Meeting:First meeting

Organiser:Second user

Secretary:LIM G WEI

Date:13 Jun 2013

Time:10:56 am

Venue:gg

Attendance
Present:LIM G WEI,President
Second user,Secretary
Absent :

1.Introduction

2.first discussion

2.1.Final Year Project I

2.1.1.first keypoint for fyp1
2.1.2.second point

Action taken: approved

Taken by: Second user

2.2.Final Year Project II

3.second discussion

3.1.Final Year Project III

fyp

Wong Li Qi B200263C BoSE20-C

Question 1 – 15 marks

A production system maintains a set of rules about the characteristics of fruits as follows:

Rule 1:	IF	Shape = long AND
		Colour = green OR yellow
	THEN	Fruit = banana
Rule 2:	IF	Shape = round OR oblong AND
		Diameter > 4 inches
	THEN	Fruitclass = vine
Rule 3:	IF	Shape = round AND
		Diameter < 4 inches
	THEN	Fruitclass = tree
Rule 4:	IF	Seedcount = 1
	THEN	Seedclass = stonefruit
Rule 5:	IF	Seedcount > 1
	THEN	Seedclass = multiple
Rule 6:	IF	Fruitclass = vine AND
		Colour = green
	THEN	Fruit = watermelon
Rule 7:	IF	Fruitclass = vine AND
		Surface = smooth AND
		Colour = yellow
	THEN	Fruit = honeydew
Rule 8:	IF	Fruitclass = vine AND
		Surface = rough AND
		Colour = tan
	THEN	Fruit = cantaloupe
Rule 9:	IF	Fruitclass = tree AND
		Colour = orange AND
		Seedclass = stonefruit
	THEN	Fruit = apricot
Rule 10:	IF	Fruitclass = tree AND
		Colour = orange AND
		Seedclass = multiple
	THEN	Fruit = orange
Rule 11:	IF	Fruitclass = tree AND
		Colour = red AND
		Seedclass = stonefruit
	THEN	Fruit = cherry

Rule 12: IF Fruitclass = tree AND
Colour = orange AND
Seedclass = stonefruit
THEN Fruit = peach

Rule 13: IF Fruitclass = tree AND
Colour = red OR yellow OR green AND
Seedclass = multiple
THEN Fruit = apple

Rule 14: IF Fruitclass = tree AND
Colour = purple AND
Seedclass = stonefruit
THEN Fruit = plum

i) Use ***FORWARD CHAINING*** to describe the production system table including its working memory, conflict set and rule fired to establish a fruit. Initial data given is :

Shape = round
Diameter > 4 inches
Surface = smooth
Colour = yellow

Terminate when the final value for Fruit in the working memory. [6 marks]

Iteration #	Working memory	Conflict set	Rule fired
0	Shape = round	2,3	Halt
1	Diameter>4 inches	2	2
2	Fruitclass = vine	6,7,8	Halt
3	Surface = smooth	7	Halt
4	Color = yellow	7	7
5	Fruit = honeydew		Halt

ii) Given the fruit to search is apple, use ***BACKWARD CHAINING*** to describe the production system table including its working memory, conflict set and rule fired toestablish the initial data for this fruit.

State the initial facts required to establish that the fruit searched is an apple. [9 marks]

Iteration #	Working memory	Conflict set	Rule fired
0	Fruit = apple Seedclass = multiple Colour = red OR yellow OR green Fruitclass = tree Seedcount > 1 Diameter < 4 inches Shape = round	 13 11,13 9,10,11,12,13,14 5 3 2,3	Halt 13 Halt Halt 5 3 Halt

The initial facts required to establish fruit to search is apple are:

Shape = round
Diameter < 4 inches
Seedcount > 1
Colour = Red or Yellow or Green

Wong Li Qi B200263C BoSE20-C

Question 1 – 15 marks

A production system maintains a set of rules about the characteristics of fruits as follows:

Rule 1:	IF	Shape = long AND
		Colour = green OR yellow
	THEN	Fruit = banana
Rule 2:	IF	Shape = round OR oblong AND
		Diameter > 4 inches
	THEN	Fruitclass = vine
Rule 3:	IF	Shape = round AND
		Diameter < 4 inches
	THEN	Fruitclass = tree
Rule 4:	IF	Seedcount = 1
	THEN	Seedclass = stonefruit
Rule 5:	IF	Seedcount > 1
	THEN	Seedclass = multiple
Rule 6:	IF	Fruitclass = vine AND
		Colour = green
	THEN	Fruit = watermelon
Rule 7:	IF	Fruitclass = vine AND
		Surface = smooth AND
		Colour = yellow
	THEN	Fruit = honeydew
Rule 8:	IF	Fruitclass = vine AND
		Surface = rough AND
		Colour = tan
	THEN	Fruit = cantaloupe
Rule 9:	IF	Fruitclass = tree AND
		Colour = orange AND
		Seedclass = stonefruit
	THEN	Fruit = apricot
Rule 10:	IF	Fruitclass = tree AND
		Colour = orange AND
		Seedclass = multiple
	THEN	Fruit = orange
Rule 11:	IF	Fruitclass = tree AND
		Colour = red AND
		Seedclass = stonefruit
	THEN	Fruit = cherry

Rule 12: IF Fruitclass = tree AND
Colour = orange AND
Seedclass = stonefruit
THEN Fruit = peach

Rule 13: IF Fruitclass = tree AND
Colour = red OR yellow OR green AND
Seedclass = multiple
THEN Fruit = apple

Rule 14: IF Fruitclass = tree AND
Colour = purple AND
Seedclass = stonefruit
THEN Fruit = plum

i) Use ***FORWARD CHAINING*** to describe the production system table including its working memory, conflict set and rule fired to establish a fruit. Initial data given is :

Shape = round
Diameter > 4 inches
Surface = smooth
Colour = yellow

Terminate when the final value for Fruit in the working memory. [6 marks]

Iteration #	Working memory	Conflict set	Rule fired
0	Shape = round	2,3	Halt
1	Diameter>4 inches	2	2
2	Fruitclass = vine	6,7,8	Halt
3	Surface = smooth	7	Halt
4	Color = yellow	7	7
5	Fruit = honeydew		Halt

ii) Given the fruit to search is apple, use ***BACKWARD CHAINING*** to describe the production system table including its working memory, conflict set and rule fired toestablish the initial data for this fruit.

State the initial facts required to establish that the fruit searched is an apple. [9 marks]

Iteration #	Working memory	Conflict set	Rule fired
0	Fruit = apple		Halt
	Seedclass = multiple	13	13
	Colour = red OR yellow OR green	11,13	Halt
	Fruitclass = tree	9,10,11,12,13,14	Halt
	Seedcount > 1	5	5
	Diameter < 4 inches	3	3
	Shape = round	2,3	Halt

The initial facts required to establish fruit to search is apple are:

Shape = round
Diameter < 4 inches
Seedcount > 1
Colour = Red or Yellow or Green

What drives research in self-driving cars? (Part 1+2)

Article · April 2014

1 author:

Göde Both
Humboldt-Universität zu Berlin

This work has been previously published in two parts on Platypus, the official blog of the Committee on the Anthropology of Science, Technology, and Computing. Please, cite the original sources:

http://blog.castac.org/2014/04/what-drives-research-in-self-driving-cars-part-1-two-major-events/

http://blog.castac.org/2014/04/what-drives-research-in-self-driving-cars-part-2-surprisingly-not-machine-learning/

What Drives Research in Self-driving Cars? (Part 1: Two Major Events)

Göde Both April 1, 2014

Self-driving cars (aka driverless cars or autonomous vehicles) are among the most visible faces of Artificial Intelligence (AI) today. In continuation with Shreeharsh Kelkar's excellent post on Artificial Intelligence last month, I would like to pick up his lead and complicate yet another story of AI – the story of the relationship between software developers and machine learning algorithms. For this purpose, I will use my ongoing field work among members of an academic research group as an example. The work of this particular research group centers around an experimental vehicle – a self-driving car (hereafter: SDC).

During my field work I experienced a couple of surprises which challenged my earlier assumptions about the research on SDCs. That is, I previously assumed that the many promises invested in machine learning would lead to its extensive use in the experimental vehicles. However, this is not the case. I started to wonder why the researchers refrain from using machine learning even though they are officially members of an AI department. After introducing you to the field of SDC research in this first part, I will provide you with three tentative answers in an upcoming second post.

A Big Mess – Situating Research in Self-driving Cars

Whenever I try to situate research in SDCs, I am confronted by a big mess. If you follow its ties to the world of cars and driving, you end up with a versatile and pervasive artifact – the automobile – with so many layers of meanings and diverse communities of practices. With 1 billion of them existing, cars dominate transportation in many countries of the world. SDC research is tied to multiple contexts. Several origin stories circulate in the field. That is why, in this post I do not even try to map the different contexts in which cars and self-driving car research are situated. I also refrain from telling one of the origin stories of self-driving-cars, although I do recommend reading "Driving the Dream" by Jameson M. Wetmore. His history of Automatic Highway Systems can be read as a marginalized back-story to self-driving cars. Instead of doing all the above things, I want to introduce you to this field by briefly drawing your attention towards two more recent events which turned out to be highly influential among the members.

Major Event #1 – DARPA Grand Challenges

In 2000 U.S. Congress passed a mandate that by 2015 one-third of the operational ground combat vehicles of U.S. military are to be unmanned. In response to this, DARPA organized a series of "Grand Challenges" in 2004, 2005, and 2007. During these events robotics researchers competed with their autonomous vehicles for a cash prizes by solving tasks, likewise completing an autonomous drive through the desert. In this case *autonomous* means that the cars are unmanned and perform without remote control by the competing teams.

In SDC research this series of events is usually told in the form of a narrative of progress. During the first Grand Challenge (2004) no vehicle finished the race through the desert. The following year a handful of vehicles succeeded. By 2007 several teams completed a series of tasks in a mixed human and nonhuman scenario which simulated suburban traffic at an abandoned airbase. Among the members of the field it is widely assumed that this development – in only 3 years – provides a proof of concept. From their perspective mastering everyday traffic is not only feasible for SDCs but only a matter of time until the technology has fully matured.

and the winner is…

Stanford Vehicle for the DARPA challenge. Photo taken from Flickr. Licensed under a Creative Commons license.

Even if one doubts this techno-optimism, one thing is clear. DARPA succeeded in enrolling a large bunch of influential researchers and thus, enlarged a research community that has been marginal before the Grand Challenges. In addition, by creating these events it steered the research community in a specific direction. A direction that primarily serves the interest of the military applications of SDCs. Steven E. Shladover argues:

The needs of the civilian road transportation and military transportation communities, as identified above, are strikingly different from each other. […] When these differences are considered in toto, significant doubts arise about the extent to which the technologies and designs developed to meet the one set of needs will be transferable to the other domain. (2009:13)

Major Event #2 – Google's Self-driving Car Project

In 2010 Google announced its SDC project. At the time of this writing Google operates the largest fleet of experimental vehicles in the world. However, this is not the sole reason why Google is a big player in the field. It can be argued that Google has made a major contribution to the normalization of SDC research. Until Google's project very few teams conducted test-drives in real-life traffic. Google has not only tested its SDCs in over 300,000 miles of driving under everyday conditions but also successfully lobbied for new traffic laws in Nevada and California.

Due to its media attention the Google project has become a technological standard to which other SDC projects are measured up against. Let's have a look at how Google performs its test-drives.

The Google Self-driving Car with its sensors. Original photo taken from Flickr and subsequently modified. Released under the CC license.

Above you see a picture of an experimental vehicle, in this case a modified Toyata Prius. The arrows point towards a variety of sensors: laser range finders on the roof, RADAR on the side, cameras behind the windshield. What you don't see is an high-accuracy odometer and a differential GPS receiver.

How does it work?

In short, the SDC works like this. All the signals from the sensors are processed by computers to make sense of its surrounding environment and to localize itself on a digital high-precision map. On the basis of the sensor data and the position on the map, the SDC makes decision on what to do next. This is only a very brief description of what actually happens, but it will suffice for now. As the text in the box at the bottom of the picture explains, the SDC is never driving by itself in everyday traffic. It is closely monitored by a safety driver and an engineer while driving in public. The safety driver can resume full control of the car whenever s/he wishes. Whenever the SDC fails to master a given situation, the safety driver will disconnect the computers and resume driving manually.

I thought that autonomous driving research would incorporate a great deal of machine learning algorithms. But there I was surprised. In part two of my post, I will talk about the relationship between software developers and machine learning algorithms in the domain of SDC research.

What Drives Research in Self-driving Cars? (Part 2: Surprisingly not Machine Learning)

Göde Both April 3, 2014

In the first part of this article, I wrote about how two major events shaped research in self-driving cars: the DARPA Grand Challenges and Google's Self-driving Car (hereafter: SDC) project. In this post, I will talk about my surprise at the unfulfilled yet pervasive promises of machine learning in SDC research.

SDC researchers often attribute the advancement in autonomous driving to the successful deployment of machine learning. For instance, Sebastian Thrun, a former robotics researcher at Stanford and one of the driving forces behind the "Google driverless car" project, told Wired Magazine that "data can make better rules", implying that SDCs ought to learn from experience rather than pre-programmed rules. In a technology assessment of self-driving cars, the authors conclude that "allowing the vehicles to write their 'own code' through machine learning" (Moore/Lu 2011:6) will lead to more dependable SDCs because the behavior is derived from the experiences of the robot rather from the human developer's intuitions.

The promises of machine learning in the field of self-driving cars are based on the belief that the computer will do most of the necessary programming by itself, thereby leading to more efficient and dependable programming solutions. In the field, any human involvement is often figured as a harmful and "inelegant" way of getting things done.

How do you teach a computer to see pedestrians?

For the remainder of this post, I will try to capture the practitioner's perspective when deploying machine learning. For example, how do the researchers teach a self-driving car to detect pedestrians? The short answer is, computer systems are trained by labeled data sets. The first step is obtaining the training data. Researchers take the research vehicle for a spin in real-life traffic and record a video with a fixed mounted camera. The data may look like the video stills in middle row of the collection of pictures below. (These stills are actually taken from freely available training set provided by the German car manufacturer Daimler, better known for its Mercedes brand).

Image taken from Dariu M. Gavrila

The second step is labeling the data. Positive samples, a.k.a. regions of the still which show a pedestrian (top row), are labeled as such by a programmer. Negative samples, i.e. stills that do not include pedestrians, are labeled as well (middle row).

During the third step, the researchers train a classifier. The classifier is a piece of software which settles the question of whether or not a pedestrian is present in a region of the video stream. The training is conducted by feeding the software with positive and negative samples, i.e. pictures with and without pedestrians in them. A mathematical function is derived from all the samples that best mimics the prior labeling of the pictures.

To use the classifier for pedestrian detection, you ask it to process certain regions of the video stream. The successful detection of a pedestrian is usually visualized by a bounding box, like you can see on the pictures in the bottom row.

Now to complicate this story of of how machine learning is used, I would like to share three surprising discoveries which occurred during my fieldwork and which shed a different light on the promises of machine learning. That is, the researchers refrain from using machine learning even though they are members of an AI department. How do they justify their decisions?

Labeling is “inhumane” work

As we have seen during the previous example, machine learning may involve a considerable amount of work done by humans. Labeling the data in cases like these cannot be automated. During an ethnographic interview, a researcher told me that he single-handedly labeled 3000 pictures of cars to generate a large enough training set. When I asked whether he would delegate the work to students researchers in the future, he shook his head. He described this kind of work as “inhumane” (German: unmenschlich). He would neither ask students to perform it nor hire Amazon’s Mechanical Turkers to do it. In conclusion, contrary to the promise of simply delegating work to the machine, new forms of undesirable work crop up.

Obscurity of the Classifier

Another disadvantage of machine learning as compared to – what practitioners call – “manual programming,” is that machine learning algorithms are often black-boxes. One researcher explained that machine learning is unreliable and possibly dangerous because

You don’t know what it learns (German “Du weißt nicht, was e[i]r lernt)

The researcher argues that there might be situations in real-life traffic which are not covered by the training samples. Hence, in these cases, the behavior of the classifier and, by implication, the SDC is unpredictable. For example, a SDC might make an emergency stop because it falsely detects a pedestrian in its way or even worse, it fails to detect a pedestrian. Both cases might lead to accidents. What surprised me here was how machine learning is seen as a source of uncertainty and danger: the behavior of the classifier is, at least in part, obscure to the developer who designed it.

Machine Learning impedes tweaking

The opacity of the classifier (discussed above) has another crucial implication. In machine learning, unlike “manual” programming, tweaking the parameters of the classifier is difficult. Tweaking is a common – if also a somewhat contested practice – in robotics. Although it is considered “inelegant”, it is often necessary in order to get the robot to work. However, making changes to the classifier or to the training set may not only eradicate unwanted behaviors, but also the desired characteristics of the classifier. Hence, adjusting a classifier to new situations involves the risk of weakening its performance.

---

[i] The literal translation from German is “he” (er) not “it”. This particular gendering of the classifier may be attributed to German grammar. However, it may also reflect the longstanding history of the computer/robot as a masculine artifact.