Algorithms for Push Prediction

Jeremy L. Wyatt

January 11, 2015

1 Definitions

We first define an ordered set of rigid body transformations at time t as $T^z = T^z(t)$, using the short-hand without the time index for compactness. T^z_f denotes the subset of T^z used for factor f in KDEF. In general there is an invertible mapping between the ordered set $T^z_f(t)$ and a corresponding parameter vector Z^t_f (or Z_f for short) for a given parameterisation of rigid body transformations. It is these parameter vectors we use as kernel centres during learning and prediction. We use $Z^{0:T}_f$ to denote a sequence of parameter vectors, and $\mathcal{Z} = \{Z\}$ to denote a set.

$$X_g \leftrightarrow T^x_g = \{T^{A_t B_t}, T^{B_t O} T^{A_t A_{t+1}}\} \tag{1}$$

$$Y_g \leftrightarrow T^y_g = \{T^{B_t B_{t+1}}\} \tag{2}$$

$$X_a \leftrightarrow T^x_a = \{T^{A^L_t A^L_{t+1}}, T^{A^L_t B^L_t}\} \tag{3}$$

$$Y_a \leftrightarrow T^y_a = \{T^{B^L_t B^L_{t+1}}\} \tag{4}$$

$$X_{e,k} \leftrightarrow T^x_{e,k} = \{T^{E^{Sk}_t B^{Sk}_t}\} \tag{5}$$

$$Y_{e,k} \leftrightarrow T^y_{e,k} = \{T^{B^{Sk}_t B^{Sk}_{t+1}}\} \tag{6}$$

$$X = (X_g, X_a, \{X_{e,k}\}_{k=1\ldots N}) \tag{7}$$

$$Y = (Y_g, Y_a, \{Y_{e,k}\}_{k=1\ldots N}) \tag{8}$$

In the current paper text, kernels are given as functions with arity three. Clearly that must mean the evaluation of a kernel for the value of the first argument. Instead here I use kernels $K(\mu, \sigma)$ of arity two, where μ is the kernel centre, and σ is the kernel bandwidth. Also a set of kernels is denoted $\mathcal{K}$.

Given this we can define the learning algorithm given all the training trials for a single object for the KDEF algorithm, which is the most complex.

Figure 1: Learning with KDEF

learn-KDEF($X^{0:T}$,$Y^{0:T}$) $\rightarrow \mathcal{K}$
$j = 1$
for each of S trials **do**
  for $t = 0$ to $T - 1$ **do**
    $K_g^{x,j} = K(X_g^t, H\mathbf{h}^{(\mathbf{X})})$
    $K_g^{y,j} = K(Y_g^t, H\mathbf{h}^{(\mathbf{Y})})$
    $K_a^{x,j} = K(X_a^t, H\mathbf{h}^{(\mathbf{X})})$
    $K_a^{y,j} = K(Y_a^t, H\mathbf{h}^{(\mathbf{Y})})$
    for $k = 1$ to N **do**
      $K_{e,k}^{x,j} = K(X_{e,k}^t, H\mathbf{h}^{(\mathbf{X})})$
      $K_{e,k}^{y,j} = K(Y_{e,k}^t, H\mathbf{h}^{(\mathbf{Y})})$
    end for
    $j = j + 1$
  end for
end for
$\mathcal{K} = \{\{K_g^{x,j}, K_g^{y,j}\}, \{K_a^{x,j}, K_a^{y,j}\}, \{K_{e,k}^{x,j}, K_{e,k}^{y,j}\}_{k=1...N}\}_{j=1...M}$

Figure 2: One-step prediction

one-step-prediction-KDEF(X) $\rightarrow Y_g^*$
$F = (g, a, (e, 1) \dots (e, N))$
for $f \in F$ **do**
  for $j = 1$ to M **do**
    $w_{f,j} = K_f^{x,j}(X_f)$
  end for
  $\vec{w_f}$ = normalisation($w_{f,1} \dots w_{f,M}$)
end for
for $i = 1$ to β **do**
  randomly sample a factor $f \in F$
  sample j from distribution $\vec{w_f}$
  $\mathcal{Y}_i = Y_f$ from density $K_f^{y,j}$
end for
maximise Eq.10 with Y_g^* = stochastic-optimisation($\mathcal{Y}$,$\mathcal{K}$)