Reward 机制， 设计reward function

Hi everyone I am xx, I will give a talk about controlling marked temporal point processes using deep reinforcement learning.

Before going to the details of temporal point processes.

I will give a bit of background.

So in practice you will see there a lot of phenomena where discrete events are involved in continuous time.

For example, when people posts tweets in Twitter.

You will find that they are posting messages at some point of time as discrete events, and these event doesn’t appear in discrete time steps. Which means they don’t appear uniformly, they appear in an asynchronous function.

Apart from these online action, we can find other examples of this sort of discrete events in continuous time like medical process that people take tests and diagnoses at random moments in continuous time, also there are examples like Consumer behavior, people buy as discrete events at some point of time.

(Human interact with the real world by taking actions and observing feedback in continuous time. Hence the feedback and the actions are a sequence of events which take place at points in continuous time.)

(The framework of temporal point processes provides a natural representation for this kind of phenomena/ to tackle this kind of shortcomings. )

This events can be modeled using the framework of MTPPs which has shown to have unprecedented predictive power by modeling social processes and human actions.

Next

So what are marked temporal point processes

Marked temporal point process is a specific type of stochastic process that consists of discrete event localizing in continuous time.

So it can be represented as a history of time T and the history consists of events that contain a tuple which consist of the time t at which the event has accurred , and some information associated with that time which is called mark.

For example, This mark can be the like or dislike associated particular message. Or anything that can be well-defined with respect to a particular event on the context.

Ti is defined in continuous time and mark can be discrete or continuous.

Next

So we can use MTPP to model the Human interact with the real world in the context of reinforcement learning.

For example, You can see that, there is a timeline of a particular user in social media, in this timeline , people are making posts, so this blue line can be thought as the posts of an agent. And this agent is trying to maximize some reward

and this agent can perform an action yj at time tj

and the environment provides feedbacks, and the reward is calculated at each episode.

Next

I will give two examples of this kind of process.

Example 1 is the problem of when-to post?

Suppose you are an advertiser and you want to get more attention from the users. Your objective is to try to gain the visibility of your post as maximal as possible from the other people.

So basically what you have in your hand is you can post to it

But the question is the timing of the post to maximize visibility.

At the end of the processes, the average rank will be calculated, which will be defined as the reward.

So this At is the action that defines the times of your post. For example, you post at 9:10

And the environment is other tweeters.

The feedback that you are getting is the rank of you post when others post. For example, other tweeters post at 9:15, at that time, the rank of your post is rank1.

At the end of each time, the reward is being calculated.

Next

Other interesting example of this mark temporal point process is spaced repetition.

Suppose you are going to learning a new language and you go to duolingo an online learning platform to learn new words.

So the online platform will give you some words, you will remember it. And then the platform will test you, they will check whether you quickly forget or not. And then they will give you a new word to learn.

The test will take place sometime after the learning period is over.

For example, duolingo takes action at 9:05 ask question about the word abandon, and the student try to recall this word, after period of study, the student will be asked to take the exam.

So here w want to schedule reviews to maximize recall probability of the learner based on a previous attempts

Next

however The current methods to control such processes, do not generalize well across different environments and reward functions.

Firstly, the methods rely on using stochastic differential equations to model the environment, but often the dynamics are close to a black box.

the feedback process can be complex or unknown in general

Secondly, the other methods require carefully-crafted reward functions to ensure that the underlying stochastic optimal control problem remains tractable, whereas the objective of interest could be arbitrary for different use cases.

Lastly, owing to their parameterization, they cannot take advantage of recent advances in deep learning.

We can overcome these limitations by using deep reinforcement learning.

Next

However, the problem is that the action or feedbacks are asynchronous here and the state of the process is very difficult to determine

Policy determines the time as well as mark of the next action

the classical reinforcement learning problem is framed in terms of discrete time seps, hence it fails to fully capture real word processes we have described.

In this paper, we developed a novel reinforcement learning algorithm to direct the agent actions, while interacting with an asynchronous environment.

We do not make any assumption about the environments behavior and can optimize for arbitrary reward functions.

Next

So I will first introduce the representation of mark temporal point process, how it presents this kind of phenomena.

And then I will describe the reinforcement learning model specifically designed for marked temporal point process, and then I will describe the policy optimization and the evaluation.

Next

The slide shows a typical thing of marked temporal point processes,

here we represent mark temporal point process as marks and times given by events as I have said.

And we model the marks and times separately.

The time is characterized by an intensity function. which is the probability that an event will come in an infinitely small time interval between time t and t plus dt.

The sign start means that the intensity depend on the history.

This lambda start gives you an urgency of facts.

For example, Suppose in the context of viral marketing, if you see that the ranks of your post is going down very swiftly, that means that you should post immediately in order to gain your visibility. That means the lambda should take a high value.

And mark is also characterized by some distribution, and it can be continuous or discrete depending on the situation.

And then, We can explicitly write the joint likelihood of a history of events.

Joint likelihood of the actions is the product of intensity and marks and the survival function.

This exponential term is called the survival function, and it characterize the probability that nothing other than this ti will occur, so it’s an integral over all the time intervals.

So if lambda is a linear function of some known features, the log-likelihood will be a convex function. But here the problem is that we not consider this lambda is the linear thing, because we want to try to model a generic setting, and there will assume no parametric representation of lambda.

Next

And then we will represent our problem in terms of MTPPs.

So here we want to control this action processes, and maximize the final reward.

And we may observe some feedback. Interestingly, we don’t model this lambda start associated in feedback process. It’s some kind of model free reinforcement learning.

That is We don’t have any idea of what’s going on in the feedback, and we can observe this thing at some point of time

The thing we want to solve is to maximize the reward, when the next event should happen and what kind of event should happened.

Next

So in order to tackle the defined reinforcement learning problem, we should first parameterize the policy, which is depending on the state.

In Traditional reinforcement learning problem, we usually consider a discrete setting where the action and state both discrete, or other cases where the actions and feedbacks are continuous signals.

But in this situations, the actions and feedbacks are both synchronous, and they’re characterized by intensity function

So in the contrast of the existing work, our main contribution is that here the policy is an intensity

And the policy is realized by sampling from the intensity.

So what we going to do is

First, design this intensity function which is the policy,

And then we realize this policy by sampling from this intensity function

Next

We can see that

The policy lambda depends on the previous history of the action events and feedback events in an unknown and complex way.

What’ more, the dimension of the state keeps on growing whenever the events coming.

So to alleviate this problem and capture complex influence of the past on the future, we use RNN to model event sequence.

Each event will be input to the RNN model, and then update the hidden state h.

Since the update function is very complex, when we use the intensity function as a non-linear function of the hidden state, we can also model the complex effects of events.

Next

This is the overall representation of this problem.

We have an environment which has both marks and times.

The time temporal differences go to this Wt and Bt, which is a weight and bias.

And we want to embed all the history events including actions yi and feedback zi, so if ei is 0, we calculated yi. Which denotes action. If ei is equal to 1 we calculated zi

And then we calculated the hidden layer by taking inputs of previous events from the input layer.

The output is intensity function and mark distribution.

Here, the b encodes a base intensity level for the occurrence of the (i + 1)-th action event, the term wt(t encodes the influence of the i-th action event, and the term V encodes the influence of previous events.

We use the Exponential function to ensure the intensity is always positive,

( if not you end up getting a finite number of events, that means you have finite sample to train, and you may not get a good quality of estimated parameters)

The distribution of the marks is also depend on the history, it is a categorical distribution here

Theta represent the trainable parameters

Next

Then we can sample action events from the policy.

As show in the algorithm, we can first sample u from a uniform distribution, and calculated the corresponding t according to the inverse function of the cumulative distribution function.

But the problem is the distribution of actions may change because of incoming feedback.

For example, suppose you at any point of time, you decide to sample that you are going to post at after five minutes, but you suddenly discovered that other user is tweeting, which means a new event has arrived,

so if feedback arrive before t, we should modify CDF, and calculated the inverse function again.

Next

In summary, this is a new RL technique to control a mark temporal point process, and it embed state to real vector using RNN, and make efficient sampling procedure to handle asynchronous feedback.

So the next question is how to optimize the policy.

Now, we have the objective function that we try to maximize with respect to both actions and feedbacks, the Rt mean the final reward of each episode.

As we don’t have any assumption of feedback process, and the reward function can be arbitrary complex.

We can use this gradient method, we can update this theta l by just stochastic gradient descent.

We can calculate the gradient of the expected reward function J by applying the traditional reinforce trick.

But the traditional reinforce trick only takes into account of discrete random variables or continuous random variables.

In this case, it is the random process with discrete events in continuous time.

So the author also prove the reinforce trick in Appendix.

And find that the REINFORCE trick is still valid if the expectation is taken over realizations of marked temporal point processes.

It turns out that the gradient of the expected reward function only depends on the action processes.

So with the reinforce trick, we can optimize policy without parametrize of feedback process, and the reward function can be arbitrary.

Next

Finally I come to the evaluation.

First I describe this space repetition.

We have the duolingo data, and the question is how to schedule lesson to maximize the test score.

The result shows that the TPP RL achieves a high accuracy with respect to the existing algorithm, the memorized and this uniform.

Memorize is an algorithm assumes that the underlying function is known and is very simple, so the memorize model get a closed form solution

Uniform is an algorithm choose items uniformly at random.

The TPPRL can get substantially improve recall here

Panel (b) shows the difficulty level of the items selected for review by different methods, it shows that given the limited study time, our method tends to focus on less difficult items

Next

In case of when to post problem, the objective is that we want to gain high visibility from the post. And the average rank is calculated by the end of the campaign. And we try to minimize this rank.

In panel c

The green bar shows the counts of the user’s posts. And the purple shows the posting times of a competing user with higher priority.

As shown in the picture, we are trying to avoid posting after the competing user with higher priority

REDQUEEN is an online algorithm specially designed to minimize the average rank in feeds sorted in reverse chronological order

Karimi is an offline algorithm specially designed to maximize the time at the top in feeds sorted in reverse chronological order

The results show that, by not making any assumption about the feed sorting algorithm, our method is able to outperform both REDQUEEN and Karimi’s method,

Next

So summarizing this is a new reinforcement learning method for temporal point process with asynchronous action and arbitrary reward function.

And the future work is that you can deriving more sophisticated reinforcement learning algorithms to solve this problem.

And you can also develop multiple agent reinforcement learning algorithms for mark temporal point processes.