Inverse Kinematics via Gaussian Mixture Modeling

Aykut C. Satici
July 08, 2021

1 Problem Statement

We want to come up with a general procedure that solves the inverse kinematics problem for redundant manipulators. We will, in particular, work with the planar three-link articulated robot. We are only interested in the position of its end-effector. Therefore, the forward kinematics map goes from the joint angles: $\theta = (\theta_1, \theta_2, \theta_3)$ to $\mathrm{x} = (x, y)$.

2 Solution of the Problem

The solution will be to utilize a Gaussian Mixture Model (GMM) to represent the inverse kinematics map. My presentation will closely follow three sources: Ghahramani (1993); McLachlan and Krishnan (2007); Xu, Chen, Lau, and Ren (2017).

We assume that the data $\Xi = \{\xi_1, \dots, \xi_N\}$ were generated independenty by a mixture density:

$$P(\xi_i) = \sum_{j=1}^{M} \underbrace{P(\omega_j)}_{\pi_j} P(\xi_i \mid \omega_j; \psi_j), \tag{1}$$

where $\{\pi_j\}_j^M$ are the mixture proportions, each component of the mixture is denoted ω_j and parametrized by ψ_j. The mixture proportions are to be picked such that

$$\sum_{j=1}^{M} \pi_j = 1.$$

The full set of parameters of this model are π_j's and ψ_j's. We will write $\Psi = (\pi_j, \psi_j)_j$. When we restrict these probability distributions to be Gaussian, we will further have $\psi_j = (\mu_j, \Sigma_j)$, where μ_j is the mean and Σ_j the covariance of the j^{th} Gaussian. The log of the likelihood of the parameters given the data set is

$$\ell(\Psi \mid \Xi) = \log \Pi_{i=1}^{N} \sum_{j=1}^{M} P(\xi_i \mid \omega_j; \psi_j) P(\omega_j) = \sum_{i=1}^{N} \log \sum_{j=1}^{M} P(\xi_i \mid \omega_j; \psi_j) P(\omega_j). \tag{2}$$

We seek to find the parameter vector Ψ that maximizes $\ell(\Psi \mid \Xi)$. However, this function is not easily maximized numerically because it involves the log of a sum. Intuitively, it is not easily maximized because for each data point, there is a "credit-assignment" problem, i.e., it is not clear which component of the mixture generated the data point and thus which parameters to adjust to fit that data point.

The expectation maximization (EM) algorithm applied to mixtures is an iterative method for overcoming this credit-assignment problem. The intuition behind it is that if one had access to a "hidden" random variable z that indicated which data point was generated by which component, then the maximization problem would decouple into a set of simple maximizations. Mathematically, given $\mathcal{Z} = \{z_1, \ldots, z_N\}$ a "complete-data" log likelihood function could be written,

$$\ell_c(\Psi \mid \Xi, \mathcal{Z}) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \log P(\xi_i \mid z_i; \Psi) P(z_i; \Psi), \tag{3}$$

such that it does not involve a log of a summation.

As proven in Dempster, Laird, and Rubin (1977), $\ell(\Psi \mid \Xi)$ can be maximized by iterating the following two steps,

$$\begin{aligned} &\text{E-step:} \quad && Q(\Psi \mid \Psi_k) = \mathbb{E}\left[\ell_c(\Psi \mid \Xi, \mathcal{Z}) \mid \Xi, \Psi_k\right] \\ &\text{M-step:} \quad && \Psi_{k+1} = \arg\max_{\Psi} Q(\Psi \mid \Psi_k). \end{aligned} \tag{4}$$

The E (Expectation) step computes the expected complete data log likelihood and the M (Maximization) step finds the parameters that maximize this likelihood.

2.1 Mixture of Gaussians

Let me now specialize to the case where the probability functions above are Gaussians. For this model, the E-step simplifies to computing $h_{ij} = \mathbb{E}\left[z_{ij} \mid \xi_i, \psi_k\right]$, the probability that Gaussian j, as defined by the mean $\hat{\mu}_j$ and covariance matrix $\hat{\Sigma}_j$ estimated at time step k, generated data point i:

$$h_{ij}^k = \frac{\left|\hat{\Sigma}_j^k\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\xi_i - \hat{\mu}_j^k\right)^\top (\hat{\Sigma}_j^k)^{-1} \left(\xi_i - \hat{\mu}_j^k\right)\right\}}{\sum_{l=1}^{M} \left|\hat{\Sigma}_j^k\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\xi_i - \hat{\mu}_l^k\right)^\top (\hat{\Sigma}_l^k)^{-1} \left(\xi_i - \hat{\mu}_l^k\right)\right\}}. \tag{5}$$

The M-step then involves re-estimating the means and covariances of the Gaussians along with the mixing proportions using the data set weighted by the h_{ij}:

$$\hat{\mu}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}^k \xi_i}{\sum_{i=1}^{N} h_{ij}^k}, \quad \hat{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}^k \left(\xi_i - \hat{\mu}_j^{k+1}\right)\left(\xi_i - \hat{\mu}_j^{k+1}\right)^\top}{\sum_{i=1}^{N} h_{ij}^k}, \quad \pi_j^{k+1} = \frac{1}{N} \sum_{i=1}^{N} h_{ij}^k. \tag{6}$$

2.2 Supervised Learning

When viewed as supervised learning, each vector ξ_i in the training set is composed of an "input" subvector x_i and a "target" or output subvector θ_i. Applying the learning algorithm, we obtain an estimate of the density of the data in this input/output space. This estimate can be used to approximate a function in the following way:

Given the input vector x_i, we extract all the relevant information from the joint probability density function (pdf) $P(\mathrm{x}, \theta)$ by marginalizing to $P(\theta \mid \mathrm{x})$. For a single Gaussian this conditional density is normal, and by linearity, since $P(\mathrm{x}, \theta)$ is a mixture of Gaussians, so is $P(\theta \mid \mathrm{x})$. In principle, this conditional density is the final output of the density estimator. That is, given a particular input, the network returns the complete conditional density of the output. However, for the purposes of comparison to function approximation methods and since many applications require a single estimate of the output, I will outline two possible ways to obtain such an estimate $\hat{\theta}_i$ of $\theta_i = f(x_i)$.

References

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Ghahramani, Z. (1993). Solving inverse problems using an em approach to density estimation. In *Proceedings of the 1993 connectionist models summer school.*

McLachlan, G. J., & Krishnan, T. (2007). *The em algorithm and extensions* (Vol. 382). John Wiley & Sons.

Xu, W., Chen, J., Lau, H. Y., & Ren, H. (2017). Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators. *The International Journal of Medical Robotics and Computer Assisted Surgery*, *13*(3), e1774.