \Lambda_j \mathbf{u}_i - \eta_j)}}{(2\pi)^{D/2} |\Psi_j|^{1/2}}$$

$$P(\mathbf{u}_i|L_{ij} = 1) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{u}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{u}_i - \mu_j)},$$

where  $D$  and  $d$  are the dimensionalities of the initial visual space and the projected representation, respectively. Those distributions are modeled, so that a tractable solution to the maximum likelihood problem is achieved.

## 2.2 Automatic supervision

The supervision part is described as follows. The mechanical measurement data is assumed to have come from a nonlinear fit, which is modeled as a General Linear Regression (GLR) [9]. GLR is appropriate for expressing nonlinear behavior and is convenient for computation because it is linear in terms of the parameters to be estimated. For each terrain type  $j$ , the regression function  $\tilde{Z}(Y) = E(Z|Y)$  is assumed to have come from a GLR with Gaussian noise:  $f_j(Y) \equiv Z(Y) = \tilde{Z}(Y) + \epsilon_j$ , where  $\tilde{Z}(Y) = \theta_j^0 + \sum_{r=1}^R \theta_j^r g_r(Y)$ ,  $\epsilon_j \sim \mathcal{N}(0, \sigma_j)$ ,  $g_r$  are several nonlinear functions selected before the learning has started. Some example nonlinear functions to be used as building blocks for slip approximation are:  $x$ ,  $x^2$ ,  $e^x$ ,  $\log x$ ,  $\tanh x$ . The parameters  $\theta_j^0, \dots, \theta_j^R, \sigma_j$  need to be learned

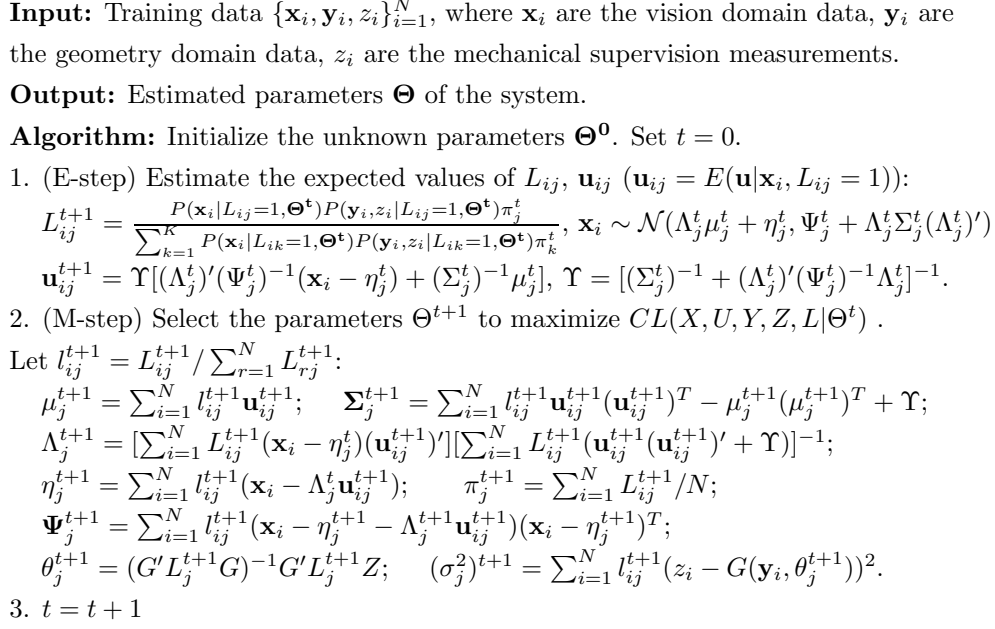


Figure 2: EM algorithm updates.

for each model  $j$ . The probability of  $z_i$  belonging to the  $j^{th}$  nonlinear model, conditioned on  $\mathbf{y}_i$ , is:

$$P(z_i|\mathbf{y}_i, L_{ij} = 1, \theta_j, \sigma_j) = \frac{1}{(2\pi)^{1/2}\sigma_j} e^{-\frac{1}{2\sigma_j^2}(z_i - G(\mathbf{y}_i, \theta_j))^2},$$

where  $G(\mathbf{y}, \theta_j) = \theta_j^0 + \sum_{r=1}^R \theta_j^r g_r(\mathbf{y})$  and  $\theta_j = (\theta_j^0, \theta_j^1, \dots, \theta_j^R)$ .  $P(\mathbf{y}_i)$  is given an uninformative (here, uniform over a range of slopes) prior.

### 3 EM updates

This section describes in details the updates for the EM algorithm to maximize the likelihood [4]. A summary of the algorithm updates is given in Figure 2. Here we consider the updates done at step  $t + 1$ . The algorithm performs the following steps until convergence:

In the E-step, the expected values of the unobserved variables  $U$  and label assignments  $L$  are estimated. In the M-step, the parameters for both the vision and the supervision side are selected, so as to maximize the complete log likelihood. Because of the conditional independence, the parameters for the vision and mechanical side are selected independently in the M-step. however, both sides interact through the labels, as they both provide information for the estimation done in the E-step.

### 3.1 E-step

Let us denote the following random variable  $\tilde{\mathbf{u}}_{ij} = \{\mathbf{u}|\mathbf{x}_i, E_{ij} = 1\}$ . Denote  $\mathbf{u}_{ij} = E[\tilde{\mathbf{u}}_{ij}] = E[\{\mathbf{u}|\mathbf{x}_i, L_{ij} = 1\}]$  ( $\mathbf{u}_{ij}^{t+1}$  is the expected value in the current iteration). In the E-step, the expected values of the hidden variables  $U$  and  $L$  are estimated with respect to the distribution  $P(L, U|X, Y, Z, \Theta^t)$ . Since  $E(\mathbf{u}, L_{ij}|\mathbf{x}_i, \mathbf{y}_i, z_i) = E(L_{ij}|\mathbf{x}_i, \mathbf{y}_i, z_i)E(\mathbf{u}|\mathbf{x}_i, \mathbf{y}_i, z_i, L_{ij})$ , we evaluate  $E(L_{ij} = 1|\mathbf{x}_i, \mathbf{y}_i, z_i)$  and  $E(\mathbf{u}|\mathbf{x}_i, \mathbf{y}_i, z_i, L_{ij} = 1)$  independently. Since  $U$  is independent of  $Y$  or  $Z$  given  $L$ ,  $E(\mathbf{u}|\mathbf{x}_i, \mathbf{y}_i, z_i, L_{ij} = 1) = E(\mathbf{u}|\mathbf{x}_i, L_{ij} = 1) = \mathbf{u}_{ij}$ .

#### 3.1.1 E-step for $L_{ij}$ :

$$\begin{aligned} L_{ij}^{t+1} &= E[L_{ij}]_{P(L|X, Y, Z, \Theta^t)} = \\ &= 0 \cdot P(L_{ij} = 0|\mathbf{x}_i, \mathbf{y}_i, z_i, \Theta^t) + 1 \cdot P(L_{ij} = 1|\mathbf{x}_i, \mathbf{y}_i, z_i, \Theta^t) = \\ &= P(L_{ij} = 1|\mathbf{x}_i, \mathbf{y}_i, z_i, \Theta^t) \end{aligned}$$

This can be evaluated as follows:

$$\begin{aligned} P(L_{ij} = 1|\mathbf{x}_i, \mathbf{y}_i, z_i, \Theta^t) &= \frac{P(\mathbf{x}_i, \mathbf{y}_i, z_i|L_{ij} = 1, \Theta^t)P(L_{ij} = 1)}{\sum_{k=1}^K P(\mathbf{x}_i, \mathbf{y}_i, z_i|L_{ik} = 1, \Theta^t)P(L_{ik} = 1)} = \\ &= \frac{P(\mathbf{x}_i|L_{ij} = 1, \Theta^t)P(\mathbf{y}_i, z_i|L_{ij} = 1, \Theta^t)\pi_{ij}}{\sum_{k=1}^K P(\mathbf{x}_i|L_{ik} = 1, \Theta^t)P(\mathbf{y}_i, z_i|L_{ik} = 1, \Theta^t)\pi_{ik}} \end{aligned}$$

That is,

$$L_{ij}^{t+1} = \frac{P(\mathbf{x}_i|L_{ij} = 1, \Theta^t)P(\mathbf{y}_i, z_i|L_{ij} = 1, \Theta^t)\pi_{ij}}{\sum_{k=1}^K P(\mathbf{x}_i|L_{ik} = 1, \Theta^t)P(\mathbf{y}_i, z_i|L_{ik} = 1, \Theta^t)\pi_{ik}} \quad (3)$$

To evaluate the above expression we need to know the distribution of the random variable  $\{\mathbf{x}_i|L_{ij} = 1\}$ . Since we know the parametric distributions of  $\{\mathbf{u}|L_{ij} = 1\} \sim \mathcal{N}(\mu_j^t, \Sigma_j^t)$  and  $\{\mathbf{x}_i|\mathbf{u}, L_{ij} = 1\} \sim \mathcal{N}(\Lambda_j^t \mathbf{u} + \eta_j^t, \Psi_j^t)$ , and since they are Gaussian,  $\{\mathbf{x}_i|L_{ij} = 1\}$  is also Gaussian and its mean and covariance matrix can be expressed as follows [3]:

$$\{\mathbf{x}_i|L_{ij} = 1\} \sim \mathcal{N}(\Lambda_j^t \mu_j^t + \eta_j^t, \Psi_j^t + \Lambda_j^t \Sigma_j^t (\Lambda_j^t)'), \quad (4)$$

So,  $p(\mathbf{x}_i|L_{ij} = 1)$  can be computed as follows:

$$P(\mathbf{x}_i|L_{ij} = 1) = \frac{e^{-\frac{1}{2}(\mathbf{x}_i - \Lambda_j^t \mu_j^t - \eta_j^t)^T (\Psi_j^t + \Lambda_j^t \Sigma_j^t (\Lambda_j^t)')^{-1} (\mathbf{x}_i - \Lambda_j^t \mu_j^t - \eta_j^t)}}{(2\pi)^{D/2} |\Psi_j^t + \Lambda_j^t \Sigma_j^t (\Lambda_j^t)'|^{1/2}} \quad (5)$$

### 3.1.2 E-step for $\mathbf{u}_{ij}$ :

We obtain the following update for  $\mathbf{u}_{ij}^{t+1}$  (where  $\mathbf{u}_{ij}^{t+1} = E(\mathbf{u}|\mathbf{x}_i, L_{ij} = 1, \Theta^t)$ ):

$$\mathbf{u}_{ij}^{t+1} = [(\Sigma_j^t)^{-1} + (\Lambda_j^t)'(\Psi_j^t)^{-1}\Lambda_j^t]^{-1}[(\Lambda_j^t)'(\Psi_j^t)^{-1}(\mathbf{x}_i - \eta_j^t) + (\Sigma_j^t)^{-1}\mu_j^t], \quad (6)$$

This is because, similarly to the previous step, the expected value and variance of the random variable  $\{\mathbf{u}|\mathbf{x}_i, L_{ij} = 1\}$  can be represented as a function of the means and covariances of the Gaussian random variables  $\{\mathbf{u}|L_{ij} = 1\}$  and  $\{\mathbf{x}_i|\mathbf{u}, L_{ij} = 1\}$ . By using the fact that  $\{\mathbf{u}|L_{ij} = 1\} \sim \mathcal{N}(\mu_j^t, \Sigma_j^t)$  and  $\{\mathbf{x}_i|\mathbf{u}, L_{ij} = 1\} \sim \mathcal{N}(\Lambda_j^t\mathbf{u} + \eta_j^t, \Psi_j^t)$ , it follows that [3]:

$$\tilde{\mathbf{u}}_{ij} = \{\mathbf{u}|x_i, L_{ij} = 1\} \sim \mathcal{N}(\Upsilon[(\Lambda_j^t)'(\Psi_j^t)^{-1}(\mathbf{x}_i - \eta_j^t) + (\Sigma_j^t)^{-1}\mu_j^t], \Upsilon), \quad (7)$$

where  $\Upsilon = [(\Sigma_j^t)^{-1} + (\Lambda_j^t)'(\Psi_j^t)^{-1}\Lambda_j^t]^{-1}$ . Note that  $Var(\tilde{\mathbf{u}}_{ij}) = \Upsilon$ . Using the matrix inversion lemma [3]:

$$\Upsilon = \Sigma_j^t - \Sigma_j^t(\Lambda_j^t)'[\Psi_j^t + \Lambda_j^t\Sigma_j^t(\Lambda_j^t)']^{-1}\Lambda_j^t\Sigma_j^t, \quad (8)$$

which is a more convenient representation for numerical computation purposes.

## 3.2 M-step

In the M-step, the expected value of the complete log likelihood function is maximized (using the expected values  $\mathbf{u}_{ij}^{t+1}$  and  $L_{ij}^{t+1}$  computed from the E-step). Note that some groups of parameters can be optimized independently of the others. For example, the parameters of the visual information side  $\mu_j$ ,  $\Sigma_j$ ,  $\eta_j$ ,  $\Lambda_j$ ,  $\Psi_j$  can be optimized independently from the parameters of the supervision side  $\theta_j$ ,  $\sigma_j$ . The same applies to the priors  $\pi_j$ . The parameters for different  $j$ 's can also be optimized independently. In the following equations we denote  $l_{ij}^{t+1} = L_{ij}^{t+1} / \sum_{r=1}^N L_{rj}^{t+1}$ .

### 3.2.1 M-step for $\mu_j$ , $\Sigma_j$ :

Taking the derivatives with respect to the parameters  $\mu_j$ :

$$\begin{aligned} \frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \mu_j} &= -\frac{1}{2} \sum_{i=1}^N L_{ij}^{t+1} \frac{\partial E[(\tilde{\mathbf{u}}_{ij} - \mu_j)^T \Sigma_j^{-1} (\tilde{\mathbf{u}}_{ij} - \mu_j)]}{\partial \mu_j} = 0 \\ \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1} - \mu_j)^T \Sigma_j^{-1} &= 0 \quad \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1} - \mu_j)^T = 0 \end{aligned}$$

Now, solving for  $\mu_j$ , we get:

$$\mu_j^{t+1} = \frac{\sum_{i=1}^N L_{ij}^{t+1} \mathbf{u}_{ij}^{t+1}}{\sum_{i=1}^N L_{ij}^{t+1}} = \sum_{i=1}^N l_{ij}^{t+1} \mathbf{u}_{ij}^{t+1} \quad (9)$$

Taking the derivatives with respect to the parameters  $\Sigma_j^{-1}$  (for convenience):

$$\begin{aligned}
& \frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \Sigma_j^{-1}} = 0 \\
& \Rightarrow -\frac{1}{2} \sum_{i=1}^N L_{ij}^{t+1} \left\{ \frac{\partial \log |\Sigma_j^{-1}|}{\partial \Sigma_j^{-1}} - \frac{\partial E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T \Sigma_j^{-1} (\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})]}{\partial \Sigma_j^{-1}} \right\} = 0 \\
& \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} \left\{ \frac{\partial \log |\Sigma_j^{-1}|}{\partial \Sigma_j^{-1}} - \frac{\partial Tr\{\Sigma_j^{-1} E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T]\}}{\partial \Sigma_j^{-1}} \right\} = 0 \\
& \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} \{(\Sigma_j^{-1})^{-1} - \sum_{i=1}^N L_{ij}^{t+1} E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T]\} = 0 \\
& \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} \{\Sigma_j - E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T]\} = 0
\end{aligned}$$

From which follows:

$$\Sigma_j = \frac{\sum_{i=1}^N L_{ij}^{t+1} E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T]}{\sum_{i=1}^N L_{ij}^{t+1}} = \sum_{i=1}^N l_{ij}^{t+1} E[(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})(\tilde{\mathbf{u}}_{ij} - \mu_j^{t+1})^T] \quad (10)$$

Since  $E[\tilde{\mathbf{u}}_{ij}] = \mathbf{u}_{ij}^{t+1}$  by definition, now we need to estimate  $E[\tilde{\mathbf{u}}_{ij}(\tilde{\mathbf{u}}_{ij})^T]$ . Since  $Var(\mathbf{u}) = E(\mathbf{u}\mathbf{u}^T) - E(\mathbf{u})E(\mathbf{u})^T$ , we can compute  $E(\tilde{\mathbf{u}}_{ij}(\tilde{\mathbf{u}}_{ij})^T) = Var(\tilde{\mathbf{u}}_{ij}) + E(\tilde{\mathbf{u}}_{ij})E(\tilde{\mathbf{u}}_{ij})^T = \Upsilon + E(\tilde{\mathbf{u}}_{ij})E(\tilde{\mathbf{u}}_{ij})^T$ , since  $Var(\tilde{\mathbf{u}}_{ij}) = \Upsilon$  from Equation (7). Finally, using Equation 9, we obtain<sup>1</sup>:

$$\Sigma_j^{t+1} = \sum_{i=1}^N l_{ij}^{t+1} \mathbf{u}_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1})^T - \mu_j^{t+1} (\mu_j^{t+1})^T + \Upsilon \quad (11)$$

### 3.2.2 M-step for $\eta_j$ :

Taking the derivatives with respect to the parameters  $\eta_j$ :

$$\begin{aligned}
& \frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \eta_j} = 0 \\
& \Rightarrow -\frac{1}{2} \sum_{i=1}^N L_{ij}^{t+1} \frac{\partial E[(\mathbf{x}_i - \Lambda_j^t \tilde{\mathbf{u}}_{ij} - \eta_j)^T \Psi_j^{-1} (\mathbf{x}_i - \Lambda_j^t \tilde{\mathbf{u}}_{ij} - \eta_j)]}{\partial \eta_j} = 0 \\
& \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \Lambda_j^t \mathbf{u}_{ij}^{t+1} - \eta_j)' \Psi_j^{-1} = 0
\end{aligned}$$

---

<sup>1</sup>It is more convenient here to use the updated version of the parameter  $\mu_j$ .

$$\Rightarrow \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \Lambda_j^t \mathbf{u}_{ij}^{t+1} - \eta_j)^t = 0$$

Now, solving for  $\eta_j$ , we get:

$$\eta_j^{t+1} = \frac{\sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \Lambda_j^t \mathbf{u}_{ij}^{t+1})}{\sum_{i=1}^N L_{ij}^{t+1}} = \sum_{i=1}^N l_{ij}^{t+1} (\mathbf{x}_i - \Lambda_j^t \mathbf{u}_{ij}^{t+1}) \quad (12)$$

### 3.2.3 M-step for $\Lambda_j$ :

Taking the derivatives with respect to the parameters  $\Lambda_j$ :

$$\begin{aligned} & \frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \Lambda_j} = 0 \\ \Rightarrow & -\frac{1}{2} \sum_{i=1}^N L_{ij}^{t+1} \frac{\partial E[(\mathbf{x}_i - \Lambda_j \tilde{\mathbf{u}}_{ij} - \eta_j^t)^T \Psi_j^{-1} (\mathbf{x}_i - \Lambda_j \tilde{\mathbf{u}}_{ij} - \eta_j^t)]}{\partial \Lambda_j} = 0 \\ \Rightarrow & \sum_{i=1}^N L_{ij}^{t+1} \left\{ \frac{\partial E[-2(\mathbf{x}_i - \eta_j^t)' \Psi_j^{-1} \Lambda_j \tilde{\mathbf{u}}_{ij}]}{\partial \Lambda_j} + \frac{\partial E[(\tilde{\mathbf{u}}_{ij})' (\Lambda_j)' \Psi_j^{-1} \Lambda_j \tilde{\mathbf{u}}_{ij}]}{\partial \Lambda_j} \right\} = 0 \\ \Rightarrow & 2 \sum_{i=1}^N L_{ij}^{t+1} \Psi_j^{-1} (\mathbf{x}_i - \eta_j^t) (\mathbf{u}_{ij}^{t+1})' = \sum_{i=1}^N L_{ij}^{t+1} \frac{\partial E\{Tr[(\Lambda_j)' \Psi_j^{-1} \Lambda_j \tilde{\mathbf{u}}_{ij} (\tilde{\mathbf{u}}_{ij})']\}}{\partial \Lambda_j} \\ \Rightarrow & 2 \sum_{i=1}^N L_{ij}^{t+1} \Psi_j^{-1} (\mathbf{x}_i - \eta_j^t) (\mathbf{u}_{ij}^{t+1})' = 2 \sum_{i=1}^N L_{ij}^{t+1} \Psi_j^{-1} \Lambda_j E[\tilde{\mathbf{u}}_{ij} (\tilde{\mathbf{u}}_{ij})'] \\ \Rightarrow & \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \eta_j^t) (\mathbf{u}_{ij}^{t+1})' = \Lambda_j \sum_{i=1}^N L_{ij}^{t+1} E[\tilde{\mathbf{u}}_{ij} (\tilde{\mathbf{u}}_{ij})'] \\ \Rightarrow & \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \eta_j^t) (\mathbf{u}_{ij}^{t+1})' = \Lambda_j \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1})' + \Upsilon) \end{aligned}$$

Now, solving for  $\Lambda_j$ , we get:

$$\Lambda_j^{t+1} = \left[ \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \eta_j^t) (\mathbf{u}_{ij}^{t+1})' \right] \left[ \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1} (\mathbf{u}_{ij}^{t+1})' + \Upsilon) \right]^{-1} \quad (13)$$

### 3.2.4 An alternative M-step for $\Lambda_j$ and $\eta_j$ :

Note that the updates for  $\Lambda_j$  and  $\eta_j$  are made using the values from the previous iteration of the parameters  $\eta_j^t$  and  $\Lambda_j^t$ , respectively. These variables can be also optimized for simultaneously, as shown in [6]. For that purpose an auxiliary

variable  $\tilde{\Lambda}_j = [\Lambda'_j, \eta_j]'$  is formed and the derivatives are taken with respect to the parameter  $\tilde{\Lambda}_j$ . An update of this type can speed up the convergence, but, in general, either of the two types of updates will work fine.

### 3.2.5 M-step for $\Psi_j$ :

Taking the derivatives with respect to the parameters  $\Psi_j^{-1}$  (for convenience):

$$\begin{aligned} \frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \Psi_j^{-1}} &= 0 \quad \Rightarrow \\ \sum_{i=1}^N L_{ij}^{t+1} \left\{ \frac{\partial \log |\Psi_j^{-1}|}{\partial \Psi_j^{-1}} - \frac{\partial E[(\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})^T \Psi_j^{-1} (\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})]}{\partial \Psi_j^{-1}} \right\} &= 0 \\ \Rightarrow \sum_{i=1}^N L_{ij}^{t+1} \{ (\Psi_j^{-1})^{-1} - E[(\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})(\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})^T] \} &= 0 \end{aligned}$$

From which follows that:

$$\begin{aligned} \Psi_j &= \frac{\sum_{i=1}^N L_{ij}^{t+1} E[(\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})(\mathbf{x}_i - \Lambda_j^{t+1} \tilde{\mathbf{u}}_{ij} - \eta_j^{t+1})^T]}{\sum_{i=1}^N L_{ij}^{t+1}} = \\ &= \frac{\sum_{i=1}^N L_{ij}^{t+1} \Lambda_j^{t+1} E[\tilde{\mathbf{u}}_{ij}(\tilde{\mathbf{u}}_{ij})'] (\Lambda_j^{t+1})' - \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \eta_j^{t+1})(\mathbf{u}_{ij}^{t+1})' (\Lambda_j^{t+1})'}{\sum_{i=1}^N L_{ij}^{t+1}} \\ &\quad - \frac{\sum_{i=1}^N L_{ij}^{t+1} \Lambda_j^{t+1} \mathbf{u}_{ij}^{t+1} (\mathbf{x}_i - \eta_j^{t+1})^T - \sum_{i=1}^N L_{ij}^{t+1} (\mathbf{x}_i - \eta_j^{t+1})(\mathbf{x}_i - \eta_j^{t+1})^T}{\sum_{i=1}^N L_{ij}^{t+1}} \end{aligned}$$

By using Equation 13 for the update for  $\Lambda_j$  we can see that the first two components cancel out. So we get:

$$\Psi_j^{t+1} = \sum_{i=1}^N l_{ij}^{t+1} (\mathbf{x}_i - \eta_j^{t+1} - \Lambda_j^{t+1} \mathbf{u}_{ij}^{t+1})(\mathbf{x}_i - \eta_j^{t+1})^T$$

### 3.2.6 M-step for $\pi_j$ :

When taking the derivatives with respect to the parameters  $\pi_j$ , we have to remember that they should satisfy the constraint  $\sum_{i=1}^N \pi_j = 1$ . So we consider the Lagrangian  $CL1 = CL(X, U, Y, Z, L|\Theta) + \lambda(\sum_{i=1}^N \pi_j - 1)$ . Taking derivatives with respect to the unknown parameters  $\pi_j$  and  $\lambda$

$$\frac{\partial E[CL1]}{\partial \pi_j} = \sum_{i=1}^N L_{ij}^{t+1} \frac{1}{\pi_j^{t+1}} + \lambda = 0; \quad \frac{\partial E[CL1]}{\partial \lambda} = \sum_{i=1}^N \pi_j^{t+1} - 1 = 0$$

we obtain

$$\pi_j^{t+1} = -\frac{1}{\lambda} \sum_{i=1}^N L_{ij}^{t+1}; \quad \sum_{i=1}^N \pi_j^{t+1} = 1$$

Now we substitute  $\pi_j^{t+1}$  in the above constraint

$$-\frac{\sum_{j=1}^K \sum_{i=1}^N L_{ij}^{t+1}}{\lambda} = 1 \quad \Rightarrow \quad \lambda = -\frac{1}{\sum_{j=1}^K \sum_{i=1}^N L_{ij}^{t+1}}$$

From which follows that

$$\pi_j^{t+1} = \frac{\sum_{i=1}^N L_{ij}^{t+1}}{\sum_{j=1}^K \sum_{i=1}^N L_{ij}^{t+1}} = \frac{\sum_{i=1}^N L_{ij}^{t+1}}{N} \quad (14)$$

### 3.2.7 M-step for $\theta_j, \sigma_j$ :

$$\frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \theta_j} = \sum_{i=1}^N L_{ij}^{t+1} \frac{1}{2\sigma_j^2} \frac{\partial (z_i - G(\mathbf{y}_i, \theta_j))^2}{\partial \theta_j} = 0$$

Now, let  $L_j$  be a  $N \times N$  matrix which has  $L_{1j}^{t+1}, \dots, L_{Nj}^{t+1}$  on its diagonal,  $G$  be a  $N \times (R+1)$  matrix, such that  $G_{ir} = g_r(\mathbf{y}_i)$ ,  $G_{i(R+1)} = 1$  and  $Z$  be a  $N \times 1$  vector containing the measurements  $z_i$ . The above can be written in an equivalent matrix form (ignoring the term  $\frac{1}{2\sigma_j^2}$ ):

$$\frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \theta_j} = \frac{\partial (Z - G\theta_j)^T L_j (Z - G\theta_j)}{\partial \theta_j} = 0$$

Taking derivatives with respect to  $\theta_j$ , we get

$$\frac{\partial (Z - G\theta_j)^T L_j (Z - G\theta_j)}{\partial \theta_j} = -(Z - G\theta_j)^T L_j G = 0$$

Finally, we receive  $G^T L_j G \theta = G^T L_j Z$  (using the fact that  $L_j$  is a symmetric matrix), from which  $\theta_j^{t+1}$  can be estimated as:

$$\theta_j^{t+1} = (G^T L_j G)^{-1} G^T L_j Z. \quad (15)$$

Taking derivatives with respect to  $\sigma_j$ , we get

$$\frac{\partial E[CL(X, U, Y, Z, L|\Theta)]}{\partial \sigma_j} = \sum_{i=1}^N L_{ij}^{t+1} \left( -\frac{1}{\sigma_j} + \frac{1}{\sigma_j^3} (z_i - G(\mathbf{y}_i, \theta_j^{t+1}))^2 \right)$$

Solving with respect to  $\sigma_j^2$ , we obtain the following update:

$$(\sigma_j^{t+1})^2 = \frac{\sum_{i=1}^N L_{ij}^{t+1} (z_i - G(\mathbf{y}_i, \theta_j^{t+1}))^2}{\sum_{i=1}^N L_{ij}^{t+1}}. \quad (16)$$

## 4 EM updates with monotonic constraints

In some applications, to address specific constraints of real-life observations, some additional constraints might need to be imposed. For example, a slip model could be required to be monotonic, as slip is expected to increase as a function of slopes in real-life observations. This section considers the EM updates when monotonic constraints are imposed on the slip models.

To impose monotonic constraints in the Generalized Linear Regression setup, we can limit the scope to monotonic basis functions only and further constrain the coefficients  $\theta$  to be non-negative:  $\theta \geq 0$ .

### 4.1 M-step for $\theta_j$ :

The updates with monotonic constraints affect the updates on the mechanical side only. We need to maximize the conditional log likelihood (only the portion involving the parameters  $\theta_j$  is shown) with respect to the parameters  $\theta_j$  and  $\sigma_j$ , subject to the constraints  $\theta \geq 0$ :

$$\begin{aligned} \max_{\theta_j} CL2(X, U, Y, Z, L|\Theta) &= - \sum_{i=1}^N L_{ij}^{t+1} \frac{1}{2\sigma_j^2} (z_i - G(\mathbf{y}_i, \theta_j))^2 \\ \text{subj. to: } &\theta_j \geq 0. \end{aligned}$$

Following Equation 18, this can be rewritten as:

$$\min_{\theta_j} (Z - G\theta_j)^T L_j (Z - G\theta_j)$$

$$\text{subj. to: } \theta_j \geq 0.$$

$$\min_{\theta_j} \frac{1}{2} \theta_j^T G^T L_j G \theta_j - Z L G^T \theta_j$$

$$\text{subj. to: } \theta_j \geq 0. \quad (17)$$

which is a quadratic programming problem and can be solved with any numerical analysis package (e.g. the *quadprog* function in MATLAB).

Minimization with respect to  $\sigma_j$  is the same as in Equation 16, with the difference that now we use the values of  $\theta_j^{t+1}$  obtained from Equation 17.

## 5 EM updates with regularization

Real-life data comes with a lot of noise. To adequately respond to noisy data we introduce some regularization. This section considers the EM updates when regularization constraints are imposed on the slip models.

The regularization affects the update of the parameters on the mechanical side only. So, the only modification is made in the M-step for the update of  $\theta_j$ . We minimize the following cost function instead of  $-CL(X, U, Y, Z, L|\Theta)$ :

$$CL3(X, U, Y, Z, L|\Theta) = (Z - G\theta_j)^T L_j (Z - G\theta_j) + \gamma \theta_j^T \theta_j,$$

where  $\gamma \geq 0$  is a regularization parameter. This is equivalent to minimizing  $-CL(X, U, Y, Z, L|\Theta)$  under a constraint of the type  $\sum_{r=1}^R (\theta_j^r)^2 \leq \gamma_0$ , for some  $\gamma_0 > 0$  [10], [7].

### 5.1 M-step for $\theta_j$ :

We take the derivatives with respect to  $\theta_j$ :

$$\frac{\partial E[CL3(X, U, Y, Z, L|\Theta)]}{\partial \theta_j} = -(Z - G\theta_j)^T L_j G + \gamma \theta_j^T = 0 \quad (18)$$

from which  $\theta_j$  is estimated as

$$\theta_j^{t+1} = (G^T L_j G + \gamma I)^{-1} G^T L_j Z. \quad (19)$$

The regularized version of the updates for  $\theta_j$  in the case when monotonic constraints are imposed (Section 4) is as follows:

$$\begin{aligned} \min_{\theta_j} \quad & \frac{1}{2} \theta_j^T (G^T L_j G + \gamma I) \theta_j - Z L G^T \theta_j \\ \text{subj. to:} \quad & \theta_j \geq 0. \end{aligned} \quad (20)$$

## References

- [1] A. Angelova, L. Matthies, D. Helmick, and P. Perona. Dimensionality reduction using automatic supervision for vision-based terrain learning. *Robotics: Science and Systems Conference*, 2007.

- [2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.
- [4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–37, 1977.
- [5] A. D’Souza. Using EM to estimate a probability density with a Mixture of Gaussians. *Technical note*.
- [6] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. *Tech. Report CRG-TR-96-1, Department of Computer Science, University of Toronto*, 1997.
- [7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [8] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.
- [9] G. Seber and C. Wild. *Nonlinear Regression*. John Wiley & Sons, New York, 1989.
- [10] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C., 1977.