```
import numpy as np

a= np.transpose(0.5)
```

Section 1 - Introduction and Primer in statistics: September 15, 2021

- Sensor fusion and nonlinear filtering is a core technology for both advanced driver assistance systems, as well as self-driving vehicles.
- Both of these rely on detailed knowledge about the surrounding traffic situation, for example, the position and heading of other vehicles.
- Typically, this is achieved by filtering and fusing the noisy observations of the traffic situation coming from a multitude of sensors, such as radars, cameras,

- **In our course, will focus on the filtering and sensor fusion** algorithms that are at the core of any tracking system but can also be used on their own to solve advanced problems related to,
- for example,
  - self-localization

- learn the fundamental theory, which these algorithms are based upon,
  - Get to implement your methods yourself
  - Build your own sensor fusion toolbox

1.1 sensor fusion and nonlinear filtering?

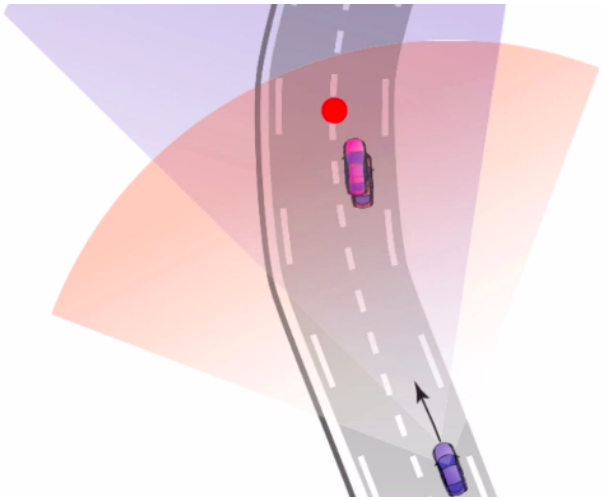
- Use a sequence--now usually we mean a time sequence here-- of noisy observations from one or more sensors.
- If we use one sensor, we call it filtering.

- Now, the goal is to filter the sequence of noisy sensor observations to get a better estimate of some unknown quantity of interest, which we in this course will refer to as our state.
- However, we are not just content with an estimate. We are also interested in finding the associated uncertainty measures, that is, how sure we are of our estimate.

Problem

About Camera

- Now, **the camera is capable of detecting and classifying vehicles in its image due to the nature of the camera**, it's fairly accurate at determining the relative angle to the vehicle as there is a simple mapping between pixel and angle.
- But much **worse at determining the distance to the object, as this cannot be measured directly using a single camera**.
- Now, let's say that the camera detects that there is vehicle here.



About Radar

- The **radar, is very good at measuring the distance to an object, as this can be measured directly from the time delay of the returning radar wave**.
- the radar can also observe the Doppler shift of the returning wave, which gives the relative radial velocity of the vehicle in this case.
- That is, the relative speed of the vehicle in the direction towards the radar.

Radar limitation

- However, the **radar is not good at measuring the angle to an object**.
- This is mainly due to limitations in the size of the radar antenna and the fact that the vehicle in itself is not a well-defined point target and we tend to get reflections from many different places on the vehicle.

Summary

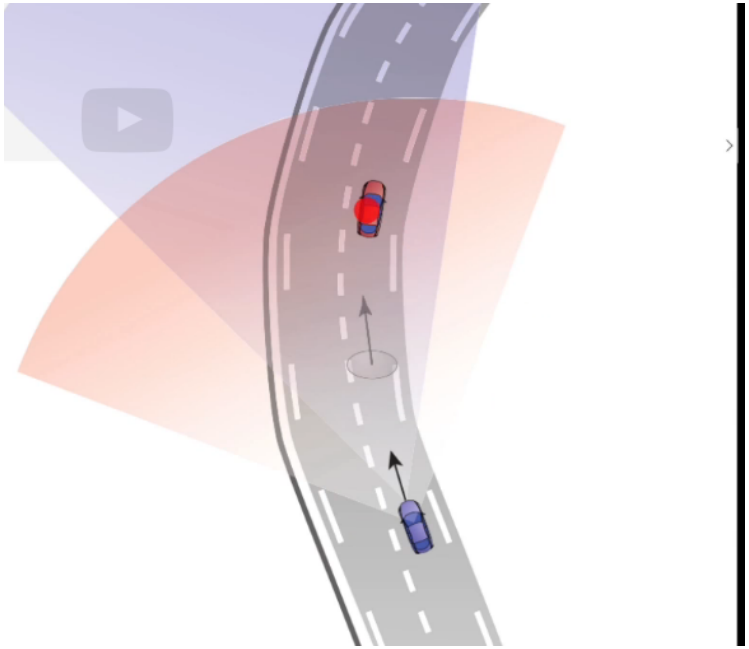
- So the radar is good at measuring distance and relative radial speed of the object but it's worse at measuring the angle to an object.
- The camera, on the other hand, is good at determining the relative angle to the object and to classify what type of object it is. It is, however, not as good as measuring the distance to the object.

Sensor Models

- With the aid of the sensor models, we will make sure that we use the strength of both sensors to get a better idea of the state of this vehicle, that is, where it is and where it's heading.

Problems

- And up until now, we only considered a single time instance.
- So what happens if we move to the next time instance where we get new information from our sensors?



- If we assume that the vehicle is continuing to move with the same velocity as it had in the previous time instance, this would mean that the vehicle should be somewhere closer at the current time.
- Now, as we are making assumptions about the motion of the vehicle and predicting into the future, **it would be reasonable that we should increase our uncertainty a bit to account for this added uncertainty**.

Motion model

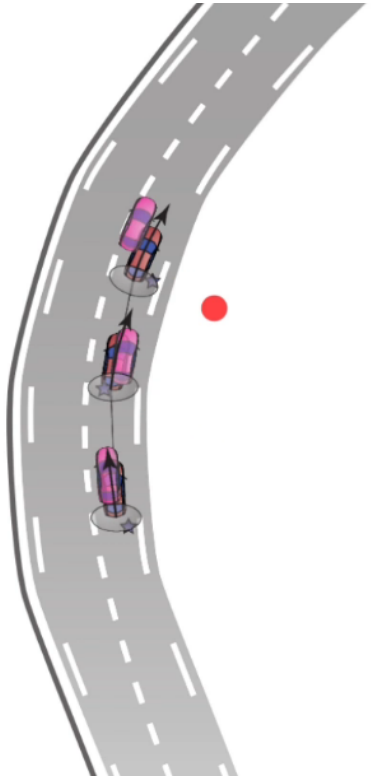
- In our filters, we call this our **prediction step** and we call the model that we use to calculate our prediction as our **motion model**.

Note

- We're adding new information about the position and velocity of the vehicle from our current observations, our uncertainty will shrink again, which is natural.
- So if we move to the next time instance, our vehicles has moved, so we predict our previous information to the current time where we get new information from our sensors, which we then use to update our knowledge about the current state of the vehicle.

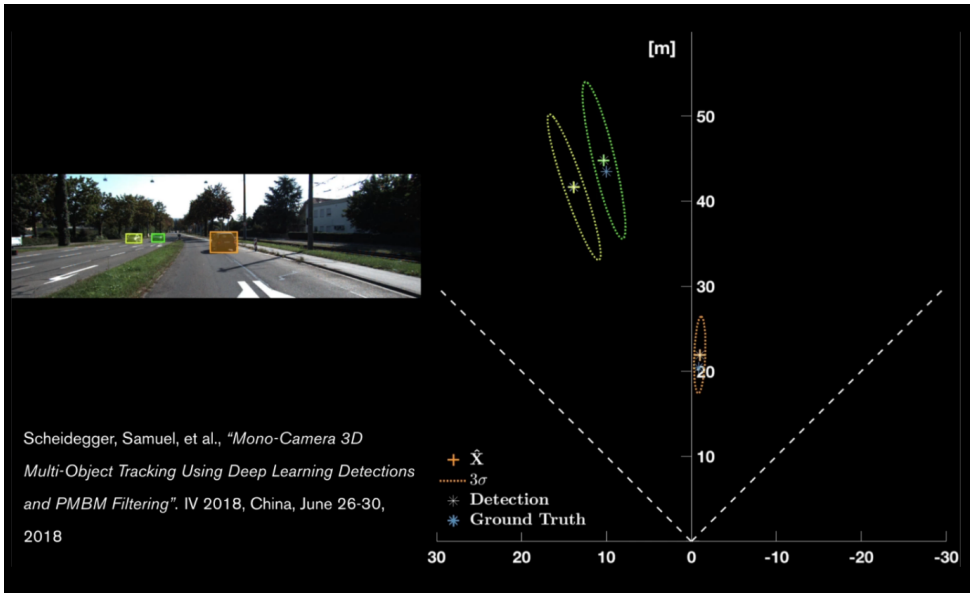
Example

- We have considered a sequence of three noisy sensor observations,
  - both from a radar and a camera,
  - which we have fused in order to get a better understanding of where this vehicle is, as well as its velocity vector in each time instance.



1.2 Demonstrations

Now let's look at three applications where we use the type of methods that you will learn in this course and apply them to real sensor observations.



- To the right is a bird's eye view of the situation, centered at the camera.
- The detection from the camera is shown as white stars. And the result of our filter is shown as these colored ellipses with the same color as the bounding boxes in the image. And the plus in the middle here indicates our estimated position.
- And the star is the actual positions of the vehicles. So ideally, we want our estimate, the plus markers, to be as close as possible to the true position of the vehicles, the stars, which would mean that our guess of where the vehicle is, is accurate.

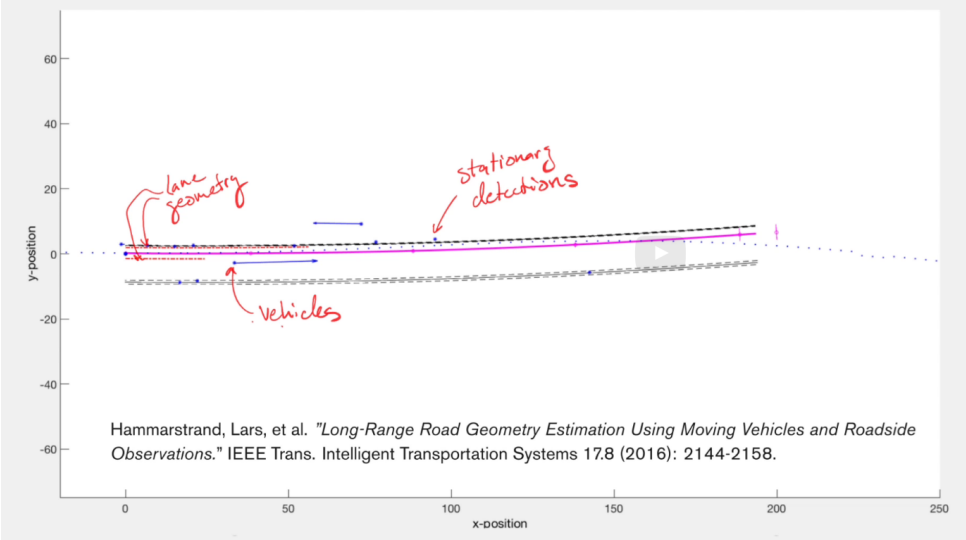
- The aim is to estimate the 3D position and velocity of vehicles from a sequence of images.
- The observations that are used come from a convolutional neural network which is trained to detect vehicles and to report a 2D bounding box of the vehicles in the image, as illustrated by these colored boxes, as well as the distance to the object.
- As we discussed previously, using a single camera, we cannot directly measure the distance to the vehicle. The distance reported by the network is therefore, instead, based on learned scale and appearance cues, and can sometimes be a bit noisy.
- However, by filtering these observations in a so-called **unscented Kalman filter**, we can get better and less noisy estimates of where the vehicles are.

Results

- What we see here is that our filter manages to filter the noisy detections from the camera and over time gets more and more certain about the position of the vehicles.

▼ **Problem :** How can use sensors to estimate the geometry of the road?

- This work was published in this paper from 2016, where we used observations from a radar and a camera to try to estimate the geometry or shape of the road up to 200 meters ahead of our host vehicle in highway scenarios.
- Now from the camera, we get information about the shape of the lane markings, of the current lane. These are shown here as red dashed lines.
- Typically, the camera is able to detect the lane markings up to roughly 50 to 60 meters, but sometimes shorter.



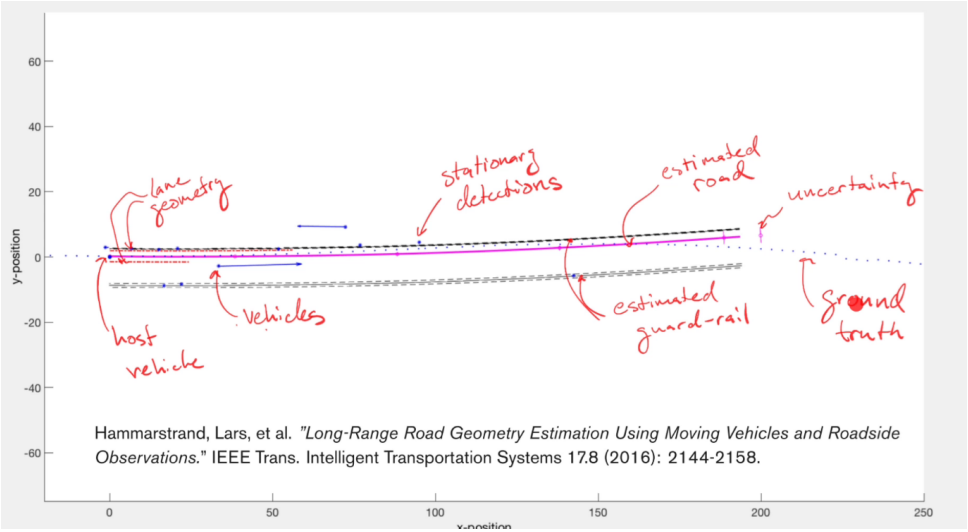
- **From the radar**, we mainly get two things.
  - First, we get the **relative position and velocity of other vehicles**.
  - The second thing that we get is **stationary detections** from the roadside objects, such as guardrails or barriers,

Well, the lane markings coming from the camera is directly related to the shape of the road, right?

- But in best case, it's only valid up to roughly 50, 60 meters, which is not enough.
- The radar, on the other hand, can see much further and typically reports objects up to 200 meters or more.
- However, the **radar cannot measure the geometry of the road directly**.

If we are able to detect a guardrail, this guardrail should be approximately parallel to the road as well.

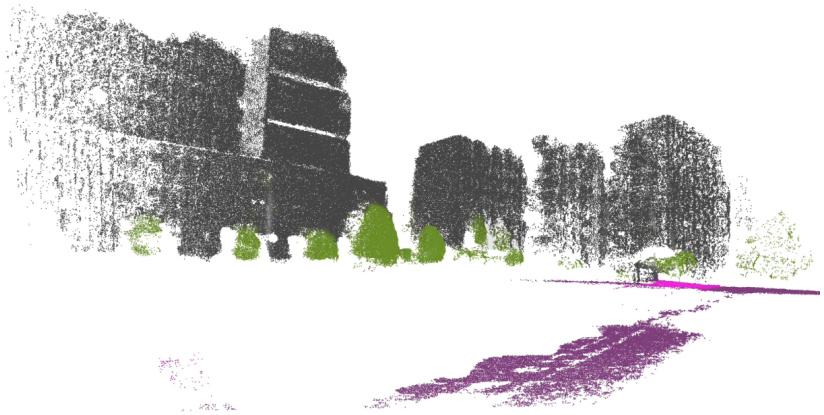
- Now, if we assume that we can describe the shape of the road using a so-called **clothoid spline**, so a mathematical model of the road geometry, we can then estimate the parameters of this spline using our sensor models and our observations in an unscented Kalman filter.



Hammarstrand, Lars, et al. *"Long-Range Road Geometry Estimation Using Moving Vehicles and Roadside Observations."* IEEE Trans. Intelligent Transportation Systems 17.8 (2016): 2144-2158.

Problem : Self-localization

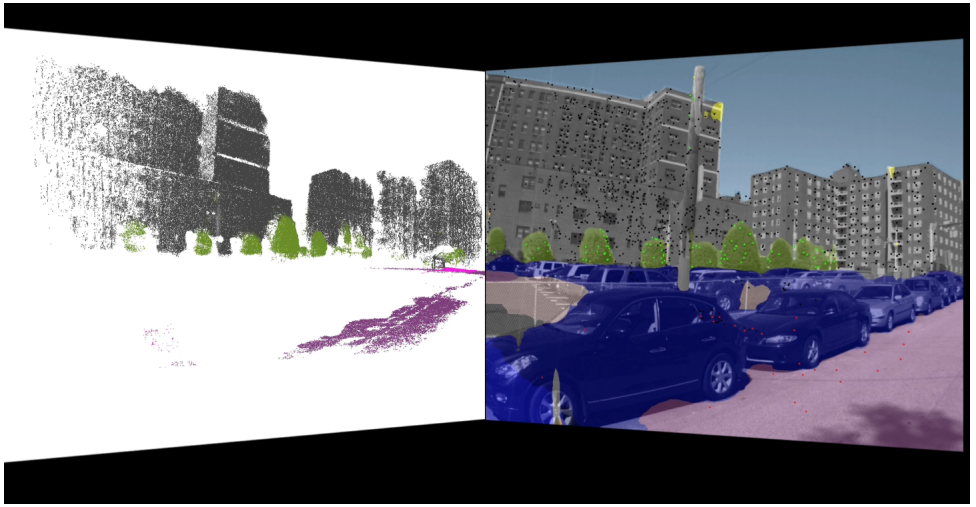
- Let's say that we have an autonomous vehicle which should navigate on its own in this type of city environment.
- One possible such map could be a semantic 3D point cloud map.



Stenborg, Erik, et. al., "Long-Term Visual Localization Using Semantically Segmented Images," 2018 IEEE ICRA, Brisbane, 2018, pp. 6484-6490.

Now, assume that we have a camera on our vehicle that can observe this scene.

- We can then try to align our pose such that if we project our semantic 3D map onto our camera image, our building points will fall on buildings and our tree points will land on trees, and so on.
- Now here is one such image from a camera observing the same scene as our map, where we have semantically labeled each pixel in the image with the same classes that we have in our map.
- We do this by using a **deep convolutional neural network** that takes an **ordinary image as input, and outputs a semantic label of each pixel**.



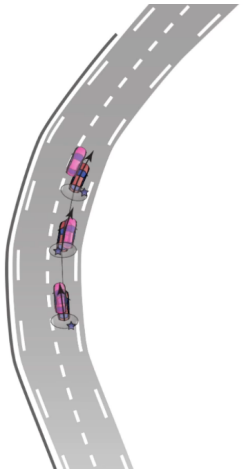
- In the image to the right, we have overlaid the semantic image on the original image where we can see that the network manages to fairly well label each pixel.

- So how can we use this to position our autonomous vehicle?
  - Using a **particle filter**, we would represent the pose of our vehicle, so position and heading, using a whole bunch of more or less randomly chosen particles.

1.3 Course structure and learning outcome

The content of this course divided into 7 sections:

1. Course introduction and a primer in statistics
2. Bayesian statistics
3. State space models and optimal filters
4. The Kalman filter and its properties
5. Motion and measurement models
6. Nonlinear Gaussian filters
7. Particle filters



So the course is divided into seven sections.

1. **Course introduction and a primer in statistics**
2. **Bayesian statistics**
   - You will learn the basic concepts of Bayesian statistics, which is Bayes' rule.
   - able to explain the fundamental principles in Bayesian estimation.
3. **State space models and design optimal filters**
   - The fundamental assumptions that are made and be able to derive expressions for an optimal filter based on these assumptions.
4. **The Kalman filter and its properties**
   - How to apply it on linear state space models.

Our state space models consist of two basic models,

5. **Motion model and measurement model**
   - Commonly used motion and measurement models for automotive applications and show how we can derive time discrete motion models from time continuous dynamic models.
   - Able to summarize and compare the most typical motion models in automotive systems in order to know when to use them in practical problems. You will also be able to describe and model commonly used sensor measurements.
6. **Nonlinear Gaussian filters**
   - The extended Kalman filter, as well as the unscented Kalman filter.
7. **The particle filters**
   - A nonlinear filter but, the particle filter do not try to describe the results using a Gaussian distribution, but rather as a set of weighted particles as we saw in the self localization problem.

2 Random Variables

- Have a good understanding of some basic statistical concepts.
- random variables
  - When we do nonlinear filtering, we need them to describe the quantity that we're interested in, for example, the position of a vehicle.

1. Discrete Value Random variables

DISCRETE-VALUED RANDOM VARIABLES

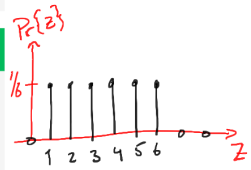
Probability mass function, pmf

- The *probability mass function* (pmf) of a discrete-valued random variable is denoted, $\Pr\{z\}$ or $\mathrm{P}\{z\}$, where

$$\Pr\{z = i\} \geq 0 \qquad \text{for all } i$$
$$\sum_z \Pr\{z\} = 1.$$

Example: A fair dice

$$\Pr\{z = i\} = \begin{cases} \frac{1}{6} & \text{if } i = 1, 2, \ldots, 6 \\ 0 & \text{otherwise.} \end{cases}$$



2. Continous valued random

CONTINUOUS-VALUED RANDOM VARIABLES

Probability density function (pdf)

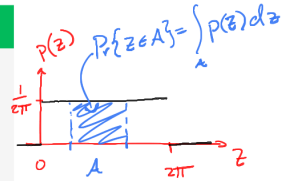
- The *probability density function* (pdf) of a continuous-valued random variable is denoted $p(z)$, where

$$p(z) \geq 0 \text{ for all } z, \quad \text{and} \quad \int p(z)\, dz = 1$$

Example: Uniform distribution

- Suppose z is uniformly distributed between 0 and 2π, it's pdf is then

$$p(z) = \begin{cases} \frac{1}{2\pi} & \text{if } 0 \leq z < 2\pi \\ 0 & \text{otherwise.} \end{cases}$$



- Now, the probability mass that z is in this interval.
- So z is in A and can be calculated by integrating pdf over the interval A.

2.2 Conditional, Joint and marginal distributions

- In this course, we're very interested in how the distribution of two or more random variables depend on each other.
- We do this by mainly considering the so-called **conditional** and **joint distributions**.
- Additionally, we are also interested in knowing the so-called **marginal distribution**, which is the isolated distribution of a single random variable, where we have removed the influence of all the other variables.

1. CONDITIONAL DISTRIBUTIONS

CONDITIONAL DISTRIBUTIONS

- Conditional distributions are indispensable components in sensor fusion, filtering and Bayesian estimation in general.

Conditional distribution (product rule)

- Let x and z be two random variables with the joint pdf $p(x, z)$.
- The *conditional density function*, $p(z|x)$, is defined through

$$p(x, z) = p(z|x)p(x),$$

and if $p(x) \neq 0$ this implies that

$$p(z|x) = \frac{p(x, z)}{p(x)}.$$

$$p(z|x=x') = \frac{p(x', z)}{p(x')} \propto p(x', z)$$

- Interpretation: $p(z|x)$ describes the distribution of z given that x is known.

CONDITIONAL DISTRIBUTIONS

Example: Candy problem

- Every day Sara decides how many pieces of candy she can have for an after lunch snack.
- With 40% probability she tosses a coin, heads means 1 piece and tails means 0 pieces
- With 60% probability she throws a dice (number on the dice = number of candies).
- If z denotes number of candies she eats

$$\Pr\{z = i | \text{Sara tosses a coin}\} = \begin{cases} 0.5 & \text{if } i = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\Pr\{z = i | \text{Sara throws a dice}\} = \begin{cases} 1/6 & \text{if } i = 1, 2, \ldots, 6 \\ 0 & \text{otherwise} \end{cases}$$

2. LAW OF TOTAL PROBABILITY

LAW OF TOTAL PROBABILITY

- Many important results in non-linear filtering is obtained from the law of total probability.

Law of total probability (sum rule)

- If x takes values in a set S_x, the law of total probability states that

$$\text{Discrete:} \quad \Pr\{z\} = \sum_{x \in S_x} \Pr\{x, z\} = \sum_{x \in S_x} \Pr\{z|x\} \Pr\{x\}$$

$$\text{Continuous:} \quad p(z) = \int_{x \in S_x} p(x, z)\, dx = \int_{x \in S_x} p(z|x) p(x)\, dx$$

EXPECTED VALUE AND COVARIANCE

- Probability distributions are often characterized by their mean vectors and covariance matrices.

Expected value (mean vector)

- The expected value (mean) of a random vector $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$ is

$$\mathbb{E}\{\mathbf{x}\} = \int \mathbf{x}\, p(\mathbf{x})\, d\mathbf{x}$$

where $\int d\mathbf{x}$ is shorthand for $\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} dx_1 \ldots dx_m$.

Covariance matrix

- The covariance matrix is ($\mathbf{x}$ is a column vector)

$$\text{Cov}\{\mathbf{x}\} = \mathbb{E}\left\{[\mathbf{x} - \mathbb{E}\{\mathbf{x}\}][\mathbf{x} - \mathbb{E}\{\mathbf{x}\}]^T\right\}$$

- For discrete-valued random variables the above integrals are replaced by the corresponding summations.

LAW OF TOTAL PROBABILITY

Example: Candy pmf

- To calculate the pmf for the number of candies we use

$$\Pr\{z\} = \sum_{x \in S_x} \Pr\{z|x\} \Pr\{x\},$$

where x is either 'Sara tosses a coin' or 'Sara throws a dice'.

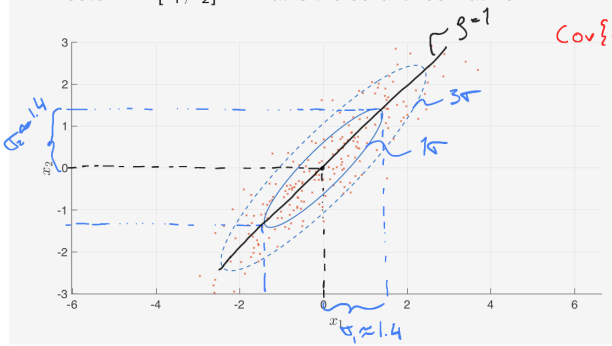
- Hint: First calculate the joint probability of $\Pr\{z, x\}$

$\Pr\{z, x\}$	z=0	z=1	z=2	z=3	z=4	z=5	z=6	$\Pr\{x\}$
x = coin	0.2	0.2	0	0	0	0	0	0.4
x = dice	0	0.1	0.1	0.1	0.1	0.1	0.1	0.6
$\Pr\{z\}$	0.2	0.3	0.1	0.1	0.1	0.1	0.1	

1. GUESS THAT COVARIANCE

Example: Guess that covariance

- Suppose we have independent samples from a zero-mean random vector $\mathbf{x} = [x_1, x_2]^T$. What is the covariance matix of $\mathbf{x}$?



- Now, if we look at our samples, we see that **there is quite a significant positive correlation here**.
- So if we **have a high value on x1**, there's a **high probability that x2 also has a high value.**
- But it's definitely not a deterministic mapping, right, as if it were, all the samples here would collapse and fall on the line like this.

- Sometimes we're interested in finding the **expected value of a random variable numerically**.

2.3 Expectation, covariance and the Gaussian distribution

- Perhaps **we are not able to solve the involved integrals explicitly**, or more commonly, when we do not actually know the underlying distribution but have access to a large number of samples from it.
- So in this case, we can use the **law of large numbers**, which states that sample averages converge to expected values for large sample sizes.

LAW OF LARGE NUMBERS

- The law of large numbers states that sample averages converge to expected values.

Law of large numbers

- If $x_1, x_2, \ldots$ are independent and identically distributed random variables distributed according to $p(x)$, then

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} x_i = \mathbb{E}_{p(x)}\{x\}.$$

Example: Throwing a dice many times...

- ...the average face value converges to the expected value

$$\frac{1+2+3+4+5+6}{6} = 3.5$$

GAUSSIAN DISTRIBUTIONS

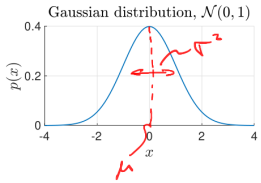
- The most important distribution is the Gaussian distribution (at least in this course).

Gaussian distribution

- We write $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q})$ to denote that $\mathbf{x}$ is a Gaussian random variable with mean $\boldsymbol{\mu}$ and covariance $\mathbf{Q}$.
- The pdf of $\mathbf{x}$ is

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{Q}) = \frac{1}{\sqrt{|2\pi\mathbf{Q}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{Q}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $|\cdot|$ denotes the determinant.



- Note: the pdf of a Gaussian random variable is completely determined by its mean and its covariance matrix.

- We denote that x is a Gaussian random variable with
  - mean μ
  - covariance $\mathbf{Q}$.

- And the pdf of x is then denoted like this, so p of x, where we use this notation here to denote the Gaussian pdf as a function of x with the parameters μ and Q.

> $p(x) = N(x; \mu, Q)$