Advanced Robotics – Lecture Week 9 Learning from Demonstration with Gaussian Mixture Models

This lecture will be very helpful for the 3rd part of the advanced robotics assignment

Text

Description automatically generated

Text

Description automatically generated

An alternative to Time-dependent systems are Time-invariant dynamic systems:

Graphical user interface, text, application

Description automatically generated

The convergence of a system is its ability to reach its steady state.

A disadvantage of this is that the state x is usually multi-dimensional, and this is a challenging machine learning problem. i.e if the state represented a manipulator that can have 67 dimensions of freedom, then the machine learning model would struggle to learn and lead to an unstable system.

A method that guarantees the stability of a system is a control theory method called the Lyapunov theorem, this maps our state to scalar values that belong to real numbers. In other for the function to be a valid Lyapunov function there are certain conditions:

Text

Description automatically generated

The function should be equal to 0 at a steady state where x\* is the point in space that we want our trajectory to reach, the goal of the trajectory.

Text

Description automatically generated

Second the function must be positive for each x in the space whilst also being more than 0 except when it is at the steady state as defined in the previous equation.

Text

Description automatically generated

Lastly, the time derivative must be less than 0 (negative) to make the motion converge at the desired point.

As the time elapses, the system should release energy until it reaches steady state.

**Geometric interpretation of the function**

Candidate Lyapunov function:

Diagram

Description automatically generatedx represent vectors of varying dimensional space that depend on the dimension of the vectors.

The Lyapunov function can be interpreted metrically as a parable. To understand this easily we can look as the Lyapunov function as an energy function where we want for example, out robot to be the centre of the space, our steady state.

The Lyapunov function puts bounds on the complexity of the learned model.

When using the Lyapunov function, every single part of the trajectory should come closer to the steady state otherwise the stability of the system would not be guaranteed.

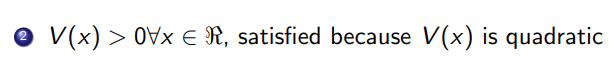
Any function that satisfies the same three properties that the Lyapunov function has can be used instead of solely using the Lyapunov function.

Next we look at how to prove the stability of a system using the three rules required by the Lyapunov theorem.

Text

Description automatically generated

For simplicy if we say that x\* are zero (empty) functions and replace the x in the functions with x\* then we see that the output is 0, thus we have achied the first objective.



For the second objective: is also satisfied by inspecting since the function is quadratic in terms. Even if x is a negative vector, when x is multiplied by itself via the d0t product, the value of this will be always positive. In accordance with how the Lyapunov function is evaluated greated than 0 for each x that belongs in the real numbers.

Text

Description automatically generated

Third objective, for this we use Linear algebra and start with a term and apply the chain rule.

Therefore, the time derivative of the chain rule can be written using the chain rule for derivatives as the derivative of the function with respect to the state \* the derivative of the state with respect to time. We do this because we can calculate the derivate of the Lyapunov function with respect to the state and then the derivate of the state with respect to time is simply the velocity.

So by applyuing the Lyapunov stability theorem using A picture containing text, clock, watch

Description automatically generated and a linear dynamic system A picture containing icon

Description automatically generatedwith steady state at the origin of our spaceA picture containing icon

Description automatically generatedwe can say that the stability of the system can be achieved as long as matrix A is negative definite.

**Some examples of Dynamic systems and how they are represented**

Chart

Description automatically generated

When we have 2 dimensional functions, the representation of this is usually done using string plots. The arrows give the direction of motion. In example (a) we can see that all the lines converge at the origin, (0,0) point when we take random indices of matrix A to satisfy the third requirement of using negative values from a negative definite matrix.

By using some more values of Matrix, A in example (b), we can also get a different kind of motion in our system.

On the other hand, the bottom two examples (c) and (d) show plots of dynamical systems where the 3rd sub constraint is violated. In these cases, convergence does not happen at the origin. Every motion taken in space actually goes away from the steady state, our goal state, this is the opposite of what we want as this means that the systems are not stable and are therefore classified as unstable dynamical systems.

It is not enough to minimize training error given the data, it is also very good practice to guarantee that the learned mapping is table and will converge to the desired state regardless of its initial state as show in example (a) and (b).

The goal still remains to learn the Trajectory from some demonstrations provide by humans

Text, letter

Description automatically generated

In the above example we assume that the demonstrations consist of positions and velocities where the positions are the inputs to the model while the velocities are the targets we want to learn. Next, we want to learn the mapping f which maps the state to its derivative in this case, mapping the positions to the velocities. It is also possible to learn accelerations here. The demonstrations in this case also include a positions, velocity and acceleration as well as the inputs to the mapping “f”.

The optimization problem becomes that we want to find the set of values for matrix A that minimises the absolute error between the predicted velocities in a dataset (derived from a motion model) and velocities demonstrated by a human.

If we don’t have to consider the stability of the system (i.e., because we are using the Lyapunov function for instance) then we only have to minimise the error between demonstrated and predicted motions.

However, because we want convergence of the system to a steady state, we also have to add one constraint:



The matrix (A) must be negative definite.

There are many solvers to this optimisation.

Chart, diagram

Description automatically generated

In the above examples the demonstrations are the red points while the green line is the reproduction of the model (the trajectory that will be followed if we start from the same point as the demonstration. The Learned model is then used to generate this motion.)

For simple trajectories this is easily done in the right-hand site plot. This model is not able to learn more complex trajectories because we have assumed some conditions to simplify the model. The first of which is:

* The predicted velocity is a linear combination of the state

This model is a simply and yet powerful method that would work if we wanted a robot to do a simple motion such as enable a mobile robot to navigate in a desired state in an empty space.

Unfortunately, the model is not capable enough to handle writing letters which has more complex trajectories.

Non-linear, more complex models such as the Gaussian Mixture Model and Neural-Networks are suited for complex tasks.

Graphical user interface, text, application, email

Description automatically generated

**Introduction to Gaussian Distribution & Gaussian Mixture Models**

In order to define a Gaussian Distribution we need a mean and variance.

Chart

Description automatically generated

The assumptions that we make when we say that our data belongs to a Gaussian distribution are that: Chart

Description automatically generated

Variance is used to provide a measurement of how much our data is spread around the mean.

The blue Gaussian is less spread than the red because the variance is much lower at 0.2 than the red at 1 which means the samples that come from red are much more spread out around the mean value.

Chart, histogram

Description automatically generated

If we take the square root of the variance we get the standard deviation which is very useful because it comes in the same units as the random variable, the variable that we model using the gaussian distribution and mean.

How do we then get the values, mean and variance of a gaussian distribution?

We use the:

Text, letter

Description automatically generated

Text, letter

Description automatically generated

Gaussian distributions are multi-dimensional cases of a probability density function. This affects the mean and variance. The mean no longer remains as a scalar value as it is in a one-dimensional case, it becomes a mean vector. We are also left with a covariance matrix which is the opposite of the matrix (A) that was a negative definite matrix. Similarly to the variance, covariance represents how values are spread along dimensions.

Chart, scatter chart

Description automatically generated

In the above examples above the first element of the diagonal represents the variance along the X dimension and the second element of the diagonal represents the diagonal across the Y dimensional. The opposite values are the cross correlation or cross co variance along those.

In this case the values of the cross covariance is close to zero and that is why the spread of the gaussian distribution represented by the iso lines are as such:

* Inner line represents 1 standard deviation
* Outer line represents 3 standard deviation.

This view is from above.

In example (a) there is no correlation along the x, y axis, there is no diagonal correlation of the samples.

Chart, scatter chart, bubble chart

Description automatically generated

That is why we can expect from the covariance matrix to have very small values for the cross covariance (non-diagonal)  and .

On the other hand, in example (b) the cross covariance along the x and y dimension is very large and that is why we expect large value on the non-diagonal compared to the diagonal.

Chart, scatter chart

Description automatically generated

Text, letter

Description automatically generatedNow that the Gaussian distribution has been defined. If we want to fit a gaussian distribution to a set of random samples N we would need a measurement of goodness such as the “likelihood function” which measure how good a specific distribution fits on given data.

The problem then switches to finding the optimal mean and co-variance, the mean and covariance that best describe our data.

Text, letter

Description automatically generated

We apply the log transformation to make our calculation easier because the likelihood function is a product of all the elements evaluated from the Gaussian distribution probability density function.

So by applying the logarithm, the product becomes a “sum” and then we can easily calculate the derivative with respect to the parameters that we are looking for with respect to the unknown parameters and then once we calculate this derivative, we will set it to zero.

Finally we solve the two maximum likelihood estimates for the unknowns, the mean and the covariance.

Examples of Gaussian on the data:

Chart, scatter chart

Description automatically generated

In Example (a) the Gaussian is not properly spread on the data and it does not capture the trend, it is circular rather than oval like in example (b). Therefore, example (a) has a lower likelihood than example (b). Thus example (b) is a better fit to out model. Model (b) is also validated because it has a larger value of likelihood compared to model (a).

Chart

Description automatically generated

It is sub-optimal; it does not capture individual trends complex trajectories. The alternative is to use more Gaussian distributions.

Chart

Description automatically generated

Text, letter

Description automatically generated

W represents the weight of the end normal distribution. If you are using more than one gaussian distribution then you would need to specify the number of weights, one for each gaussian, the number of mean vectors and the number of covariance matrices.

The problem here is that there is no closed solution like the case of a single Gaussian Distribution. Therefore, we use an iterative algorithm called the Expectation Maximization.

Text

Description automatically generated

A picture containing diagram

Description automatically generatedA picture containing calendar

Description automatically generated

Text

Description automatically generated

If we run the EM multiple times and each of those times, the initialisation is completely random then we may end up with different solutions.

CANNOT be stuck in a loop but cannot lead to a global optimum. The solution to this is to run the algorithm multiple times with multiple random initialisations at the beginning and then pick the model that provides the largest likelihood. At the end we pick the model that fits the data the best.

**Summary**

Text

Description automatically generated

USE THE SCIKIT-LEARN PACKAGE, no need to implement it yourself