Intro to Autonomous Driving: 1st Circle

Bayesian Filter I: Bayesian Filter in General, Kalman Filter

Jianan Liu

Gothenburg

May 23, 2018

Disclaimer

All opinions and statements in this presentation are mine and do not in any way represent the company
Any comment or correction of error is welcome, please contact me chisyliu@hotmail.com All slides are reproduced from Python Machine Learning 1st Extra Book Circle

In Memory of Rudolf Kalman

In Memory of Rudolf Kalman

When I reformulate this presentation and make it published online, I notice the date is over 2nd July of 2017. It has been 1 year after passing away of Dr. Rudolf Kalman, the famous scientist, engineer, who created lots of brand new theories in many area. Kalman filter is one of the greatest technologies created in 21st century, we actually see the application of Kalman filter algorithm everywhere nowadays, but the most famous application was using Kalman filter in the project to land on Moon. By having Kalman filter, the classical least square error for linear Gaussian static system which was found by Sir Isaac Newton has been expanded. From a view of signal processing for wireless communication engineer, Kalman filter is equally important as Information theory and wiener filter

Acknowledgement

Acknowledgement

Here I would like to thank for Mengbai Tao, Ziqi Peng, Hui Wen from Baseband Software Division in Ericsson. The inspiration of writing the materials about Bayesian filters and 'relationship between Bayesian filter and Kalman filter' was actually coming from a question Ziqi asked us during one of the lunches all we had together, how many 'commonly used' so called filters there are? Although we are not working together nowadays, I wish Mengbai achieve new success as scrum master in Combitech, Hui Wen and Ziqi Peng good luck in UK and make lots of money in Amazon and Goldman Sachs.

For My Friends

This is for a friend, be happy forever

Poetry

残琴孤鹤暮霭闻 清流石转芳菲韵 仙界三朝白首顿 奈何由命不由君

Bayesian Filter I: Bayesian Filter in General, Kalman Filter

In this presentation

- I am sure this presentation will give you sufficient knowledge to understand Bayesian filter, and Kalman filter. They are very important and useful in the object tracking area and autonomous driving
- This slide will introduce Bayesian filter, Kalman Filter step by step. We will also analyze what are relationships between Bayesian filter, Kalman filter, and RLS/maximize likilihood
- If you want to know more details about Bayesian filter, Kalman filter, etc., I recommend reading Simo Sarkka's book Bayesian Filtering and Smoothing and course slide on Nonlinear Filtering and Estimation, or chapter 3 and 4 of Timothy D. Barfoot's book State Estimation for Robotics
- The context in this circle is NOT included in the book

Bayesian Filter I: Bayesian Filter in General, Kalman Filter

We actually have similar training samples for Bayesian filter and other machine learning algorithm(i.e. RLS for linear regression), but with little difference. We will discuss the details regarding the difference later, but for now we just define two kinds of training samples first

Data set $(\mathbf{y}, \mathbf{X})$ for RLS

Suppose we have L samples, each sample has D features/dimensions(So the input $\mathbf{X}$ is L by D matrix, label $\mathbf{y}$ is L by 1 vector). If we only consider the l^{th} data sample pair, $\mathbf{x}^{(l)}$ is 1 by D vector which represents D features/dimension for l^{th} training data sample, $\mathbf{w}$ is D by 1 vector which represents weight and $y^{(l)}$ represents the result of l^{th} data sample

Bayesian Filter I: Bayesian Filter in General, Kalman Filter

Data series/sequence $(\mathbf{y}_t, \mathbf{x}_t)$ for Bayesian filter

Suppose we have T samples available in time series, each sample has D state variable $\mathbf{x}_t$ and observation $\mathbf{y}_t$ in time t

Note: Here we don't use word "data set" but "data series" to describe the training samples cause the data samples for Bayesian filter is available from time to time and the order of data matters the result

Overview

1. Bayesian Filter in General and Kalman Filter
   - Bayesian Filter in General Format
   - Behind Bayesian Filter: Hidden Markov Model
   - Specific Bayesian Filter: Kalman Filter via Bayesian Inference
2. Revisit Kalmen Filter: Detailed Analysis
   - Kalman Filter via Optimization Problem of MAP
   - Revisit Least Square Estimation: LMS/Gradient Descent
   - Revisit Least Square Estimation: Recursive Least Square for Linear Static State System
   - From Gaussian Static to Gaussian Dynamic State System: RLS vs Kalman
3. Conclusion

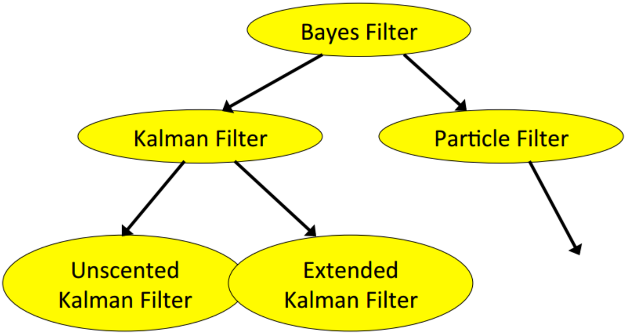
Outline for Section 1

1. Bayesian Filter in General and Kalman Filter
   - Bayesian Filter in General Format
   - Behind Bayesian Filter: Hidden Markov Model
   - Specific Bayesian Filter: Kalman Filter via Bayesian Inference
2. Revisit Kalmen Filter: Detailed Analysis
   - Kalman Filter via Optimization Problem of MAP
   - Revisit Least Square Estimation: LMS/Gradient Descent
   - Revisit Least Square Estimation: Recursive Least Square for Linear Static State System
   - From Gaussian Static to Gaussian Dynamic State System: RLS vs Kalman
3. Conclusion

SLAM: Simultaneous Localization and Mapping

Bayesian Filter in General Format

- Figure from Ashutosh Saxena's course Robot Learning Course in Cornell University shows relationship between all Bayesian filters clearly



Bayesian Filter in General Format

Let's follow the model we introduced in SLAM problem, the Bayesian filter is actually used to solve the problem in SLAM: **state estimation**

- The state estimation is to calculate $P(\mathbf{x}|\mathbf{z})$ according to notations used for modeling SLAM problem, where x stands for state of a system, u is given of observations
- How to calculate this probability?
- Answer is using Bayes rule, we will see how to do it in the following page

Bayesian Filter in General Format

Recall Bayes rule

Bayes Rule

- $P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(A \bigcap B)}{P(B)}$

Bayesian Filter in General Format

Recall definition of independent random variables

Definition of Independent Random Variables

-

Now let's use Bayes rule and assumption of independent measurements on calculating of state estimation, $P(\mathbf{x}|\mathbf{z})$, to derivate the general form of Bayesian filter. Let's see what happens:)

Bayesian Filter in General Format

- For the time stamp t, the estimation of state $\mathbf{x}_t$ could be expressed as $P(\mathbf{x}_t|\mathbf{y}_{1:t})$, where $\mathbf{y}_{1:t}$ stands for the sequence of observation from time stamp 1 to t
- Clearly this is a MAP rule, we try to check the probability of obtaining $\mathbf{x}_t$ based on availability of sequence of observation $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t$
- Now let's see what will be happened on the estimation of state $\mathbf{x}_t$: $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ if applying Bayes rule and assumption of independent measurements
- By applying Bayes rule, $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = P(\mathbf{y}_{1:t}|\mathbf{x}_t)P(\mathbf{x}_t)/P(\mathbf{y}_{1:t})$

Bayesian Filter in General Format

- If we assume all measurements $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t$ are conditional independent on state $\mathbf{x}_t$, and we also have $P(\mathbf{y}_{1:t}|\mathbf{x}_t) \equiv P((\mathbf{y}_{1:t-1}|\mathbf{x}_t) \bigcap (\mathbf{y}_t|\mathbf{x}_t)))$. Then we can get $P(\mathbf{y}_{1:t}|\mathbf{x}_t) \equiv P((\mathbf{y}_{1:t-1}|\mathbf{x}_t) \bigcap (\mathbf{y}_t|\mathbf{x}_t)) = P(\mathbf{y}_{1:t-1}|\mathbf{x}_t)P(\mathbf{y}_t|\mathbf{x}_t)$
- So we will have $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = P(\mathbf{y}_{1:t}|\mathbf{x}_t)P(\mathbf{x}_t)/P(\mathbf{y}_{1:t}) = P(\mathbf{y}_{1:t-1}|\mathbf{x}_t)P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t)/P(\mathbf{y}_{1:t})$
- Switch first and second part then we get $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{y}_{1:t-1}|\mathbf{x}_t)P(\mathbf{x}_t)/P(\mathbf{y}_{1:t})$
- We apply Bayes rule once more on the second and third parts: $P(\mathbf{y}_{1:t-1}|\mathbf{x}_t)P(\mathbf{x}_t) = P(\mathbf{x}_t|\mathbf{y}_{1:t-1})P(\mathbf{y}_{1:t-1})$

Bayesian Filter in General Format

- Plug into the origin, we have $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1})P(\mathbf{y}_{1:t-1})/P(\mathbf{y}_{1:t}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})}P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$
- Until now, we have three parts in the equation which represents the Posterior of state estimation $P(\mathbf{x}_t|\mathbf{y}_{1:t})$:

- 1st part: $\frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})}$, which is a normalization constant cause $\mathbf{y}$ is observation(which is given value)
- 2nd part: $P(\mathbf{y}_t|\mathbf{x}_t)$, which clearly represents a likelihood function (consider a least square regression model $\mathbf{y}_t = \mathbf{x}_t\mathbf{w}_t$ in case $\mathbf{x}_t$ is the variable/weight and $\mathbf{w}_t$ is t^{th} training samples)
- 3rd part: $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$

Bayesian Filter in General Format

What is the third part? Could we rewrite it into other format, e.g. a format of $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$? If it could be written in that format, we could achieve a relationship between $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ in the recursive format for our original expression of state estimation $P(\mathbf{x}_t|\mathbf{y}_{1:t})$. So let's consider if it is possible

- First let's consider a joint probability $P((\mathbf{x}_t|\mathbf{y}_{1:t-1})\bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}))$
- We just rewrite the joint probability in another way: $P((\mathbf{x}_t|\mathbf{y}_{1:t-1})\bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})) \equiv P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
- With setting $A = (\mathbf{x}_t|\mathbf{y}_{1:t-1}), B = (\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$, we have $P((\mathbf{x}_t|\mathbf{y}_{1:t-1})\bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})) \equiv P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1})P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ cause $P(A\bigcap B) = P(A|B)P(B)$

Bayesian Filter in General Format

- Due $P(A) = \int P(A \bigcap B)\, dB = \int P(A|B)P(B)\, dB$, so we get $P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int P((\mathbf{x}_t|\mathbf{y}_{1:t-1}) \bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}))\, d\mathbf{x}_{t-1} = \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1})P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$ which is the **prediction step**
- Here the equation $\int P(A \bigcap B)\, dB = \int P(A|B)P(B)\, dB = P(A)$ is called law of total probability. (It is $\sum_B P(A|B)P(B)$ in discrete case)
- Now we plug the new expression of $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ into third part on the original expression of $P(\mathbf{x}_t|\mathbf{y}_{1:t})$, we have $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})} P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})} \cdot P(\mathbf{y}_t|\mathbf{x}_t) \cdot \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1})P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$ which is the **update step**

Bayesian Filter in General Format

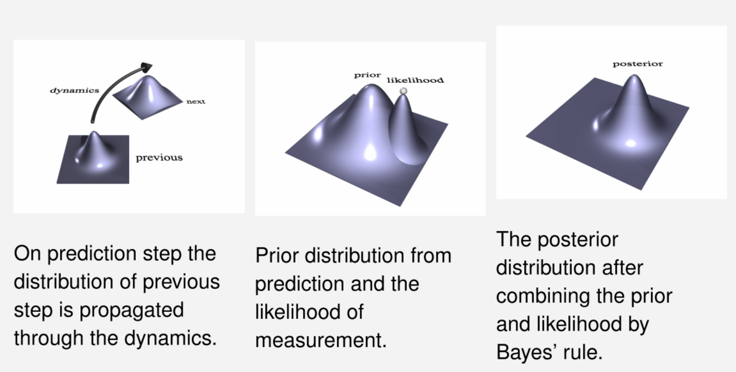
- By assumption of Markov procession of dynamic model $P(\mathbf{x}_t|\mathbf{x}_{t-1})$, which means we have $\mathbf{x}_t$ is only dependent on previous $\mathbf{x}_{t-1}$
- On the other word, $P(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1}) = P(\mathbf{x}_t|\mathbf{x}_{t-1})$
- It means we finally fetch the connection between $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ in a recursive expression

Recursive Expression of Bayesian Filter in General Format

- $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{z}_{1:t})} P(\mathbf{y}_t|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})} \cdot P(\mathbf{y}_t|\mathbf{x}_t) \cdot \int P(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{1:t-1}) P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1} = \underbrace{\frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})} \cdot P(\mathbf{y}_t|\mathbf{x}_t) \cdot \underbrace{\int P(\mathbf{x}_t|\mathbf{x}_{t-1}) P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}}_{prediction}}_{update}$

Graphic Explanation of Distribution Movement from Prediction to Update in Bayesian Filter

Now let's borrow the figure from Simo Sarkkas course Nonlinear Filtering and Estimation in Aalto University, to show how the distribution moves from prediction part to update part in Bayesian filter in general format



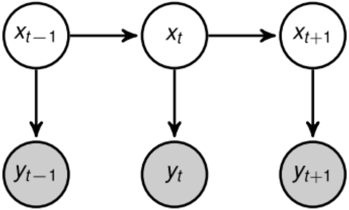
Graphic Explanation of Distribution Movement from Prediction to Update in Bayesian Filter

- Clearly the distribution of previous step $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ has been propagated on prediction step
- In the update step, the distribution of likelihood measurement $P(\mathbf{y}_t|\mathbf{x}_t)$ has been combined with prior distribution from prediction $\int P(\mathbf{x}_t|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$
- According to Bayes rule, we actually get the posterior distribution $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ after combining the prior and likelihood

Behind Bayesian Filter: Hidden Markov Model

The Bayesian filter could be explained by using hidden markov modle(HMM) as well

- Let's see an example of HMM in figure, where the hidden state $\mathbf{x}_t$ are only dependent on previous state $\mathbf{x}_{t-1}$, but the observation $\mathbf{y}_t$ are independent on all other observations and the observation $\mathbf{y}_t$ are conditioned on the hidden state $\mathbf{x}_t$



Behind Bayesian Filter: Hidden Markov Model

The Bayesian filter could be explained by using hidden markov modle(HMM) as well

- Clearly, our needed assumptions for Bayesian filter are same as the characteristics belongs to HMM. Which are:
  1. All observations $\mathbf{y}_t$ are conditional independent on state $\mathbf{x}_t$
  2. The state $\mathbf{x}_t$ is a Markov process
- It means the HMM could describe Bayesian filter for dynamic system very well

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

We already understood how a Bayesian filter works in a recursive expression which established relationship between $P(\mathbf{x}_t|\mathbf{y}_{1:t})$ and $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$. But that is just a generic format without specifying what is the exact relationship(a specific mathematical expression) between hidden state $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, and the exact relationship between hidden state $\mathbf{x}_t$ and observation/measurements $\mathbf{y}_t$. So now the question comes, how does Bayesian filter work for a exact real system (model)?

- From this sub section, we will introduce the concept of Gaussian linear dynamic state system and see how a Bayesian filter in general evolves to Kalman filter in this specific system model

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

First let's check what is Gaussian linear dynamic state system

- A Gaussian linear dynamic state system could be described by two equations:
  1. $\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{q}_{t-1}$
  2. $\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{r}_t$
- For 1st equation, where $\mathbf{x}_t$ stands for the system state(which is a variable we do NOT know, and the task is to estimate the value of $\mathbf{x}_t$), $\mathbf{F}_{t-1}$ stands for the transition of state from t-1 to t, $\mathbf{q}_{t-1}$ is a Gaussian noise with zero mean and covariance $\mathbf{Q}_{t-1}$
- For 2nd equation, where $\mathbf{y}_t$ stands for the observation/measurement, $\mathbf{H}_t$ stands for the measurement loss of $\mathbf{x}_t$, $\mathbf{r}_t$ is a Gaussian noise with zero mean and covariance $\mathbf{R}_t$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- Notice $\mathbf{x}_t$ follows Gaussian distribution by the first equation. And $\mathbf{y}_t$ follows Gaussian as well cause second equation and $\mathbf{x}_t$ is Gaussian

In short, Gaussian linear dynamic state system looks as below

Gaussian Linear Dynamic State System Model

- $\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{q}_{t-1}$
- $\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{r}_t$
- $\mathbf{q}_{t-1} \sim \mathcal{N}(0, \mathbf{Q}_{t-1})$
- $\mathbf{r}_t \sim \mathcal{N}(0, \mathbf{R}_t)$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

From now we start to see how Bayesian filter evolves Kalman filter for Gaussian Linear Dynamic State System Model, and more importantly, how Kalman filter works

- Firstly, we could represent the state $\mathbf{x}_t$ based on observations as Gaussian distribution as below:
- $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t)$ which denotes the probability of state $\mathbf{x}_t$ based on observations from 1 to t
- $P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-)$ which denotes the probability of state $\mathbf{x}_t$ based on observations from 1 to t-1(lack of observation $\mathbf{y}_t$, on the other word, it is the estimation of state $\mathbf{x}_t$ before acquiring observation $\mathbf{y}_t$)

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

Now let's introduce a lemma(it is correct and we could use it)

Lemma 1

- if two variables $\mathbf{a}$ and $\mathbf{b}$ have:
  $P(\mathbf{a}) = \mathcal{N}(\mathbf{a}|\mathbf{m}, \mathbf{P})$ (variable $\mathbf{a}$ follows Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{P}$) and
  $P(\mathbf{b}|\mathbf{a}) = \mathcal{N}(\mathbf{b}|\mathbf{Ha}, \mathbf{R})$ (variable $\mathbf{b}$'s mean has the relationship $\mathbf{Ha}$ with variable $\mathbf{a}$ and covariance is $\mathbf{R}$), actually means $\mathbf{b} = \mathbf{Ha} + n$ where n is a Gaussian variable follows $\mathcal{N}(0, \mathbf{R})$)
- then the joint distribution of $\mathbf{a}$ and $\mathbf{b}$ is:
  $$\mathbf{a} \bigcap \mathbf{b} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{Hm} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{PH}^T \\ \mathbf{HP} & \mathbf{HPH}^T + \mathbf{R} \end{pmatrix} \right)$$
- and marginal distribution of $\mathbf{b}$ is:
  $\mathbf{b} \sim \mathcal{N}(\mathbf{Hm}, \mathbf{HPH}^T + \mathbf{R})$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- From the two conditional probabilities we defined on the page before previous one, we actual could have:
  $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_{t-1}, \mathbf{P}_{t-1})$
  $P(\mathbf{x}_{t}|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_{t}^{-}, \mathbf{P}_{t}^{-})$
- Recall our prediction step for Bayesian filter:
  $P(\mathbf{x}_{t}|\mathbf{y}_{1:t-1}) = \int P((\mathbf{x}_{t}|\mathbf{y}_{1:t-1}) \bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}))\, d\mathbf{x}_{t-1} =$
  $\int P(\mathbf{x}_{t}|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}) P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$
  Here inside the integration we actually have a joint probability of "$\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$" and "$\mathbf{x}_{t}|\mathbf{y}_{1:t-1}$" in prediction step and we actually want to represent the prediction
  $P(\mathbf{x}_{t}|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_{t}^{-}, \mathbf{P}_{t}^{-})$ by using
  $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_{t-1}, \mathbf{P}_{t-1})$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- Notice $\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{q}_{t-1}$, which satisfies the requirement between two Gaussian variables in lemma 1.
- So we could apply lemma 1 on $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$ and $\mathbf{x}_t|\mathbf{y}_{1:t-1}$ by define $\mathbf{a} = \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$ and $\mathbf{b} = \mathbf{x}_t|\mathbf{y}_{1:t-1}$
- On the other word, we have $P(\mathbf{a}) = P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_{t-1}, \mathbf{P}_{t-1})$
  and $P(\mathbf{b}|\mathbf{a}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\mathbf{F}_{t-1}\mathbf{x}_{t-1}, \mathbf{Q}_{t-1})$ cause of $\mathbf{x}_t$ is Markov process and $\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{q}_{t-1}$. It means $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$ and $\mathbf{x}_t|\mathbf{y}_{1:t-1}$ satisfy the "if" requirement in lemma 1

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- So we get the joint probability:
  $P(\mathbf{a} \bigcap \mathbf{b}) = P((\mathbf{x}_t|\mathbf{y}_{1:t-1}) \bigcap P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})) =$
  $P((\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}))=$
  $\mathcal{N}\left(\begin{pmatrix} \mathbf{m}_{t-1} \\ \mathbf{F}_{t-1}\mathbf{m}_{t-1} \end{pmatrix}, \begin{pmatrix} \mathbf{P}_{t-1} & \mathbf{P}_{t-1}\mathbf{F}_{t-1}^T \\ \mathbf{F}_{t-1}\mathbf{P}_{t-1} & \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1} \end{pmatrix}\right)$
- also marginal distribution of $\mathbf{b} = \mathbf{x}_t|\mathbf{y}_{1:t-1}$:
  $P(\mathbf{b}) = P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{F}_{t-1}\mathbf{m}_{t-1}, \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})$
- cause we also defined $P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-)$, then we can use the mean and covariance of $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}$, $\mathbf{m}_{t-1}$ and $\mathbf{P}_{t-1}$, to represent the mean and covariance of $\mathbf{x}_t|\mathbf{y}_{1:t-1}$, $\mathbf{m}_t^-$ and $\mathbf{P}_t^-$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

By getting $P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-) = \mathcal{N}(\mathbf{F}_{t-1}\mathbf{m}_{t-1}, \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})$:

- $\mathbf{m}_t^- = \mathbf{F}_{t-1}\mathbf{m}_{t-1}$
- and $\mathbf{P}_t^- = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}$
- Notice $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ stands for the prediction step of Bayesian filter, and now we are in Gaussian linear dynamic system model, so here we actually get the prediction step in Kalman filter

Prediction Step of Kalman Filter

- $\mathbf{m}_t^- = \mathbf{F}_{t-1}\mathbf{m}_{t-1}$
- $\mathbf{P}_t^- = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- Now we already got $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ in the expression of $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ from prediction step, we move to consider the update step $P(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{P(\mathbf{y}_{1:t-1})}{P(\mathbf{y}_{1:t})}P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$
- Our purpose is to find the relationship between $P(\mathbf{x}_t|\mathbf{z}_{1:t})$ and $P(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ through $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$
- This time define $\mathbf{a} = \mathbf{x}_t|\mathbf{y}_{1:t-1}$ and $\mathbf{b} = \mathbf{y}_t|\mathbf{y}_{1:t-1}$
- We notice $P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = P(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1})P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = P(\mathbf{b}|\mathbf{a})P(\mathbf{a}) = P(\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1}) = P(\mathbf{a}\bigcap\mathbf{b})$ cause $P(\mathbf{y}_t|\mathbf{x}_t$ is independent on $P(\mathbf{y}_{1:t-1})$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- And we also notice $\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{r}_t$
- So we could get $P(\mathbf{b}|\mathbf{a}) = P(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{1:t-1}) = P(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t|\mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t)$
- Previously we defined $P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-)$, so we have $P(\mathbf{a}) = P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{m}_t^-, \mathbf{P}_t^-)$
- By applying lemma 1 on variable $\mathbf{a} = \mathbf{x}_t|\mathbf{y}_{1:t-1}$ and $\mathbf{b} = \mathbf{y}_t|\mathbf{y}_{1:t-1}$, we could get the joint distribution of $\mathbf{a}$ and $\mathbf{b}$ $P((\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1}))$ on next page:

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- The joint probability:
  $P(\mathbf{a} \bigcap \mathbf{b}) = P((\mathbf{x}_t|\mathbf{y}_{1:t-1}) \bigcap P(\mathbf{y}_t|\mathbf{y}_{1:t-1})) =$
  $P((\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1}))=$
  $\mathcal{N}\left(\begin{pmatrix} \mathbf{m}_t^- \\ \mathbf{H}_t\mathbf{m}_t^- \end{pmatrix}, \begin{pmatrix} \mathbf{P}_t^- & \mathbf{P}_t^-\mathbf{H}_t^T \\ \mathbf{H}_t\mathbf{P}_t^- & \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t \end{pmatrix}\right)$
- also marginal distribution of $\mathbf{b} = \mathbf{y}_t|\mathbf{y}_{1:t-1}$:
  $P(\mathbf{b}) = P(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{H}_t\mathbf{m}_t^-, \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

Now let's introduce second lemma

Lemma 2

- if we have two variables $\mathbf{a}$ and $\mathbf{b}$ and the joint distribution of $\mathbf{a}$ and $\mathbf{b}$ is:
$$\mathbf{a} \bigcap \mathbf{b} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \begin{pmatrix} \mathbf{X} & \mathbf{Z}^T \\ \mathbf{Z} & \mathbf{Y} \end{pmatrix} \right)$$
- then the marginal and conditional densities are:
$$P(\mathbf{a}) = \mathcal{N}(\alpha, \mathbf{X})$$
$$P(\mathbf{b}) = \mathcal{N}(\beta, \mathbf{Y})$$
$$P(\mathbf{a}|\mathbf{b}) = \mathcal{N}(\alpha + \mathbf{Z}^T\mathbf{Y}^{-1}(\mathbf{b} - \beta), \mathbf{x} - \mathbf{Z}^T\mathbf{Y}^{-1}\mathbf{Z})$$
$$P(\mathbf{b}|\mathbf{a}) = \mathcal{N}(\beta + \mathbf{Z}\mathbf{X}^{-1}(\mathbf{a} - \alpha), \mathbf{Y} - \mathbf{Z}\mathbf{X}^{-1}\mathbf{Z}^T)$$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- By applying lemma 2 on the joint probability of $\mathbf{a}$ and $\mathbf{b}$ we got on the page before previous one, $P(\mathbf{a} \bigcap \mathbf{b}) = P((\mathbf{x}_t|\mathbf{y}_{1:t-1}) \bigcap P(\mathbf{y}_t|\mathbf{y}_{1:t-1})) = P((\mathbf{x}_t, \mathbf{y}_t|\mathbf{y}_{1:t-1})) = \mathcal{N}\left(\begin{pmatrix} \mathbf{m}_t^- \\ \mathbf{H}_t\mathbf{m}_t^- \end{pmatrix}, \begin{pmatrix} \mathbf{P}_t^- & \mathbf{P}_t^-\mathbf{H}_t^T \\ \mathbf{H}_t\mathbf{P}_t^- & \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t \end{pmatrix}\right)$
- We notice $P(\mathbf{a}|\mathbf{b}) = P(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{y}_t) = P(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the final variable we want to calculate from update step, and we already defined $P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t)$
- We could get the conditional density function $P(\mathbf{a}|\mathbf{b}) = P(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{y}_t) = P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{m}_t, \mathbf{P}_t) = \mathcal{N}(\mathbf{m}_t^- + \mathbf{P}_t^-\mathbf{H}_t^T(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{m}_t^-), \mathbf{P}_t^- - \mathbf{P}_t^-\mathbf{H}_t^T(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)^{-1}\mathbf{H}_t\mathbf{P}_t^-)$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

- So it means:
  $\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1} (\mathbf{y}_t - \mathbf{H}_t \mathbf{m}_t^-)$
  and $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1} \mathbf{H}_t \mathbf{P}_t^-$
- We use new variables to make the equations simpler:
  Let $\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t$
  $\mathbf{v}_t = \mathbf{y}_t - \mathbf{H}_t \mathbf{m}_t^-$
  So we have $\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1} = \mathbf{P}_t^- \mathbf{H}_t^T \mathbf{S}_t^{-1}$
- Then we rewrite the two equations, we get:
  $\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{K}_t \mathbf{v}_t$
  $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^T$

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

So we actually rewrite what we got together, it is from update step of Bayesian filter with applying for Gaussian linear dynamic system model. So we actually get update step of Kalman filter

Measurements Update Step of Kalman Filter

- $\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t$
- $\mathbf{v}_t = \mathbf{y}_t - \mathbf{H}_t \mathbf{m}_t^-$
- $\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}_t)^{-1} = \mathbf{P}_t^- \mathbf{H}_t^T \mathbf{S}_t^{-1}$
- $\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{K}_t \mathbf{v}_t$
- $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^T$

where $\mathbf{K}_t$ is Kalman gain

Kalman Filter: A Specific Bayesian Filter for Dynamic State System in Gaussian Distribution

Combine prediction and update, we get Kalman filter($\mathbf{m}_t = \mathbf{x}_t$ for random variable with Gaussian distribution)

Predication Step of Kalman Filter

- $\mathbf{m}_t^- = \mathbf{F}_{t-1}\mathbf{m}_{t-1}$
- $\mathbf{P}_t^- = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}$

Measurements Update Step of Kalman Filter

- $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t$
- $\mathbf{v}_t = \mathbf{y}_t - \mathbf{H}_t\mathbf{m}_t^-$
- $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^T(\mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t)^{-1} = \mathbf{P}_t^-\mathbf{H}_t^T\mathbf{S}_t^{-1}$
- $\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{K}_t\mathbf{v}_t$
- $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T$

Kalman Filter Algorithm(Start from t = 1)

for t in T **do**
  // Prediction step of Kalman Filter
  $\mathbf{x}_t^- = \mathbf{F}_{t-1}\mathbf{x}_{t-1}$
  $\mathbf{P}_t^- = \mathbf{F}_{t-1}\mathbf{P}_{t-1}\mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1}$
  // Measurements Update Step of Kalman Filter
  $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{R}_t$
  $\mathbf{v}_t = \mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t^-$
  $\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}_t^T\mathbf{S}_t^{-1}$
  $\mathbf{x}_t = \mathbf{x}_t^- + \mathbf{K}_t\mathbf{v}_t$
  $\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T$
  t = t + 1
end for

Outline for Section 2

1. Bayesian Filter in General and Kalman Filter
   - Bayesian Filter in General Format
   - Behind Bayesian Filter: Hidden Markov Model
   - Specific Bayesian Filter: Kalman Filter via Bayesian Inference
2. Revisit Kalmen Filter: Detailed Analysis
   - Kalman Filter via Optimization Problem of MAP
   - Revisit Least Square Estimation: LMS/Gradient Descent
   - Revisit Least Square Estimation: Recursive Least Square for Linear Static State System
   - From Gaussian Static to Gaussian Dynamic State System: RLS vs Kalman
3. Conclusion

Kalman Filter: As Optimization Problem of Maximize A Posterior(MAP)

- Previously in the first section "Bayesian Filter in General and Kalman Filter", we derivate Kalman filter analytic solution from Bayesian filter in conceptual/general by using two lemmas regarding the Gaussian distribution in following of Bayesian inference. In this sub section, we treat Kalman filter from the angel of a optimization problem which maximizes posterior in Gaussian
- But we only show the ideas and procedures rather than giving the exact detail of derivation, cause the purpose of showing this is just giving the other way of understanding Kalman filter. Refer to the chapter 3 of Timothy D. Barfoot's book State Estimation for Robotics if you are interested in detail how to derivate Kalman filter via MAP

Kalman Filter: As Optimization Problem of Maximize A Posterior(MAP)

From this page, we show the basic idea of how to get Kalman filter via MAP optimization problem

- First we actually want to do is to find out the estimation of $\mathbf{x}$ which could maximize the posterior $P(\mathbf{x}|\mathbf{y})$
- In the other words, $\widehat{\mathbf{x}} = argmax_{\mathbf{x}} P(\mathbf{x}|\mathbf{y})$
- Due to Bayes rule we have $P(\mathbf{x}|\mathbf{y}) = P(\mathbf{y}|\mathbf{x})P(\mathbf{x})/P(\mathbf{y})$
- So we actually want to $\widehat{\mathbf{x}} = argmax_{\mathbf{x}} P(\mathbf{y}|\mathbf{x})P(\mathbf{x})/P(\mathbf{y}) = argmax_{\mathbf{x}} P(\mathbf{y}|\mathbf{x})P(\mathbf{x})$

Kalman Filter: As Optimization Problem of Maximize A Posterior(MAP)

- Due to HMM, we know
  $P(\mathbf{y}|\mathbf{x}) = \Pi_{t=1}^{T} P(\mathbf{y}_t|\mathbf{x}_t)$
  and $P(\mathbf{x}) = P(\mathbf{x}_0|\widehat{\mathbf{x}}_0)\Pi_{t=1}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1})$
- $P(\mathbf{x}_0|\widehat{\mathbf{x}}_0)$, $P(\mathbf{x}_t|\mathbf{x}_{t-1})$, $P(\mathbf{y}_t|\mathbf{x}_t)$ are all Gaussian distribution. We could use the definition of Gaussian distribution density function to express them mathematically by using $\mathbf{F}_{t-1}$, $\mathbf{P}_t$, $\mathbf{Q}_t$, $\mathbf{R}_t$, etc.
- It means we could get the mathematical expression of $P(\mathbf{x}_0|\widehat{\mathbf{x}}_0)$, $P(\mathbf{x}_t|\mathbf{x}_{t-1})$, $P(\mathbf{y}_t|\mathbf{x}_t)$ exactly

Kalman Filter: As Optimization Problem of Maximize A Posterior(MAP)

- Then we go back to the original optimization problem, we take the ln on the objective function, so we have $\widehat{\mathbf{x}} = \mathit{argmax}_{\mathbf{x}}\ \mathit{ln}(P(\mathbf{y}|\mathbf{x})P(\mathbf{x})) = \mathit{argmax}_{\mathbf{x}}\ \mathit{ln}(\Pi_{t=1}^{T} P(\mathbf{y}_t|\mathbf{x}_t)) + \mathit{ln}(P(\mathbf{x}_0|\widehat{\mathbf{x}}_0)) + \mathit{ln}(\Pi_{t=1}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}))$
- Then we take the inverse of maximize of posterior, so we get $\widehat{\mathbf{x}} = \mathit{argmin}_{\mathbf{x}}\ -\mathit{ln}(\Pi_{t=1}^{T} P(\mathbf{y}_t|\mathbf{x}_t)) - \mathit{ln}(P(\mathbf{x}_0|\widehat{\mathbf{x}}_0)) - \mathit{ln}(\Pi_{t=1}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}))$
- This is a convex function, we set the derivative of objective which equal to 0 and calculate the analytical result of $\mathbf{x}$
- Then by manipulating a bit of result, we will get the same format as Kalman filter from Bayesian inference by using 2 Gaussian lemma

Kalman Filter: As Optimization Problem of Maximize A Posterior(MAP)

- Again, we do NOT derivate detail of this procedure via MAP step by step, but all detail could be found on chapter 3 of Timothy D. Barfoot's book State Estimation for Robotic
- So far, we know Kalman filter could be derivated either from Bayesian inference by using two Gaussian lemma, or from optimization problem maximize posterior solved analytically by using "inverse of ln" and definition of Gaussian distribution density function

Revisit Least Square Estimation: LMS/Gradient Descent for Machine Learning Problem

Firstly let us recall linear regression model $\mathbf{y} = \mathbf{X}\mathbf{w}$ which we mentioned in section 3 of 1st presentation Python Machine Learning book circle, there we introduced LMS

- LMS is a batch gradient descent based searching algorithm which searches in the direction of inversion of gradient iteratively.
- LMS uses 'average' of correlation of observed data samples to replace of 'expectation' of correlation of ensemble data set, which leads to a LS/ML sense approximation of MMSE analytical solution
- LMS needs to take all observed data samples into consideration for each iteration, which makes LMS NOT suitable for online/sequential learning/estimation

Revisit Least Square Estimation: LMS/Gradient Descent for Machine Learning Problem

- More importantly, LMS/Gradient Descent searching algorithm actually works for static system where the variable/weight $\mathbf{w}$ is static(NOT changing depends on time stamp t). If the variable/weight $\mathbf{w}$ is changing from time to time(e.g. $\mathbf{w}_t \neq \mathbf{w}_{t-1}$, it is dynamic state system), the $\mathbf{w}$ is a time-series where the order of data is important(you can NOT change the order of data samples for different time cause the trend of data carries information). LMS/Gradient Descent searching algorithm does NOT work for dynamic state system(It is why LMS/Gradient Descent searching algorithm, as algorithm for LS/ML criteria only works least square regression kind problems because the linear regression model is a static model where variable $\mathbf{w}$ NOT related to time)

Revisit Least Square Estimation: LMS/Gradient Descent for Machine Learning Problem

LMS/Gradient Descent does NOT fit for state estimation

- So LMS/Gradient Descent usually only works for machine learning problem, statistical signal processing and wireless communication, e.g.(which are static system and the variable does NOT change from time to time) rather than tracking, time-series analysis, e.g.(which are dynamic system and the variable varies on time) because the order of training data does NOT matter(training data does NOT depend on time) and the model in machine learning is usually NOT dynamic model(the variables/weights does NOT vary in different t/NOT related to t). All in all, LMS/Gradient Descent does NOT fit for (dynamic system)state estimation problem

Revisit Least Square Estimation: Recursive Least Square for Online/Sequential Estimation

Now recall section 4 of 1st presentation in Python Machine Learning book circle, an alternative solution RLS was introduced for the same linear model $\mathbf{y} = \mathbf{X}\mathbf{w}$

- RLS is NOT gradient descent based searching algorithm
- Instead of searching in the direction of inversion of gradient iteratively, **RLS actually calculate the $\mathbf{w}_{LS}(l)$ of current l samples for l^{th} recursive step**
- Calculation of $\mathbf{w}_{LS}(l)$ for each recursive step is not directly from analytic solution but using matrix inversion lemma to convert the analytic solution in another format to avoid computing of matrix inversion
- RLS has faster converge speed and suitable for online learning/sequential estimation

Revisit Least Square Estimation: Recursive Least Square for Online/Sequential Estimation

We know the recursive least square algorithm is

RLS algorithm

- $\mathbf{P}_{l+1} = \mathbf{P}_l - \frac{\mathbf{P}_l \mathbf{x}^{(l+1)} {\mathbf{x}^{(l+1)}}^T \mathbf{P}_l}{1 + {\mathbf{x}^{(l+1)}}^T \mathbf{P}_l \mathbf{x}^{(l+1)}}$
- $\widehat{\mathbf{r}}(l+1) = \widehat{\mathbf{r}}(l) + {\mathbf{x}^{(l+1)}}^T y^{(l+1)}$
- $\mathbf{w}(l+1) = \mathbf{P}_{l+1} \widehat{\mathbf{r}}(l+1)$

Revisit Least Square Estimation: Recursive Least Square for Online/Sequential Estimation

RLS algorithm in recursive form of $\mathbf{w}(l)$ is as below

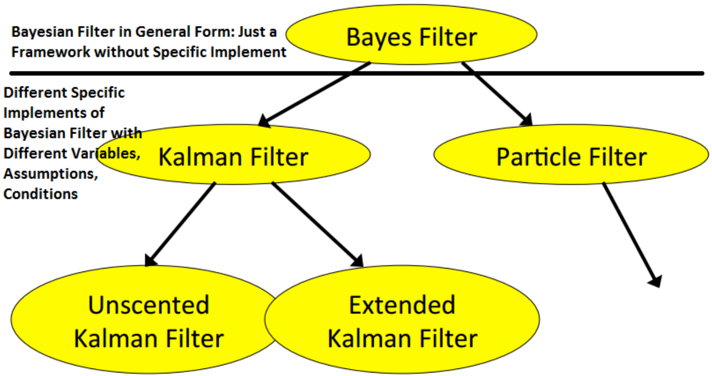
RLS algorithm in recursive form of $\mathbf{w}(l)$

- $\mathbf{k}_{l+1} = \frac{\mathbf{P}_l {\mathbf{x}^{(l+1)}}^T}{1+\mathbf{x}^{(l+1)}\mathbf{P}_l {\mathbf{x}^{(l+1)}}^T}$
- $\mathbf{w}(l+1) = \mathbf{w}(l) + \mathbf{k}_{l+1}(y^{(l+1)} - x^{(l+1)}\mathbf{w}(l))$
- $\mathbf{P}_{l+1} = \mathbf{P}_l - \mathbf{k}_{l+1}\mathbf{x}^{(l+1)}\mathbf{P}_l$

For details of derivation of RLS in recursive form of $\mathbf{w}(l)$, please either refer to Lecture 10, Adaptive Signal Processing Course of Tampere University of Technology by Ioan Tabus, wiki Recursive Least Square Filter, or section 4 of Python Machine Learning: The 1st Book Circle

From Gaussian Static to Gaussian Dynamic State System: RLS vs Kalman

We still use figure from Ashutosh Saxena's course Robot Learning Course in Cornell University.



RLS is Part of Kalman Filter

Let us have a quick look of **Kalman Filter**(Here I used the same equations from table 1.1 summary of kalman filter from page 10 in Haykin's book 'Kalman Filtering and Neural Network') without explaining too much regarding what they are, but we come back to analyze details later

- Firstly we define the state space model

State Space Model

- $\mathbf{x}_k = \mathbf{F}_{k,k-1}\mathbf{x}_{k-1} + \mathbf{n}_{k-1}$
- $\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$

- Secondly we have propagation/predication and measurement update equations

From Static to Dynamic State System: RLS to Kalman

Propagation/Predication

- State estimation propagation/predication
  $\widehat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}\widehat{\mathbf{x}}_{k-1}^-$
- Error covariance propagation
  $\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1}$

Measurement Update

- Kalman gain matrix
  $\mathbf{G}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
- State estimation update
  $\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_k^-)$
- Error covariance update
  $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_k^-$

From Recursive Least Square Algorithm to Kalman

Now let us compare 'measurement update' with RLS algorithm

RLS algorithm in recursive form of $\mathbf{w}(l)$

- $\mathbf{k}_{l+1} = \frac{\mathbf{P}_l \mathbf{x}^{(l+1)^T}}{1+\mathbf{x}^{(l+1)}\mathbf{P}_l \mathbf{x}^{(l+1)^T}}$
- $\mathbf{w}(l+1) = \mathbf{w}(l) + \mathbf{k}_{l+1}(y^{(l+1)} - x^{(l+1)}\mathbf{w}(l))$
- $\mathbf{P}_{l+1} = \mathbf{P}_l - \mathbf{k}_{l+1}\mathbf{x}^{(l+1)}\mathbf{P}_l$

Measurement Update

- Kalman gain matrix
  $\mathbf{G}_k = \mathbf{P}_k^- \mathbf{H}_k^T(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
- State estimation update
  $\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^- + \mathbf{G}_k(\mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_k^-)$
- Error covariance update
  $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{G}_k \mathbf{H}_k \mathbf{P}_k^-$

From Recursive Least Square Algorithm to Kalman

They are same. The measurement update process of Kalman filter is indeed a RLS adaptive filter. It means RLS is Part of Kalman Filter. Why?

- Now revisit second equation of state space model of Kalman filter

Measument Equation of State Space Model

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

- The second equation there, the measurement model is actual happened in a linear Gaussian static system(in the measurement, x does NOT change) which is actual a linear regression model in our machine learning problem.

From Recursive Least Square Algorithm to Kalman

- On the other hand, if we look at from the Bayesian filter point of view, the "update step/part" contains the unknown part "likelihood measurement $P(\mathbf{z}_t|\mathbf{x}_t)$" to be decided(as we already got prior distribution $P(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int P(\mathbf{x}_t|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\, d\mathbf{x}_{t-1}$ the from "prediction step"). As we discussed in previous pages(and recall we learned in the machine learning 1st book circle), which criteria the linear static Gaussian system(which is a linear regression model)? Yes, maximize likelihood! And what could be used to maximize likelihood when the observation/measurements are NOT all available once(the training data samples are given in a sequence, from time to time)? Yes, RLS!

That is why We actually use RLS estimation in the Kalman filter

Conclusion of Kalman Filter

Kalman filter works for estimation of random state $\mathbf{x}$ in linear dynamic system which follows Gaussian distribution

Conclusion of Kalmen Filter

- Kalman filter could be seen as an extension of "RLS algorithm works for estimation problem of a static linear Gaussian system with maximize likelihood criteria", it uses maximize posterior criteria for estimation problem of a dynamic linear Gaussian system recursively
-

Outline for Section 3

1. Bayesian Filter in General and Kalman Filter
   - Bayesian Filter in General Format
   - Behind Bayesian Filter: Hidden Markov Model
   - Specific Bayesian Filter: Kalman Filter via Bayesian Inference
2. Revisit Kalmen Filter: Detailed Analysis
   - Kalman Filter via Optimization Problem of MAP
   - Revisit Least Square Estimation: LMS/Gradient Descent
   - Revisit Least Square Estimation: Recursive Least Square for Linear Static State System
   - From Gaussian Static to Gaussian Dynamic State System: RLS vs Kalman
3. Conclusion

Conclusion

-
-
-
-

Conclusion

-

References

- Simo Sarkka, Nonlinear Filtering and Estimation Course, Aalto University
- Zhe Chen, Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond
- Timothy D. Barfoot, State Estimation for Robotics
- Michal Reinštein, From Bayes to Extended Kalman Filter
- Greg Welch, Gary Bishop, An Introduction to the Kalman Filter
- Simon Haykin, Kalman Filtering and Neural Network
- Simo Sarkka, Kalman Bayesian Filtering and Smoothing

References

- Dan Simon, Crash Course on Kalman Filtering, Cleveland State University
- Wikipedia: Simultaneous Localization and Mapping (SLAM) Link
- Ashutosh Saxena, Robot Learning Course, Cornell University
- Cyrill Stachniss, Robot Mapping and SLAM Course, University of Freiburg/University of Bonn
- Gustaf Hendeby, Sensor Fusion Course, Linköping University
- Ioan Tabus, Adaptive Signal Processing Course, Tampere University of Technology
- John Paisley, Machine Learning Course, Columbia University/Edx

Question?