Ghi chú của một coder

Vũ Anh

Tháng 01 năm 2018

Mục lục

Mục lục 3

I Toán 4

1 Xác suất 5
- 1.1 Probability Distributions . . . 5
  - 1.1.1 Event Spaces . . . 5
  - 1.1.2 Probability Distributions . . . 6
- 1.2 Basic Concepts in Probability . . . 7
  - 1.2.1 Conditional Probability . . . 7
- 1.3 Chain Rule and Bayes Rule . . . 7
- 1.4 Random Variables and Joint Distributions . . . 8
  - 1.4.1 Motivation . . . 8
  - 1.4.2 What Is a Random Variable? . . . 9
- 1.5 Independence and Conditional Independence . . . 11
- 1.6 Querying a Distribution . . . 13
  - 1.6.1 5.1 Probability Queries . . . 13
  - 1.6.2 5.2 MAP Queries . . . 13
  - 1.6.3 5.3 Marginal MAP Queries . . . 14
- 1.7 Continuous Spaces . . . 14
- 1.8 Kì vọng và phương sai . . . 16
  - 1.8.1 Kì vọng . . . 16
  - 1.8.2 Phương sai . . . 17
- 1.9 Các hàm phân phối thông dụng . . . 18
  - 1.9.1 Biến rời rạc . . . 18
  - 1.9.2 Biến ngẫu nhiên Bernoulli . . . 19
  - 1.9.3 Phân phối Bernoulli . . . 19
  - 1.9.4 Phân phối Binomial . . . 20
  - 1.9.5 Phân phối Poisson . . . 20
  - 1.9.6 Phân phối Geometric . . . 20
  - 1.9.7 Phân phối Negative Binomial . . . 20
  - 1.9.8 Phân phối Hyper Geometric . . . 20
  - 1.9.9 Biến liên tục . . . 20
  - 1.9.10 Phân phối đều . . . 20

2 Thống kê 22

II Linh tinh **23**

3 Nghiên cứu **24**
3.1 Các công cụ 24
3.1.1 Google Scholar & Semantic Scholar 24
3.1.2 Mendeley 24
3.1.3 Hội nghị và tạp chí 24
3.1.4 Câu chuyện của Scihub 25
3.2 Làm sao để nghiên cứu tốt 25
3.3 Sách giáo khoa 25
3.4 Lượm lặt 26
3.5 Thuyết trình 26

4 Nghề lập trình **27**

5 Latex **28**

6 Chào hàng **30**

7 Phát triển phần mềm **31**

8 Phương pháp làm việc **32**

9 Agile **34**
9.1 Introduction 34
9.2 Team 35
9.3 Artifacts 37
9.4 Meetings 37

10 Docker **41**
10.1 Get Started 41
10.2 Dev 43
10.3 Dockerfile 43
10.4 Container 44
10.5 Compose 45
10.6 Swarm 46
10.7 UCP 47
10.8 Labs 47
10.9 GUI 48

11 Lean **50**
11.1 Lean Canvas 50
11.2 Workflow 50
11.3 Validated Learning 51

Tài liệu **52**

Chỉ mục **53**

Ghi chú **54**

Phần I

Toán

Chương 1

Xác suất

Xác suất là công cụ để mô hình hóa các sự vật tưởng như ngẫu nhiên.

19/01/2018 Nay tìm được khóa CS 109 của bác Chris Piech quá hay

1.1 Probability Distributions

When we use the word "probability" in day-to-day life, we refer to a degree of confidence that an event of an uncertain nature will occur. For example, the weather report might say "there is a low probability of light rain in the afternoon." Probability theory deals with the formal foundations for discussing such estimates and the rules they should obey. Before we discuss the representation of probability, we need to define what the events are to which we want to assign a probability. These events might be different outcomes of throwing a die, the outcome of a horse race, the weather configurations in California, or the possible failures of a piece of machinery.

1.1.1 Event Spaces

event Formally, we define events by assuming that there is an agreed upon space of possible outcomes, outcome space which we denote by . For example, if we consider dice, we might set = 1, 2, 3, 4, 5, 6. In the case of a horse race, the space might be all possible orders of arrivals at the finish line, a much larger space.

measurable event In addition, we assume that there is a set of measurable events S to which we are willing to assign probabilities. Formally, each event S is a subset of . In our die example, the event 6 represents the case where the die shows 6, and the event 1, 3, 5 represents the case of an odd outcome. In the horse-race example, we might consider the event "Lucky Strike wins," which contains all the outcomes in which the horse Lucky Strike is first. Probability theory requires that the event space satisfy three basic properties: • It contains the empty event , and the trivial event . • It is closed under union. That is, if , S, then so is . • It is closed under complementation. That is, if S, then so is . The requirement that the event space is closed under union and complementation implies that it is also closed under other Boolean operations, such as intersection and set dierence.

1.1.2 Probability Distributions

Definition 2.1 A probability distribution P over (, S) is a mapping from events in S to real values that satisfies probability distribution the following conditions: • P() 0 for all S. • P() = 1. • If , S and = , then P( ) = P() + P(). The first condition states that probabilities are not negative. The second states that the "trivial event," which allows all possible outcomes, has the maximal possible probability of 1. The third condition states that the probability that one of two mutually disjoint events will occur is the sum of the probabilities of each event. These two conditions imply many other conditions. Of particular interest are P() = 0, and P( ) = P() + P() P( ).

1.3 Interpretations of Probability Before we continue to discuss probability distributions, we need to consider the interpretations that we might assign to them. Intuitively, the probability P() of an event quantifies the degree of confidence that will occur. If P() = 1, we are certain that one of the outcomes in occurs, and if P() = 0, we consider all of them impossible. Other probability values represent options that lie between these two extremes. This description, however, does not provide an answer to what the numbers mean. There are two common interpretations for probabilities. frequentist The frequentist interpretation views probabilities as frequencies of events. More precisely, the interpretation probability of an event is the fraction of times the event occurs if we repeat the experiment indefinitely. For example, suppose we consider the outcome of a particular die roll. In this case, the statement P() = 0.3, for = 1, 3, 5, states that if we repeatedly roll this die and record the outcome, then the fraction of times the outcomes in will occur is 0.3. More precisely, the limit of the sequence of fractions of outcomes in in the first roll, the first two rolls, the first three rolls, . . ., the first n rolls, . . . is 0.3.

The frequentist interpretation gives probabilities a tangible semantics. When we discuss concrete physical systems (for example, dice, coin flips, and card games) we can envision how these frequencies are defined. It is also relatively straightforward to check that frequencies must satisfy the requirements of proper distributions. The frequentist interpretation fails, however, when we consider events such as "It will rain tomorrow afternoon." Although the time span of "Tomorrow afternoon" is somewhat ill defined, we expect it to occur exactly once. It is not clear how we define the frequencies of such events. Several attempts have been made to define the probability for such an event by finding a reference class reference class of similar events for which frequencies are well defined; however, none of them has proved entirely satisfactory. Thus, the frequentist approach does not provide a satisfactory interpretation for a statement such as "the probability of rain tomorrow afternoon is 0.3." An alternative interpretation views probabilities as subjective degrees of belief. Under subjective interpretation this interpretation, the statement P() = 0.3 represents a subjective statement about one's own degree of belief that the event will come about. Thus, the statement "the probability of rain tomorrow afternoon is 50 percent" tells us that in the opinion of the speaker, the chances of rain and no rain tomorrow afternoon are the same. Although tomorrow afternoon will occur only once, we can still have uncertainty about its outcome, and represent it using numbers (that is, probabilities). This description still does not resolve what exactly it means to hold a particular degree of belief. What stops a person from stating that the probability that Bush will win the election is 0.6 and the probability that he will lose is 0.8?

The source of the problem is that we need to explain how subjective degrees of beliefs (something that is internal to each one of us) are reflected in our actions. This issue is a major concern in subjective probabilities. One possible way of attributing degrees of beliefs is by a betting game. Suppose you believe that P() = 0.8. Then you would be willing to place a bet of $1 against $3. To see this, note that with probability 0.8 you gain a dollar, and with probability 0.2 you lose $3, so on average this bet is a good deal with expected gain of 20 cents. In fact, you might be even tempted to place a bet of $1 against $4. Under this bet the average gain is 0, so you should not mind. However, you would not consider it worthwhile to place a bet $1 against
4 and 10 cents, since that would have negative expected gain. Thus, by finding which bets you are willing to place, we can assess your degrees of beliefs. The key point of this mental game is the following. If you hold degrees of belief that do not satisfy the rule of probability, then by a clever construction we can find a series of bets that would result in a sure negative outcome for you. Thus, the argument goes, a rational person must hold degrees of belief that satisfy the rules of probability.1 In the remainder of the book we discuss probabilities, but we usually do not explicitly state their interpretation. Since both interpretations lead to the same mathematical rules, the technical definitions hold for both interpretations.

1.2 Basic Concepts in Probability

1.2.1 Conditional Probability

To use a concrete example, suppose we consider a distribution over a population of students taking a certain course. The space of outcomes is simply the set of all students in the population. Now, suppose that we want to reason about the students' intelligence and their final grade. We can define the event
$\alpha todenote\imath allstudentswithgradeA, \jmath andtheeventtodenote\imath allstudentswithhighintelligence.\jmath Usingourdistrib$
$P()P()(2.1)Thatis, theprobabilitythatistruegiventhatweknowistherelativeproportionofoutcomessatisfying$
$0.)Theconditionalprobabilitygivenanevent(say)satisfiesthepropertiesofdefinition2.1(seeexercise2.4), and$

1.3 Chain Rule and Bayes Rule

From the definition of the conditional distribution, we immediately see that P( ) = P()P(|). (2.2) chain rule This equality is known as the chain rule of conditional probabilities. More generally, if 1, . . . , k are events, then we can write P(1 . . . k) = P(1)P(2 | 1) · · · P(k | 1 . . . k1). (2.3) In other words, we can express the probability of a combination of several events in terms of the probability of the first, the probability of the second given the first, and so on. It is important to notice that we can expand this expression using any order of events — the result will remain the same. Bayes' rule Another immediate consequence of the definition of conditional probability is Bayes' rule P(|) = P(|)P() P()
A more general conditional version of Bayes' rule, where all our probabilities are conditioned on some background event , also holds: P(|) = P(|)P(|) P(|) . Bayes' rule is important in that it allows us to compute the conditional probability P(|) from the "inverse" conditional probability P(|). Example

2.1 Consider the student population, and let Smart denote smart students and GradeA denote students who got grade A. Assume we believe (perhaps based on estimates from past statistics) that P(GradeA | Smart) = 0.6, and now we learn that a particular student received grade A. Can we estimate the probability that the student is smart? According to Bayes' rule, this depends on prior our prior probability for students being smart (before we learn anything about them) and the prior probability of students receiving high grades. For example, suppose that P(Smart) = 0.3 and P(GradeA) = 0.2, then we have that P(Smart | GradeA) = 0.6 0.3/0.2 = 0.9. That is, an A grade strongly suggests that the student is smart. On the other hand, if the test was easier and high grades were more common, say, P(GradeA) = 0.4 then we would get that P(Smart | GradeA) = 0.6 0.3/0.4 = 0.45, which is much less conclusive about the student. Another classic example that shows the importance of this reasoning is in disease screening. To see this, consider the following hypothetical example (none of the mentioned figures are related to real statistics). Example 2.2 Suppose that a tuberculosis (TB) skin test is 95 percent accurate. That is, if the patient is TB-infected, then the test will be positive with probability 0.95, and if the patient is not infected, then the test will be negative with probability 0.95. Now suppose that a person gets a positive test result. What is the probability that he is infected? Naive reasoning suggests that if the test result is wrong 5 percent of the time, then the probability that the subject is infected is 0.95. That is, 95 percent of subjects with positive results have TB. If we consider the problem by applying Bayes' rule, we see that we need to consider the prior probability of TB infection, and the probability of getting positive test result. Suppose that 1 in 1000 of the subjects who get tested is infected. That is, P(TB) = 0.001. What is the probability of getting a positive test result? From our description, we see that $0.001 \cdot 0.95$ infected subjects get a positive result, and $0.999 \cdot 0.05$ uninfected subjects get a positive result. Thus, P(Positive) = 0.0509. Applying Bayes' rule, we get that P(TB | Positive) = $0.001 \cdot 0.95/0.0509$ 0.0187. Thus, although a subject with a positive test is much more probable to be TB-infected than is a random subject, fewer than 2 percent of these subjects are TB-infected.

1.4 Random Variables and Joint Distributions

1.4.1 Motivation

Our discussion of probability distributions deals with events. Formally, we can consider any event from the set of measurable events. The description of events is in terms of sets of outcomes. In many cases, however, it would be more natural to consider attributes of the outcome. For example, if we consider a patient, we might consider attributes such as "age,"
"gender," and "smoking history" that are relevant for assigning probability over possible diseases and symptoms. We would like then consider events such as "age > 55, heavy smoking history, and suers from repeated cough." To use a concrete example, consider again a distribution over a population of students in a course. Suppose that we want to reason about the intelligence of students, their final grades, and so forth. We can use an event such as GradeA to denote the subset of students that received the grade A and use it in our formulation. However, this discussion becomes rather cumbersome if we also want to consider students with

grade B, students with grade C, and so on. Instead, we would like to consider a way of directly referring to a student's grade in a clean, mathematical way. The formal machinery for discussing attributes and their values in dierent outcomes are random variable random variables. A random variable is a way of reporting an attribute of the outcome. For example, suppose we have a random variable Grade that reports the final grade of a student, then the statement P (Grade = A) is another notation for P (GradeA).

n the statement P (Grade = A) is another notation for P (GradeA).

1.4.2 What Is a Random Variable?

Formally, a random variable, such as Grade, is defined by a function that associates with each outcome in a value. For example, Grade is defined by a function fGrade that maps each person in to his or her grade (say, one of A, B, or C). The event Grade = A is a shorthand for the event   : fGrade() = A. In our example, we might also have a random variable Intelligence that (for simplicity) takes as values either "high" or "low." In this case, the event "Intelligence = high" refers, as can be expected, to the set of smart (high intelligence) students. Random variables can take dierent sets of values. We can think of categorical (or discrete) random variables that take one of a few values, as in our two preceding examples. We can also talk about random variables that can take infinitely many values (for example, integer or real values), such as Height that denotes a student's height. We use Val(X) to denote the set of values that a random variable X can take. In most of the discussion in this book we examine either categorical random variables or random variables that take real values. We will usually use uppercase roman letters X, Y, Z to denote random variables. In discussing generic random variables, we often use a lowercase letter to refer to a value of a random variable. Thus, we use x to refer to a generic value of X. For example, in statements such as "P (X = x) 0 for all x Val(X)." When we discuss categorical random variables, we use the notation x1, . . . , xk, for k = |Val(X)| (the number of elements in Val(X)), when we need to enumerate the specific values of X, for example, in statements such as kX i =1 P (X = xi) = 1. multinomial The distribution over such a variable is called a multinomial. In the case of a binary-valued distribution random variable X, where Val(X) = false, true, we often use x1 to denote the value true for X, and x0 to denote the value false. The distribution of such a random variable is called a Bernoulli Bernoulli distribution. distribution We also use boldface type to denote sets of random variables. Thus, X, Y , or Z are typically used to denote a set of random variables, while x, y, z denote assignments of values to the
variables in these sets. We extend the definition of Val(X) to refer to sets of variables in the obvious way. Thus, x is always a member of Val(X). For Y X, we use xhY i to refer to the assignment within x to the variables in Y . For two assignments x (to X) and y (to Y), we say that x y if they agree on the variables in their intersection, that is, xhX Y i = yhX Y i. In many cases, the notation P(X = x) is redundant, since the fact that x is a value of X is already reported by our choice of letter. Thus, in many texts on probability, the identity of a random variable is not explicitly mentioned, but can be inferred through the notation used for its value. Thus, we use P(x) as a shorthand for P(X = x) when the identity of the random variable is clear from the context. Another shorthand notation is that Px refers to a sum over all possible values that X can

take. Thus, the preceding statement will often appear as Px P(x) = 1. Finally, another standard notation has to do with conjunction. Rather than write P((X = x) (Y = y)), we write P(X = x, Y = y), or just P(x, y).
3.3 Marginal and Joint Distributions Once we define a random variable X, we can consider the distribution over events that can be marginal described using X. This distribution is often referred to as the marginal distribution over the distribution random variable X. We denote this distribution by P(X). Returning to our population example, consider the random variable Intelligence. The marginal distribution over Intelligence assigns probability to specific events such as P(Intelligence = high) and P(Intelligence = low), as well as to the trivial event P(Intelligence high, low). Note that these probabilities are defined by the probability distribution over the original space. For concreteness, suppose that P(Intelligence = high) = 0.3, P(Intelligence = low) = 0.7. If we consider the random variable Grade, we can also define a marginal distribution. This is a distribution over all events that can be described in terms of the Grade variable. In our example, we have that P(Grade = A) = 0.25, P(Grade = B) = 0.37, and P(Grade = C) = 0.38. It should be fairly obvious that the marginal distribution is a probability distribution satisfying the properties of definition 2.1. In fact, the only change is that we restrict our attention to the subsets of S that can be described with the random variable X. In many situations, we are interested in questions that involve the values of several random variables. For example, we might be interested in the event "Intelligence = high and Grade = A." joint distribution To discuss such events, we need to consider the joint distribution over these two random variables. In general, the joint distribution over a set X = X1, . . . , Xn of random variables is denoted by P(X1, . . . , Xn) and is the distribution that assigns probabilities to events that are specified in terms of these random variables. We use to refer to a full assignment to the variables in X , that is, Val(X). The joint distribution of two random variables has to be consistent with the marginal distribution, in that P(x) = P y P(x, y). This relationship is shown in figure 2.1, where we compute the marginal distribution over Grade by summing the probabilities along each row. Similarly, we find the marginal distribution over Intelligence by summing out along each column. The resulting sums are typically written in the row or column margins, whence the term "marginal distribution." Suppose we have a joint distribution over the variables X = X1, . . . , Xn. The most fine-grained events we can discuss using these variables are ones of the form "X1 = x1 and X2 = x2, . . ., and Xn = xn" for a choice of values x1, . . . , xn for all the variables. Moreover,
Intelligence low high A 0.07 0.18 0.25 Grade B 0.28 0.09 0.37 C 0.35 0.03 0.38 0.7 0.3 1 Figure 2.1 Example of a joint distribution P(Intelligence, Grade): Values of Intelligence (columns) and Grade (rows) with the associated marginal distribution on each variable. any two such events must be either identical or disjoint, since they both assign values to all the variables in X . In addition, any event defined using variables in X must be a union of a set of canonical such events. Thus, we are eectively working in a canonical outcome space: a space where each outcome space outcome corresponds to a joint assignment to X1, . . . , Xn. More precisely, all our probability computations remain the same whether we consider the original outcome space (for example, all students), or the canonical space (for example, all combinations of intelligence and grade). atomic outcome We use to denote these atomic outcomes: those assigning a value to each variable in X . For example, if we let X = Intelligence, Grade,

there are six atomic outcomes, shown in figure 2.1. The figure also shows one possible joint distribution over these six outcomes. Based on this discussion, from now on we will not explicitly specify the set of outcomes and measurable events, and instead implicitly assume the canonical outcome space.
3.4 Conditional Probability The notion of conditional probability extends to induced distributions over random variables. For conditional example, we use the notation P (Intelligence | Grade = A) to denote the conditional distribution distribution over the events describable by Intelligence given the knowledge that the student's grade is A. Note that the conditional distribution over a random variable given an observation of the value of another one is not the same as the marginal distribution. In our example, P (Intelligence = high) = 0.3, and P (Intelligence = high | Grade = A) = 0.18/0.25 = 0.72. Thus, clearly P (Intelligence | Grade = A) is dierent from the marginal distribution P (Intelligence). The latter distribution represents our prior knowledge about students before learning anything else about a particular student, while the conditional distribution represents our more informed distribution after learning her grade. We will often use the notation P (X | Y) to represent a set of conditional probability distributions. Intuitively, for each value of Y , this object assigns a probability over values of X using the conditional probability. This notation allows us to write the shorthand version of the chain rule: P (X, Y) = P (X)P (Y | X), which can be extended to multiple variables as P (X1, . . . , Xk) = P (X1)P (X2 | X1) · · · P (Xk | X1, . . . , Xk1). (2.5) Similarly, we can state Bayes' rule in terms of conditional probability distributions: P (X | Y) = P (X)P (Y | X) P (Y) . (2.6)

1.5 Independence and Conditional Independence

4.1 Independence As we mentioned, we usually expect P(|)P(|) to be different from P()P(). That is, learning that is true changes our probability over . However, in some situations equality can occur, so that P(|)=P()P(|)=P(). That is, learning that occurs did not change our probability of .
Definition independent events
We say that an event is independent of event in PP, denoted P()P(), if P(|)=P()P(|)=P() or if P()=0P()=0.
We can also provide an alternative definition for the concept of independence:
Proposition 2.1
A distribution PP satisfies ()() if and only if P()=P()P()P()=P()P().
PROOF Consider first the case where P()=0P()=0; here, we also have P()=0P()=0, and so the equivalence immediately holds. When P()0P()0, we can use the chain rule; we write P()=P(|)P()P()=P(|)P(). Since is independent of , we have that P(|)=P()P(|)=P(). Thus, P()=P()P()P()=P()P(). Conversely, suppose that P()=P()P()P()=P()P(). Then, by definition, we have that
P(|)=P()P()=P()P()P()=P(). P(|)=P()P()=P()P()P()=P(). As an immediate consequence of this alternative definition, we see that independence is a symmetric notion. That is, ( ) implies ( ). Example 2.3 For example, suppose that we toss two coins, and let be the event "the first toss results in a head" and the event "the second toss results in a head." It is not hard to convince ourselves that we expect that these two events to be independent. Learning that is true would not change our probability of . In this case, we see two dierent physical

processes (that is, coin tosses) leading to the events, which makes it intuitive that the probabilities of the two are independent. In certain cases, the same process can lead to independent events. For example, consider the event denoting "the die outcome is even" and the event denoting "the die outcome is 1 or 2." It is easy to check that if the die is fair (each of the six possible outcomes has probability 1 6), then these two events are independent.
4.2 Conditional Independence While independence is a useful property, it is not often that we encounter two indepen- dent events. A more common situation is when two events are independent given an additional event. For example, suppose we want to reason about the chance that our student is accepted to graduate studies at Stanford or MIT. Denote by Stanford the event "admitted to Stanford" and by MIT the event "admitted to MIT." In most reasonable distributions, these two events are not independent. If we learn that a student was admitted to Stanford, then our estimate of her probability of being accepted at MIT is now higher, since it is a sign that she is a promising student
Now, suppose that both universities base their decisions only on the student's grade point average (GPA), and we know that our student has a GPA of A. In this case, we might argue that learning that the student was admitted to Stanford should not change the probability that she will be admitted to MIT: Her GPA already tells us the information relevant to her chances of admission to MIT, and finding out about her admission to Stanford does not change that. Formally, the statement is P(MIT | Stanford, GradeA) = P(MIT | GradeA). In this case, we say that MIT is conditionally independent of Stanford given GradeA. Definition 2.3 We say that an event is conditionally independent of event given event in P, denoted conditional independence P |= (|), if P(|) = P(|) or if P( ) = 0. It is easy to extend the arguments we have seen in the case of (unconditional) independencies to give an alternative definition. Proposition 2.2 P satisfies (|) if and only if P(|) = P(|)P(|).
4.3 Independence of Random Variables Until now, we have focused on independence between events. Thus, we can say that two events, such as one toss landing heads and a second also landing heads, are independent. However, we would like to say that any pair of outcomes of the coin tosses is independent. To capture such statements, we can examine the generalization of independence to sets of random variables. Definition 2.4 Let X, Y , Z be sets of random variables. We say that X is conditionally independent of Y given conditional independence Z in a distribution P if P satisfies (X = x Y = y | Z = z) for all values x Val(X), y Val(Y), and z Val(Z). The variables in the set Z are often said to be observed. If the set observed variable Z is empty, then instead of writing (X Y |), we write (X Y) and say that X and Y are marginally independent. marginal independence Thus, an independence statement over random variables is a universal quantification over all possible values of the random variables. The alternative characterization of conditional independence follows immediately: Proposition 2.3 The distribution P satisfies (X Y | Z) if and only if P(X, Y | Z) = P(X | Z)P(Y | Z). Suppose we learn about a conditional independence. Can we conclude other independence properties that must hold in the distribution? We have already seen one such example: symmetry • Symmetry: (X Y | Z) = (Y X | Z). (2.7) There are several other properties that hold for conditional independence, and that often provide a very clean method for proving important properties about distributions. Some key properties are:
• Decomposition: (X Y , W | Z) = (X Y | Z). (2.8) weak union • Weak union:

(X Y , W | Z) = (X Y | Z, W). (2.9) contraction • Contraction: (X W | Z, Y)(X Y | Z) = (X Y , W | Z). (2.10) An additional important property does not hold in general, but it does hold in an important subclass of distributions. Definition 2.5 A distribution P is said to be positive if for all events S such that 6= , we have that positive distribution P() > 0. For positive distributions, we also have the following property: intersection • Intersection: For positive distributions, and for mutually disjoint sets X, Y , Z, W : (X Y | Z, W)(X W | Z, Y) = (X Y , W | Z). (2.11) The proof of these properties is not dicult. For example, to prove Decomposition, assume that (X Y, W | Z) holds. Then, from the definition of conditional independence, we have that P(X, Y, W | Z) = P(X | Z)P(Y, W | Z). Now, using basic rules of probability and arithmetic, we can show P(X, Y | Z) = X w P(X, Y, w | Z) = X w P(X | Z)P(Y, w | Z) = P(X | Z) X w P(Y, w | Z) = P(X | Z)P(Y | Z). The only property we used here is called "reasoning by cases" (see exercise 2.6). We conclude that (X Y | Z).

1.6 Querying a Distribution

Our focus throughout this book is on using a joint probability distribution over multiple random variables to answer queries of interest.

1.6.1 5.1 Probability Queries

probability query Perhaps the most common query type is the probability query. Such a query consists of two parts: evidence • The evidence: a subset E of random variables in the model, and an instantiation e to these variables; query variables • the query variables: a subset Y of random variables in the network. Our task is to compute P(Y | E = e), posterior that is, the posterior probability distribution over the values y of Y , conditioned on the fact that distribution E = e. This expression can also be viewed as the marginal over Y , in the distribution we obtain by conditioning on e.

1.6.2 5.2 MAP Queries

A second important type of task is that of finding a high-probability joint assignment to some subset of variables. The simplest variant of this type of task is the MAP query (also called MAP assignment most probable explanation (MPE)), whose aim is to find the MAP assignment — the most likely assignment to all of the (non-evidence) variables. More precisely, if we let W = X E, our task is to find the most likely assignment to the variables in W given the evidence E = e: MAP(W | e) = argmax w P(w, e), (2.12) where, in general, argmaxx f(x) represents the value of x for which f(x) is maximal. Note that there might be more than one assignment that has the highest posterior probability. In this case, we can either decide that the MAP task is to return the set of possible assignments, or to return an arbitrary member of that set. It is important to understand the dierence between MAP queries and probability queries. In a MAP query, we are finding the most likely joint assignment to W . To find the most likely assignment to a single variable A, we could simply compute P(A | e) and then pick the most likely value. However, the assignment where each variable individually picks its most likely value can be quite dierent from the most likely joint assignment to all variables simultaneously. This phenomenon can occur even in the simplest case, where we have no evidence. Example 2.4 Consider a two node chain A B where A and B are both binary-valued. Assume that: a0 a1 0.4 0.6 A b0 b1 a0 0.1 0.9 a1 0.5 0.5 (2.13) We can see that P(a1) > P(a0), so that MAP(A) = a1. However, MAP(A, B) = (a0, b1): Both values of B have the same probability given a1. Thus, the most likely assignment containing a1 has probability 0.6 0.5 = 0.3. On the other hand, the distribution over values of B is more skewed given a0, and the most likely assignment (a0, b1) has the probability 0.4 0.9 = 0.36. Thus, we have that argmaxa,b P(a, b) 6= (argmaxa P(a),argmaxb P(b)).

1.6.3 5.3 Marginal MAP Queries

To motivate our second query type, let us return to the phenomenon demonstrated in example 2.4. Now, consider a medical diagnosis problem, where the most likely disease has multiple possible symptoms, each of which occurs with some probability, but not an overwhelming probability. On the other hand, a somewhat rarer disease might have only a few symptoms, each of which is very likely given the disease. As in our simple example, the MAP assignment to the data and the symptoms might be higher for the second disease than for the first one. The solution here is to look for the most likely assignment to the disease variable(s) only, rather than the most likely assignment to both the disease and symptom variables. This approach suggests marginal MAP the use of a more general query type. In the marginal MAP query, we have a subset of variables Y that forms our query. The task is to find the most likely assignment to the variables in Y given the evidence E = e: MAP(Y | e) = arg max y P(y | e). If we let Z = X Y E, the marginal MAP task is to compute: MAP(Y | e) = arg max Y X Z P(Y , Z | e). Thus, marginal MAP queries contain both summations and maximizations; in a way, it contains elements of both a conditional probability query and a MAP query. Note that example 2.4 shows that marginal MAP assignments are not monotonic: the most likely assignment MAP(Y1 | e) might be completely dierent from the assignment to Y1 in MAP(Y1, Y2 | e). Thus, in particular, we cannot use a MAP query to give us the correct answer to a marginal MAP query.

1.7 Continuous Spaces

In the previous section, we focused on random variables that have a finite set of possible values. In many situations, we also want to reason about continuous quantities such as weight, height, duration, or cost that take real numbers in IR. When dealing with probabilities over continuous random variables, we have to deal with some technical issues. For example, suppose that we want to reason about a random variable X that can take values in the range between 0 and 1. That is, Val(X) is the interval [0, 1]. Moreover, assume that we want to assign each number in this range equal probability. What would be the probability of a number x? Clearly, since each x has the same probability, and there are infinite number of values, we must have that P(X = x) = 0. This problem appears even if we do not require uniform probability.

6.1 Probability Density Functions How do we define probability over a continuous random variable? We say that a function density function p : IR 7 IR is a probability density function or (PDF) for X if it is a nonnegative integrable function such that Z Val(X) p(x)dx = 1. That is, the integral over the set of possible values of X is 1. The PDF defines a distribution for X as follows: for any x in our event space: P(X a) = aZ p(x)dx. cumulative The function P is the cumulative distribution for X. We can easily employ the rules of distribution probability to see that by using the density function we can evaluate the probability of other events. For example, P(a X b) = bZa p(x)dx. Intuitively, the value of a PDF p(x) at a point x is the incremental amount that x adds to the cumulative distribution in the integration process. The higher the value of p at and around x, the more mass is added to the cumulative distribution as it passes x. The simplest PDF is the uniform distribution. Definition 2.6 A variable X has a uniform distribution over [a, b], denoted X Unif[a,b] if it has the PDF uniform distribution p(x) = 0 otherwise b 1a b x a. Thus, the probability of any subinterval of [a, b] is proportional its size relative to the size of [a, b]. Note that, if b a < 1, then the density can be greater than 1. Although this looks unintuitive, this situation can occur even in a legal PDF, if the interval over which the value is greater than 1 is not too large. We have only to satisfy the constraint that the total area under the PDF is 1. As a more complex example, consider the Gaussian distribution. Definition 2.7 A random variable X has a Gaussian distribution with mean and variance 2, denoted X Gaussian distribution N ; 2, if it has the PDF p(x) = 21 e (x2 2)2 . standard A standard Gaussian is one with mean 0 and variance 1. Gaussian A Gaussian distribution has a bell-like curve, where the mean parameter controls the location of the peak, that is, the value for which the Gaussian gets its maximum value. The variance parameter 2 determines how peaked the Gaussian is: the smaller the variance, the

more peaked the Gaussian. Figure 2.2 shows the probability density function of a few dierent Gaussian distributions. More technically, the probability density function is specified as an exponential, where the expression in the exponent corresponds to the square of the number of standard deviations that x is away from the mean . The probability of x decreases exponentially with the square of its deviation from the mean, as measured in units of its standard deviation.

6.2 Joint Density Functions

The discussion of density functions for a single variable naturally extends for joint distributions of continuous random variables. Definition 2.8 Let P be a joint distribution over continuous random variables X1, . . . , Xn. A function p(x1, . . . , xn) joint density is a joint density function of X1, . . . , Xn if • p(x1, . . . , xn) 0 for all values x1, . . . , xn of X1, . . . , Xn. • p is an integrable function. • For any choice of a1, . . . , an, and b1, . . . , bn, P(a1 X1 b1, . . . , an Xn bn) = b1 Za1 · · · b nZ a n p(x1, . . . , xn)dx1 . . . dxn. Thus, a joint density specifies the probability of any joint event over the variables of interest. Both the uniform distribution and the Gaussian distribution have natural extensions to the multivariate case. The definition of a multivariate uniform distribution is straightforward. We defer the definition of the multivariate Gaussian to section 7.1. From the joint density we can derive the marginal density of any random variable by integrating out the other variables. Thus, for example, if p(x, y) is the joint density of X and Y

then p(x) = Z p(x, y)dy. To see why this equality holds, note that the event

a X b is, by definition, equal to the event "a X b and Y ." This rule is the direct analogue of marginalization for discrete variables. Note that, as with discrete probability distributions, we abuse notation a bit and use p to denote both the joint density of X and Y and the marginal density of X. In cases where the distinction is not clear, we use subscripts, so that pX will be the marginal density, of X, and pX,Y the joint density.
6.3 Conditional Density Functions As with discrete random variables, we want to be able to describe conditional distributions of continuous variables. Suppose, for example, we want to define P(Y | X = x). Applying the definition of conditional distribution (equation (2.1)), we run into a problem, since P(X = x) = 0. Thus, the ratio of P(Y, X = x) and P(X = x) is undefined. To avoid this problem, we might consider conditioning on the event x X x + , which can have a positive probability. Now, the conditional probability is well defined. Thus, we might consider the limit of this quantity when 0. We define P(Y | x) = lim 0 P(Y | x X x +). When does this limit exist? If there is a continuous joint density function p(x, y), then we can derive the form for this term. To do so, consider some event on Y , say a Y b. Recall that P(a Y b | x X x +) = P(a Y b, x X x +) P(x X x +) = Ra b Rx x + p(x0, y)dydx0 Rx x + p(x0)dx0 . When is suciently small, we can approximate x+ Z x p(x0)dx0 2p(x). Using a similar approximation for p(x0, y), we get P(a Y b | x X x +) Ra b 2p(x, y)dy 2p(x) = bZa p(x, y) p(x) dy. We conclude that p(x,y) p(x) is the density of P(Y | X = x).
Let p(x, y) be the joint density of X and Y . The conditional density function of Y given X is conditional density function defined as p(y | x) = p(x, y) p(x) When p(x) = 0, the conditional density is undefined. The conditional density p(y | x) characterizes the conditional distribution P(Y | X = x) we defined earlier. The properties of joint distributions and conditional distributions carry over to joint and conditional density functions. In particular, we have the chain rule p(x, y) = p(x)p(y | x) (2.14) and Bayes' rule p(x | y) = p(x)p(y | x) p(y) . (2.15) As a general statement, whenever we discuss joint distributions of continuous random variables, we discuss properties with respect to the joint density function instead of the joint distribution, as we do in the case of discrete variables. Of particular interest is the notion of (conditional) independence of continuous random variables. Definition 2.10 Let X, Y , and Z be sets of continuous random variables with joint density p(X, Y , Z). We say conditional that X is conditionally independent of Y given Z if independence p(x | z) = p(x | y, z) for all x, y, z such that p(z) > 0.

1.8 Kì vọng và phương sai

1.8.1 Kì vọng

kì vọng

Cho X là một biến rời rạc nhận các giá trị số, khi đó kì vọng của X dưới phân phối P là

$$\mathbb{E}_P[X] = \sum_x x \cdot P(x).$$

Nếu X là một biến liên tục, khi đó sử dụng hàm mật độ

$$\mathbb{E}_P[X] = \int x \cdot p(x).$$

Ví dụ, nếu X là số chấm hiện ra khi tung một xúc sắc với xác suất xuất hiện mỗi mặt là $\frac{1}{6}$. Khi đó, kì vọng của $\mathbb{E}[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \cdots + 6 \cdot \frac{1}{6} = 3.5$. Nếu trong trường hợp xúc sắc không cân bằng $P(X = 6) = 0.5$ và $P(X = x) = 0.1)$ với $x < 6$, khi đó, $\mathbb{E}[X] = 1 \cdot 0.1 + 2 \cdot 0.1 + \cdots + 5 \cdot 0.1 + 6 \cdots 0.5 = 4.5$.

Often we are interested in expectations of a function of a random variable (or several random variables). Thus, we might consider extending the definition to consider the expectation of a functional term such as X2 + 0.5X. Note, however, that any function g of a set of random variables X1, . . . , Xk is essentially defining a new random variable Y : For any outcome , we define the value of Y as g(fX1(), . . . , fXk()). Based on this discussion, we often define new random variables by a functional term. For example Y = X2, or Y = eX. We can also consider functions that map values of one or more categorical random variables to numerical values. One such function that we use quite often is indicator function the indicator function, which we denote 11X = x. This function takes value 1 when X = x, and 0 otherwise. In addition, we often consider expectations of functions of random variables without bothering to name the random variables they define. For example IEP [X + Y]. Nonetheless, we should keep in mind that such a term does refer to an expectation of a random variable. We now turn to examine properties of the expectation of a random variable. First, as can be easily seen, the expectation of a random variable is a linear function in that random variable. Thus, IE[a · X + b] = aIE[X] + b. A more complex situation is when we consider the expectation of a function of several random variables that have some joint behavior. An important property of expectation is that the expectation of a sum of two random variables is the sum of the expectations. Proposition 2.4 IE[X + Y] = IE[X] + IE[Y]. linearity of This property is called linearity of expectation. It is important to stress that this identity is true expectation even when the variables are not independent. As we will see, this property is key in simplifying many seemingly complex problems. Finally, what can we say about the expectation of a product of two random variables? In general, very little: Example 2.5 Consider two random variables X and Y , each of which takes the value +1 with probability 1/2, and the value 1 with probability 1/2. If X and Y are independent, then IE[X · Y] = 0. On the other hand, if X and Y are correlated in that they always take the same value, then IE[X · Y] = 1. However, when X and Y are independent, then, as in our example, we can compute the expectation simply as a product of their individual expectations: Proposition 2.5 If X and Y are independent, then IE[X · Y] = IE[X] · IE[Y]. conditional We often also use the expectation given some evidence. The conditional expectation of X expectation given y is IEP [X | y] = X x x · P(x | y).

1.8.2 Phương sai

phương sai

Kì vọng của X chỉ giá trị trung bình của X. Tuy nhiên, nó không chỉ sự khác nhau giữa các giá trị mà X có thể nhận.

$$\mathbb{V}ar_P[X] = \mathbb{E}_P\left[(X - \mathbb{E}_P[X])^2\right].$$

A measure of this deviation is the variance of X. VVarP [X] = IEP h(X IEP

[X])2i. Thus, the variance is the expectation of the squared dierence between X and its expected value. It gives us an indication of the spread of values of X around the expected value. An alternative formulation of the variance is VVar[X] = IEX2 (IE[X])2 . (2.16) (see exercise 2.11). Similar to the expectation, we can consider the expectation of a functions of random variables. Proposition 2.6 If X and Y are independent, then VVar[X + Y] = VVar[X] + VVar[Y]. It is straightforward to show that the variance scales as a quadratic function of X. In particular, we have: VVar[a · X + b] = a2VVar[X]. For this reason, we are often interested in the square root of the variance, which is called the standard standard deviation of the random variable. We define deviation X = pVVar[X]. The intuition is that it is improbable to encounter values of X that are farther than several standard deviations from the expected value of X. Thus, X is a normalized measure of "distance" from the expected value of X. As an example consider the Gaussian distribution of definition 2.7. Proposition 2.7 Let X be a random variable with Gaussian distribution N(, 2), then IE[X] = and VVar[X] = 2. Thus, the parameters of the Gaussian distribution specify the expectation and the variance of the distribution. As we can see from the form of the distribution, the density of values of X drops exponentially fast in the distance x . Not all distributions show such a rapid decline in the probability of outcomes that are distant from the expectation. However, even for arbitrary distributions, one can show that there is a decline. Theorem 2.1 (Chebyshev inequality): Chebyshev's inequality P (|X IEP [X]| t) VVarP [X] t2 .
We can restate this inequality in terms of standard deviations: We write t = kX to get P(|X IEP [X]| kX) 1 k2. Thus, for example, the probability of X being more than two standard deviations away from IE[X] is less than 1/4.

1.9 Các hàm phân phối thông dụng

Phần này có thêm khảo Goodfellow u.a. (2016) và giáo trình xác suất thống kê của thạc sỹ Trần Thiện Khải, đại học Trà Vinh [1]

17/01/2018 Lòng vòng thế nào hôm nay lại tìm được blog của bạn Đỗ Minh Hải, rất hay

1.9.1 Biến rời rạc

Tổng kết các phân phối rời rạc tham khảo slide 3, chapter 9 của khóa CS 109 [2]
Phân phối Bernoulli:

- xác suất xuất hiện mặt ngửa $X \sim Ber(p)$

Phân phối Bernoulli:

- xác suất xuất hiện mặt ngửa $X \sim Ber(p)$

Phân phối Binomial:

- xác suất thành công n lần $X \sim B(n, p)$

Phân phối Poisson:

[1] http://www.ctec.tvu.edu.vn/ttkhai/xacsuatthongke_dh.htm
[2] Slide 3, chapter 9, CS109 Stanford

- xác suất thành công n lần $X \sim Poi(\lambda)$

Phân phối Geometric:

- số lần thử đến khi thành công $X \sim Geo(p)$

Phân phối Negative Binomial:

- số lần thử đến khi r thành công $X \sim NegBin(r, p)$

Phân phối Hyper Geometric:

- số bóng trắng lấy được trong N bóng chứa m bóng trắng $X \sim HypG(n, N, m)$

1.9.2 Biến ngẫu nhiên Bernoulli

Phép thử Bernoulli là một phép thử chỉ có hai khả năng xảy ra "thành công" hoặc "thất bại". Trong thực tế, các ví dụ của phép thử Bernoulli xuất hiện rất phổ biến như *tung đồng xu*, *sự kiện một ổ đĩa bị hỏng*, *sự kiện một ai đó thích xem một bộ phim trên Netflix*. iên Định nghĩa X là một biến ngẫu nhiên, nhận giá trị bằng 1 nếu sự kiên thành công, nhận giá trị bằng 0 nếu sự kiện thất bại. Xác suất sự kiện thành công (X nhận giá trị bằng 1) là $P(X = 1) = p(1) = p$, xác suất sự kiên thất bại $P(X = 0) = p(0) = 1 - p$
X là biến ngẫu nhiên Bernoulli

$$X \sim Ber(p)$$

Khi đó, kì vọng của X

$$E[X] = p$$

Phương sai của X

$$Var(X) = p(1 - p)$$

1.9.3 Phân phối Bernoulli

Như đã đề cập về phép thử Béc-nu-li rằng mọi phép thử của nó chỉ cho 2 kết quả duy nhất là A với xác suất p và $\bar{A}$ với xác suất $q = 1 - p$ Biến ngẫu nhiên X tuân theo phân phối Béc-nu-li

$$X \sim B(p)$$

với tham số $p \in \mathbb{R}, 0 \le p \le 1$ là xác suất xuất hiện của A tại mỗi phép thử

Định nghĩa		Giá trị
PMF	p(x)	$p(x) \mid p^x(1-p)^{1-x}, x \in \{0, 1\}$
CDF	$F(x; p)$	$\begin{cases} 0 & \text{for } x < 0 \\ 1 - p & \text{for } 0 \le x < 1 \\ 1 & \text{for } x \ge 1 \end{cases}$
Kỳ vọng	$E[X]$	p
Phương sai	$Var(X)$	$p(1-p)$

1.9.4 Phân phối Binomial

- xác suất thành công n lần $X \sim B(n, p)$

1.9.5 Phân phối Poisson

- xác suất thành công n lần $X \sim Poi(\lambda)$

1.9.6 Phân phối Geometric

- số lần thử đến khi thành công $X \sim Geo(p)$

1.9.7 Phân phối Negative Binomial

- số lần thử đến khi r thành công $X \sim NegBin(r, p)$

1.9.8 Phân phối Hyper Geometric

- số bóng trắng lấy được trong N bóng chứa m bóng trắng $X \sim HypG(n, N, m)$

1.9.9 Biến liên tục

1.9.10 Phân phối đều

phân phối đều — Là phân phối mà xác suất xuất hiện của các sự kiện là như nhau.

Biến ngẫu nhiên X tuân theo phân phối đều rời rạc

$$X \sim Unif(a, b)$$

với tham số $a, b \in \mathbb{Z}; a < b$ là khoảng giá trị của X, đặt $n = b - a + 1$

Ta sẽ có:

Định nghĩa	Giá trị
PMF	$p(x) \mid \frac{1}{n}, \forall x \in [a, b]$
CDF - $F(x; a, b)$	$\frac{x - a + 1}{n}, \forall x \in [a, b]$
Kỳ vọng - $E[X]$	$\frac{a + b}{2}$
Phương sai - $Var(X)$	$\frac{n^2 - 1}{12}$

Ví dụ 1:

bài toán chờ xe buýt — Lịch chạy của xe buýt tại một trạm xe buýt như sau: chiếc xe buýt đầu tiên trong ngày sẽ khởi hành từ trạm này vào lúc 7 giờ, cứ sau mỗi 15 phút sẽ có một xe khác đến trạm. Giả sử một hành khách đến trạm trong khoảng thời gian từ 7 giờ đến 7 giờ 30. Tìm xác suất để hành khách này chờ:

a) Ít hơn 5 phút.

b) Ít nhất 12 phút.

Giải

Gọi X là số phút sau 7 giờ mà hành khách đến trạm.
Ta có: $X \sim R[0; 30]$.
a) Hành khách sẽ chờ ít hơn 5 phút nếu đến trạm giữa 7 giờ 10 và 7 giờ 15 hoặc giữa 7 giờ 25 và 7 giờ 30. Do đó xác suất cần tìm là:

$$P(0 < X < 15) + P(25 < X < 30) = \frac{5}{30} + \frac{5}{30} = \frac{1}{3}$$

b) Hành khách chờ ít nhất 12 phút nếu đến trạm giữa 7 giờ và 7 giờ 3 phút hoặc giữa 7 giờ 15 phút và 7 giờ 18 phút. Xác suất cần tìm là:

$$P(0 < X < 3) + P(15 < X < 18) = \frac{3}{30} + \frac{3}{30} = \frac{1}{5}$$

Chương 2

Thống kê

Thống kê làm việc với dữ liệu trong thực tế.

Phần II

Linh tinh

Chương 3

Nghiên cứu

01/11/2017 Không biết mình có phải làm nghiên cứu không nữa? Vừa kiêm phát triển, vừa đọc paper mỗi ngày. Thôi, cứ (miễn cưỡng) cho là nghiên cứu viên đi.

3.1 Các công cụ

3.1.1 Google Scholar & Semantic Scholar

Google Scholar vẫn là lựa chọn tốt

- Tìm kiếm tác giả theo lĩnh vực nghiên cứu và quốc gia: sử dụng filter label: + đuôi
  - ví dụ: danh sách các nhà nghiên cứu Việt Nam thuộc lĩnh vực xử lý ngôn ngữ tự nhiên
- danh sách này đã sắp xếp theo lượng trích dẫn

Bên cạnh đó còn có semanticscholar (một project của allenai) với các killer features

- Tìm kiếm các bài báo khoa học với từ khóa và filter theo năm, tên hội nghị
- Xem những người ảnh hưởng, ảnh hưởng bởi một nhà nghiên cứu, cũng như xem co-author, journals và conferences mà một nhà nghiên cứu hay gửi bài

3.1.2 Mendeley

Mendeley rất tốt cho việc quản lý và lưu trữ. Tuy nhiên điểm hạn chế lại là không lưu thông tin về citation

3.1.3 Hội nghị và tạp chí

Các hội nghị tốt về xử lý ngôn ngữ tự nhiên

- Rank A: ACL, EACL, NAACL, EMNLP, CoNLL

- Rank B: SemEval

Các tạp chí

- Computational Linguistics (CL)

3.1.4 Câu chuyện của Scihub

Sci-Hub được tạo ra vào ngày 5 tháng 9 năm 2011, do nhà nghiên cứu đến từ Kazakhstan, Alexandra Elbakyan
Hãy nghe chia sẻ của cô về sự ra đời của Sci-Hub
> Khi tôi còn là một sinh viên tại Đại học Kazakhstan, tôi không có quyền truy cập vào bất kỳ tài liệu nghiên cứu. Những bai bao tôi cần cho dự án nghiên cứu của tôi. Thanh toán 32 USD thi thât la điên rô khi bạn cần phải đọc lướt hoặc đọc hàng chục hoặc hàng trăm tơ để làm nghiên cứu. Tôi có được những bai bao nhơ vao trôm chúng. Sau đó tôi thấy có rất nhiều và rất nhiều nhà nghiên cứu (thậm chí không phải sinh viên, nhưng các nhà nghiên cứu trường đại học) giống như tôi, đặc biệt là ở các nước đang phát triển. Họ đã tạo ra các cộng đồng trực tuyến (diễn đàn) để giải quyết vấn đề này. Tôi là một thành viên tích cực trong một cộng đồng như vậy ở Nga. Ở đây ai cần có một bài nghiên cứu, nhưng không thể trả tiền cho nó, có thể đặt một yêu cầu và các thành viên khác, những người có thể có được những giấy sẽ gửi nó cho miễn phí qua email. Tôi có thể lấy bất cứ bai nao, vì vậy tôi đã giải quyết nhiều yêu cầu và người ta luôn rất biết ơn sự giúp đỡ của tôi. Sau đó, tôi tạo Sci-Hub.org, một trang web mà chỉ đơn giản là làm cho quá trình này tự động và các trang web ngay lập tức đã trở thành phổ biến.
Về phần mình, là một nhà nghiên cứu trẻ, đương nhiên phải đọc liên tục. Các báo cáo ở Việt Nam về xử lý ngôn ngữ tự nhiên thì thường không tải lên các trang mở như arxiv.org, các kỷ yếu hội nghị cũng không public các proceedings. Thật sự scihub đã giúp mình rất nhiều.
Scihub bị chặn
Vào thời điểm này (12/2017), scihub bị chặn quyết liệt. Hóng được trên page facebook của scihub các cách truy cập scihub. Đã thử các domain khác như .tw, .hk. Mọi chuyện vẫn ổn cho đến hôm nay (21/12/2017), không thể truy cập vào nữa.
Đành phải cài tor để truy cập vào scihub ở địa chỉ http://scihub22266oqcxt.onion. Và mọi chuyện lại ổn.

3.2 Làm sao để nghiên cứu tốt

- Làm việc mỗi ngày
- Cập nhật các kết quả từ các hội nghị, tạp chí
- Viết nhật ký nghiên cứu mỗi tuần (tổng kết công việc tuần trước, các ý tưởng mới, kế hoạch tuần này)

3.3 Sách giáo khoa

Machine Learning Yearning, by Andrew Ng

3.4 Lượm lặt

Review các khóa học Deep Learning

3.5 Thuyết trình

Tự nhiên hôm nay (22/01/2018) lại đọc được bài You suck at PowerPoint, thấy hay quá.
Sau đây là 10 lỗi thường gặp khi làm bài thuyết trình

1. Quá nhiều chữ trong 1 slide.
2. Màu chữ và màu nền không tương phản với nhau.
3. Dùng clip art, word art.
4. Hình ảnh sử dụng trong slide chất lượng kém, scale sai tỉ lệ.
5. Sử dụng nhiều font chữ trong 1 slide.
6. Lạm dụng quá nhiều hiệu ứng (animation/transition).
7. Bài presentation không có cấu trúc.
8. Slide không ăn nhập gì với nội dung trình bày.
9. Không ghi rõ nguồn khi sử dụng tài liệu, hình ảnh của người khác.
10. Ý thức của người làm slide

Chương 4

Nghề lập trình

Chân kinh con đường lập trình: [Teach Yourself Programming in Ten Years. Peter Norvig](http://norvig.com/21-days.html)
Trang web hữu ích
* Chia sẻ thú vị: [15 năm lập trình ở Việt Nam](https://vozforums.com/showthread.php?t=3431312) của Blanic (vozfourm) * Trang web chứa cheatsheet so sánh các ngôn ngữ lập trình và công nghệ http://hyperpolyglot.org/
01/11/2017
Vậy là đã vào nghề (đi làm full time trả lương) được 3 năm rưỡi rồi. Thời gian trôi qua nhanh như *ó chạy ngoài đồng thật. Tâm đắc nhất với câu trong một quyển gì đó của anh lead HR google. Có 4 level của nghề nghiệp. 1 là thỏa mãn được yêu cầu cả bản. 2 là dự đoán được tương lai. 3 là cá nhân hóa (ý nói là tận tình với các khách hàng). 4 là phiêu diêu tự tại. Hay thật! Bao giờ mới được vào mức 4 đây.

Chương 5

Latex

15/12/2017:
Hôm nay tự nhiên nổi hứng vẽ hình trên latex. Thấy blog này là một guide line khá tốt về viết blog phần mềm. Quyết định cài latex
Theo [hướng dẫn này](http://milq.github.io/install-latex-ubuntu-debian/)
"' sudo apt-get install texlive-full sudo apt-get install texmaker "'
Tìm được ngay bên này https://www.overleaf.com/ có vẻ rất hay luôn
Hướng dẫn cực kì cơ bản http://www.math.uni-leipzig.de/ hellmund/LaTeX/pgf-tut.pdf
Chương trình đầu tiên, vẽ diagram cho LanguageFlow

```
\documentclass[border=10pt]{standalone}
\usepackage{verbatim}
\begin{comment}
\end{comment}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}
    \node[draw] (model) at (0, 0) {Model Folder};
    \node[draw] (analyze) at (6, 0) {Analyze Folder};
    \node[draw] (board) at (3,2) {Board};
    \node[draw] (logger) at (3, -2) {Logger};

    \path[->, densely dotted] (board.east)
        edge [out=0, in=90]
        node[ fill =white, pos=.5] {\tiny (1) init }
        (analyze.north) ;
    \path[->, densely dotted] (board.south)
        edge [out=-90, in=180]
        node[ fill =white, pos=.3] {\tiny (2) serve}
        (analyze.west) ;
        \path[->, densely dotted] (logger.west)
        edge [out=180, in=-90]
        node[ fill =white, pos=.7] {\tiny (1) read}
        (model.south) ;
        \path[->, densely dotted] (logger.east)
```

```
            edge [out=0, in=−90]
            node[ fill =white, pos=.7] {\tiny (2) write}
            (analyze.south) ;
    \end{tikzpicture}
    \end{document}
```

Doc! Doc! Doc! https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ

Chương 6

Chào hàng

16/01/2018 Bố khỉ. Hôm nay gửi lời mời kết bạn đến một thằng làm research về speech mà nó "chửi" mình không biết pitch. Tổ sư. Tuy nhiên, nó cũng dạy mình một bài học hay về pitch.

Chửi nó là vậy nhưng lần sau sẽ phải đầu tư nhiều hơn cho các lời pitch.

Vẫn không ưa Huyền Chíp như ngày nào, nhưng [bài này](https://www.facebook.com/notes/huyen-chip/k

Tóm lại skill này có 4 phần

1. Ngôn ngữ không trau chuốt 2. Giới thiệu bản thân không tốt 3. Không chỉ ra cho người nhận rằng họ sẽ được gì 4. Không có phương án hành động

Đối với email, thì cần triển khai thế này

* [Chào hỏi] * [Giới thiệu bản thân một cách nào đó để người đọc quan tâm đến bạn] * [Giải thích lý do bạn biết đến người này và bạn ấn tượng thế nào với họ – ai cũng thích được nghe khen] * [Bạn muốn gì từ người đó và họ sẽ được gì từ việc này] * [Kết thúc]

Chương 7

Phát triển phần mềm

* Phát triển phần mềm là một việc đau khổ. Từ việc quản lý code và version, packing, documentation. Dưới đây là lượm lặt những nguyên tắc cơ bản của mình.
Quản lý phiên bản
Việc đánh số phiên bản các thay đổi của phần mềm khi có hàm được thêm, lỗi được sửa, hay các phiên bản tiền phát hành cần thống nhất theo chuẩn của [semversion]. Điều này giúp nhóm có thể tương tác dễ hơn với người dùng cuối.
![](https://raw.githubusercontent.com/magizbox/magizbox/master/wordpress/phat$_t rien_p han_m em/version.$
Đánh số phiên bản
Phiên bản được đánh theo chuẩn của [semversion](https://semver.org/).
* Mỗi khi một bug được sửa, phiên bản sẽ tăng lên một patch * Mỗi khi có một hàm mới được thêm, phiên bản sẽ tăng lên một patch. * Khi một phiên bản mới được phát hành, phiên bản sẽ tăng lên một minor. * Trước khi phát hành, bắt đầu với x.y.z-rc, x.y.z-rc.1, x.y.z-rc.2. Cuối cùng mới là x.y.z * Mỗi khi phiên bản rc lỗi, khi public lại, đặt phiên bản alpha x.y.z-alpha.t (một phương án tốt hơn là cài đặt thông qua github)
Đánh số phiên bản trên git

Ở nhánh develop, mỗi lần merge sẽ được đánh version theo PATCH, thể hiện một bug được sửa hoặc một thay đổi của hàm

Ở nhánh master, mỗi lần release sẽ được thêm các chỉ như x.y1.0-rc, x.y1.0-rc.1, x.y1.0-rc, x.y1.0
Vẫn còn lăn tăn:
* Hiện tại theo workflow này thì chưa cần sử dụng alpha, beta (chắc là khi đó đã có lượt người sử dụng mới cần đến những phiên bản như thế này)
Tải phần mềm lên pypi
Làm theo hướng dẫn [tại đây](http://peterdowns.com/posts/first-time-with-pypi.html)
1. Cấu hình file ‘.pypirc‘ 2. Upload lên pypi
“‘ python setup.py sdist upload -r pypi “‘

Chương 8

Phương pháp làm việc

Xây dựng phương pháp làm việc là một điều không đơn giản. Với kinh nghiệm 3 năm làm việc, trải qua 2 project. Mà vẫn chưa produce được sản phẩm cho khách hàng. Thiết nghĩ mình nên viết phương pháp làm việc ra để xem xét lại. Có lẽ sẽ có ích cho mọi người.
Làm sao để làm việc hiệu quả, hay xây dựng phương pháp làm việc hữu ích? Câu trả lời ngắn gọn là "Một công cụ không bao giờ đủ".
<!–more–>
Nội dung
1. [Làm sao để đánh giá công việc trong khoảng thời gian dài hạn?](section1) 2. [Làm sao để quản lý project?](section2) 3. [Làm sao để công việc trôi chảy?](section3) 4. [Làm sao để xem xét lại quá trình làm việc?](section4)
<p id="section1">nbsp;</p>
Làm sao để đánh giá công việc trong khoảng thời gian dài hạn?
Câu trả lời OKR (Objectives and Key Results)
`![](https://image.slidesharecdn.com/20170829-ale2017-okr-170829060043/95/agile-leadership-and-goal-management-with-objectives-key-results-okrs-ale-2017-prague-23-638.jpg?cb=1503986541)` *OKR Framework*
Đầu mỗi quý , nên dành vài ngày cho việc xây dựng mục tiêu và những kết quả quan trọng cho quý tới. Cũng như review lại kết quả quý trước.
Bước 1: Xây dựng mục tiêu cá nhân (Objectives)
Bước 2: Xây dựng các Key Results cho mục tiêu này
Bước 3: Lên kế hoạch để hiện thực hóa các Key Results
<p id="section2">nbsp;</p>
Làm sao để quản lý một project
Meistertask
`![](https://focus.meisterlabs.com/wp-content/uploads/2015/03/Projectboard`$_{M}T_{E}N-1.png) * MeisterTask*$
<p id="section3">nbsp;</p>
Làm sao để công việc trôi chảy?
Có vẻ trello là công cụ thích hợp
Bước 1: Tạo một team với một cái tên thật ấn tượng (của mình là Strong Coder)
Trong phần Description của team, nên viết Objectives and Key Results của quý này
Sau đây là một ví dụ
"‘ Objectives and Key Results

-> Build Vietnamese Sentiment Analysis -> Develop underthesea -> Deep Learning Book "'
Bước 2: Đầu mỗi tuần, tạo một board với tên là thời gian ứng với tuần đó (của mình là '2017 | Fight 02 (11/12 - 16/12)')
Board này sẽ gồm 5 mục: "TODO", "PROGRESSING", "Early Fight", "Late Fight", "HABBIT", được lấy cảm hứng từ Kanban Board
* TrelloBoardexample*$
* Mỗi khi không có việc gì làm, xem xét card trong "TODO" * [FOCUS] tập trung làm việc trong "PROGRESSING" * Xem xét lại thói quen làm việc với "HABBIT"
Một Card cho Trello cần có
* Tên công việc (title) * Độ quan trọng (thể hiện ở label xanh (chưa quan trọng), vàng (bình thường), đỏ (quan trọng)) * Hạn chót của công việc (due date)
Sắp xếp TODO theo thứ tự độ quan trọng và Due date
<p id="section4">nbsp;</p>
Làm sao để xem xét lại quá trình làm việc?
Nhật lý làm việc hàng tuần . Việc này lên được thực hiện vào đầu tuần . Có 3 nội dung quan trọng trong nhật ký làm việc (ngoài gió mây trăng cảm xúc, quan hệ với đồng nghiệp...)
* Kết quả công việc tuần này * Những công việc chưa làm? Lý do tại sao chưa hoàn thành? * Dự định cho tuần tới
Đang nghiên cứu
Làm sao để lưu lại các ý tưởng, công việc cần làm?: Dùng chức năng checklist của card trong meister. Khi có ý tưởng mới, sẽ thêm một mục trong checklist
Làm sao để tập trung vào công việc quan trọng?: Dùng chức năng tag của meister, mỗi một công việc sẽ được đánh sao (với các mức 5 sao, 3 sao, 1 sao), thể hiện mức độ quan trọng của công việc. Mỗi một sprint nên chỉ tập trung vào 10 star, một product backlog chỉ nên có 30 star.
Tài liệu của dự án: Sử dụng Google Drive, tài liệu mô tả dự án sẽ được link vào card tương ứng trong meister.

Chương 9

Agile

View online http://magizbox.com/training/agile/site/

9.1 Introduction

What is Agile? 2 Agile methodology is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.
Agile Manifesto
Scrum Framework
A manifesto for small teams doing important work 4
What is the difference between Scrum and Agile Development? 1 Scrum is just one of the many iterative and incremental agile software development method. You can find here a very detailed description of the process.
In the SCRUM methodology a sprint is the basic unit of development. Each sprint starts with a planning meeting, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made. A Sprint ends with a review or retrospective meeting where the progress is reviewed and lessons for the next sprint are identified. During each sprint, the team creates finished portions of a product.
In the Agile methods each iteration involves a team working through a full software development cycle, including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders.
Agile Stories and Teasers 2011, PRESENTATION: HOW OUR TEAM LIVES SCRUM 2010, The real life of a Scrum team – with photos 2009, How Scrum Helped Our Team Agile Tools 3 Agilefant (4/ 3/ 3) What is the difference between Scrum and Agile Development?
Agile Methodology
agile-tools.net/
A manifesto for small teams doing important work

9.2 Team

Scrum Team 1 Within the Scrum Framework three roles are defined:
Product Owner Scrum Master Development team Each of these roles has a defined set of responsibilities and only if they fulfill these responsibilities, closely interact and work together they can finish a project successfully.
Scrum Roles Stakeholders
Product Owner One of the most important things for the success of scrum is the role of the Product Owner, who serves as an interface between the team and other involved parties (stakeholders). It can be said that in companies that use scrum, the tasks and responsibilities of the particular Product Owner are never the same. Starting with the choice of that person provided with the proper and necessary skills, make them take specific trainings, up to the responsibility they take; the role of the Product Owner –short PO- is the most complex one regarding that procedure.
Often the PO has to "fight" on both sides. Whereas the team can work a certain fraction of time (time boxed) "protected" by the Scrum Master, the Product Owner often needs to deal with marketing, management or the customers in order to be able to present the software requirements (User Stories) quite precisely to the team (see the box "criteria for User Stories).
Furthermore the Product Owner is responsible for the return on investment (ROI). He validates the solutions and verifies whether the quality is acceptable or not from the end-users' point of view. He also has to decide over the importance of single features in order to prioritize these in their treatment and he has to tell the team what the product should look like in the end (product vision). Since one of the teams' tasks is to work effectively, the Product Owner must react fast on call-backs. Hence he fulfills the role of a communicator, as he must be in contact with all stakeholders, sponsors and last but not least the team throughout a project. After all it is his task to coordinate the financial side of the product development, which is successful through his continuous work and prioritizing the advancing tasks (Product Backlog). All these diverse requests demonstrate how important the selection of the "right" person for the role of the PO is for the success of a project.
The nomenclature is definite. The Product Owner is at Scrum not only the manager of a product, but also the Owner and therefore he is the one responsible for the correct creation of a product. Being a Product Owner means:
You are responsible for the success of the outcome of the product delivered by the team. You make Business decisions of importance and priorities. You deliver the vision of the product to the team. You prepare the User Stories for the team of development. You should possess severe domain knowledge. You validate the outcomes and test them for their quality. You react promptly on callbacks. You communicate on a continual base with all Stakeholders, financiers and the team. You control the financial side of the project.
Scrum Master The most obvious difference between a team leader and a Scrum Master is represented by the name itself though. Whereas one is leading the team and sets the tasks, the other one is in charge of observing that the team obeys the rules and realizes the method of Scrum entirely. The Scrum Master does not interfere into the decisions of the team regarding specifically the development, but rather is there for the team as an advisor. He only interferes actively when anybody of the team or any other participant of a project (Stakeholder) does not

obey the rules of Scrum. Whereas a team leader often gives requirements and takes responsibility for the completion of those, an experienced Scrum Master gives only impulses and advises to the team to lead the correct way, to use the right method or to choose the right technology. In fine the Scrum Master acts more like a Team Coach than a team leader.
ScrumMaster and Impediments Another important task of the Scrum Master is to get rid of all possible impediments that might disturb the work of the team. Usually problems can be classified in three different categories. The first one is problems the team cannot solve. E.g. the team cannot do any kind of performance-tests because the hardware is not in place, the IT-department does not provide Bug tracker, or the ordered software just still did not reach the team. Another impediment could be that the marketing or sales manager was there again demanding that another feature gets integrated "quickly".
The second one regards impediments that result through the organizational structure or strategic decisions. Maybe the office is not capable of handling the important meetings or teamwork – e.g. because there is no media. One mistake that occurs quite often regards the problem that the Scrum Master is seen as the personnel responsible for the team members. This is often because of the classical role of a project leader, but using Scrum it only leads to conflicts of interests and is strongly against its major principle: The team owns a management role in the method of Scrum and is therefore coequal with the Scrum Master and the Product Owner. Another aspect can be the insufficient bandwidth of the internet for the new project.
The third problem refers to the individuals. Someone needs a hand with the debugging. Another one cannot solve a task alone and needs someone else for the pair programming. Someone else has to reset his computer....
Even though a Scrum Master can't and shouldn't realize some requirements himself, he is still responsible for solving and getting rid of problems and needs to give proper criteria. This task often takes up a lot of time and requires great authority and backbone. The Scrum Master has to create an optimal working-condition for the team and is responsible for this condition to be retained, in order to meet the goals of every sprint – i.e. for a short sprint the defined requirements.
Scrum Development Team Different from other methods, in Scrum a team is not just the executive organ that receives its tasks from the project leader, it rather decides self dependent, which requirements or User Stories it can accomplish in one sprint. It constructs the tasks and is responsible for the permutation of those – the team becomes a manager. This new self-conception of the team and the therewith aligned tasks and responsibilities necessarily change the role of the team leader/project leader. The Scrum Master does not need to delegate all the work and to plan the project, he rather takes care that the team meets all conditions in order to reach the self-made goals. He cleans off any impediments, provides an ideal working environment for the team, coaches and is responsible for the observation of Scrum-rules – he becomes the so-called Servant Leader.
The changed role perception is one of the most important aspects, when someone wants to understand Scrum and with the intent to introduce it in their own company.
Scrum Roles

9.3 Artifacts

Product Backlog 1
Force-ranked list of desired functionality Visible to all stakeholders Any stakeholder (including the Team) can add items Constantly re-prioritized by the Product Owner Items at top are more granular than items at bottom Maintained during the Backlog Refinement Meeting
Product Backlog Item (PBI) 1
Specifies the what more than the how of a customer-centric feature Often written in User Story form Has a product-wide definition of done to prevent technical debt May have item-specific acceptance criteria Effort is estimated by the team, ideally in relative units (e.g., story points) Effort is roughly 2-3 people 2-3 days, or smaller for advanced teams Sprint Backlog 1
Consists of committed PBIs negotiated between the team and the Product Owner during the Sprint Planning Meeting
Scope commitment is fixed during Sprint Execution
Initial tasks are identified by the team during Sprint Planning Meeting
Team will discover additional tasks needed to meet the fixed scope commitment during Sprint execution
Visible to the team
Referenced during the Daily Scrum Meeting
Sprint Task 1
Specifies how to achieve the PBI's what
Requires one day or less of work
Remaining effort is re-estimated daily, typically in hours
During Sprint Execution, a point person may volunteer to be primarily responsible for a task
Owned by the entire team; collaboration is expected
Personal Sprint Board with Sticky Notes (Windows)
Sprint Burndown Chart 1
Indicates total remaining team task hours within one Sprint Re-estimated daily, thus may go up before going down Intended to facilitate team self-organization Fancy variations, such as itemizing by point person or adding trend lines, tend to reduce effectiveness at encouraging collaboration Seemed like a good idea in the early days of Scrum, but in practice has often been misused as a management report, inviting intervention. The ScrumMaster should discontinue use of this chart if it becomes an impediment to team self-organization.
Product / Release Burndown Chart
Tracks the remaining Product Backlog effort from one Sprint to the next May use relative units such as Story Points for Y axis Depicts historical trends to adjust forecasts scrum-reference-card

9.4 Meetings

Meetings Scrum Meetings 1 Sprint Planning Daily Scrum Sprint Review Sprint Retrospective Product Backlog Refinement
Sprint Planning 1
Goals – Discuss and make sure the whole team understands the upcoming Work Items to deliver (quantity, complexity) – Select the work items to achieve during

the Sprint (Create the Sprint Backlog) – Rationally forecast the amount of work and commit to accomplish it – Plan how this work will be done
Attendees 1st part (what?): Whole team 2nd part (how?): Even though the product owner does not attend, he should remain to answer questions. Duration The meeting is time-boxed: 2 hours / week of sprint duration.
Typical meeting roadmap In Scrum, the sprint planning meeting has two parts:
1. What work will be done?
– the Product Owner presents the ordered product backlog items to the development team – the whole Scrum team collaborates to understand the work and select work items starting from the top of the product backlog
2. How will the work be accomplished?
– the development team discusses to define how to produce the next product increment in accordance with the current Definition of Done – The team does just the sufficient design and planning to be confident of completing the work during the sprint – The upcoming work to be done in the early days is broken down into small tasks of one day or less – Work to be done later may be left in larger units to be decomposed later (this is called Just-In-Time planning in Lean) – Final commitment to do the work
Important to know / Good practices – The development team is alone responsible to determine the number of product backlog items that will be "pulled" to the sprint. Nobody else should interfere in that decision, based on the current state of the project, the past performance and the current availability of the team. – It is a good practice to set a sprint goal to keep focus on the big picture and not on the details. – It is necessary for the Sprint planning meeting success that the product backlog is well ordered and refined. This is the Product Owner's responsibility. – The development team is responsible to decide how to do the work (self-organization). – There is no point, especially at the beginning, to try to make hourly estimates of the work and compare them to availability. This habit is inherited from traditional PM methods and may be counterproductive, as reduces ownership of the team's commitments. The best for the team is to intuitively estimate its own capacity to do the work, reduce the amount of committed work to deliver, and get experienced at estimating during the first sprints.
Daily Scrum 1
Goals Ensure, every day, at the same palace, that the team is on track for attaining the sprint goal and that team members are all on the same page. Spot blocks and problems. The Scrum Master can resolve impediments that are beyond the team's control. This is NOT a management reporting meeting to anyone. The team members communicate together as a team. Based on what comes up in the meeting, the development team reorganizes the work as needed to accomplish the sprint goal. Create transparency, trust and better performance. Build the team's self-organization and self-reliance. Attendees Development Team + Scrum Master. The Product Owner presence is not required but it I almost always interesting for him / here to be present, especially to clarify requirements if needed. Other stakeholders can attend to get a good and quick overview of the progress and project status, although having managerial presence may cause a "trigger" effect and pollute the meeting's effectiveness. Duration No more than 15 minutes. A good practice is to allow 2 minutes to each team member. Typical meeting roadmap Every team member answers 3 questions:
What I have accomplished since the last daily Scrum What I plan to accomplish

between now and the next daily Scrum What (if anything) is impeding my progress No discussion during the meeting. Only brief questions to clarify the previous list. Any subsequent discussion should take place after the meeting with the concerned team members.

Important to know / Good practices The Daily Scrum is sometimes called "Daily Stand Up". This name gives a good overview of the tone and shortness of this meeting. Each team member moves the tasks in the taskboard while speaking (if not done before) The Sprint Burndown chart can be updated by the Scrum Master at the end of the meeting Having a "being blocked" task list is useful. Personally I add a dedicated column in the taskboard Only team members speak during the daily Scrum. Nobody else. TheDaily Scrum is a proof of the team's sef-organizing capacity as it shows how much team members collaborate together as they address themselves to the whole team (and not only the Scrum Master) It is quite common to uncover additional tasks during the Sprint to achieve the Sprint Goal Sprint Review Meeting 1

Goals After Sprint execution, the team holds a Sprint Review Meeting to demonstrate a working product increment to the Product Owner and everyone else who is interested. The meeting should feature a live demonstration, not a report. Attendees Product Team Product Owner When After Sprint execution Duration 4 hours given a 30-day Sprint (much longer than anyone recommends nowadays), the maximum time for a Sprint Review Meeting is 4 hours Roadmap Demonstration After the demonstration, the Product Owner reviews the commitments made at the Sprint Planning Meeting and declares which items he now considers done. For example, a software item that is merely "code complete" is considered not done, because untested software isn't shippable. Incomplete items are returned to the Product Backlog and ranked according to the Product Owner's revised priorities as candidates for future Sprints. The ScrumMaster helps the Product Owner and stakeholders convert their feedback to new Product Backlog Items for prioritization by the Product Owner. Often, new scope discovery outpaces the team's rate of development. If the Product Owner feels that the newly discovered scope is more important than the original expectations, new scope displaces old scope in the Product Backlog. The Sprint Review Meeting is the appropriate meeting for external stakeholders (even end users) to attend. It is the opportunity to inspect and adapt the product as it emerges, and iteratively refine everyone's understanding of the requirements. New products, particularly software products, are hard to visualize in a vacuum. Many customers need to be able to react to a piece of functioning software to discover what they will actually want. Iterative development, a value-driven approach, allows the creation of products that couldn't have been specified up front in a plan-driven approach.

Sprint Retrospective Meeting 1 2

Goals At this meeting, the team reflects on its own process. They inspect their behavior and take action to adapt it for future Sprints.

When Each Sprint ends with a retrospective.

Duration 45 minutes

Roadmap

Dedicated ScrumMasters will find alternatives to the stale, fearful meetings everyone has come to expect. An in-depth retrospective requires an environment of psychological safety not found in most organizations. Without safety, the retrospective discussion will either avoid the uncomfortable issues or deteriorate

into blaming and hostility.
A common impediment to full transparency on the team is the presence of people who conduct performance appraisals.
Another impediment to an insightful retrospective is the human tendency to jump to conclusions and propose actions too quickly. Agile Retrospectives, the most popular book on this topic, describes a series of steps to slow this process down: Set the stage, gather data, generate insights, decide what to do, close the retrospective. (1) Another guide recommended for ScrumMasters, The Art of Focused Conversations, breaks the process into similar steps: Objective, reflective, interpretive, and decisional (ORID). (2)
A third impediment to psychological safety is geographic distribution. Geographically dispersed teams usually do not collaborate as well as those in team rooms.
Retrospectives often expose organizational impediments. Once a team has resolved the impediments within its immediate influence, the ScrumMaster should work to expand that influence, chipping away at the organizational impediments.
ScrumMasters should use a variety of techniques to facilitate retrospectives, including silent writing, timelines, and satisfaction histograms. In all cases, the goals are to gain a common understanding of multiple perspectives and to develop actions that will take the team to the next level.
Product Backlog Refinement Meeting 1
How to: A Great Product Backlog Refinement Workshop
PRESENTATION: HOW OUR TEAM LIVES SCRUM

Chương 10

Docker

View online http://magizbox.com/training/docker/site/
Docker is an open platform for building, shipping and running distributed applications. It gives programmers, development teams and operations engineers the common toolbox they need to take advantage of the distributed and networked nature of modern applications.

10.1 Get Started

Docker Promise Docker provides an integrated technology suite that enables development and IT operations teams to build, ship, and run distributed applications anywhere.
Build: Docker allows you to compose your application from microservices, without worrying about inconsistencies between development and production environments, and without locking into any platform or language.
Composable You want to be able to split up your application into multiple services.
Ship: Docker lets you design the entire cycle of application development, testing and distribution, and manage it with a consistent user interface.
Portable across providers You want to be able to move your application between different cloud providers and your own servers, or run it across several providers.
Run: Docker offers you the ability to deploy scalable services, securely and reliably, on a wide variety of platforms.
Portable across environments You want to be able to define how your application will run in development, and then run it seamlessly in testing, staging and production.
1. Docker Architecture
Docker Containers vs Virtual Hypervisor Model
In the Docker container model, the Docker engine sits atop a single host operating system. In contrast, with the traditional virtualization hypervisor mode, a separate guest operating system is required for each virtual machine.
Docker Images and Docker Containers
2.1 Docker Images Docker images are the build component of Docker.
For example, an image could contain an Ubuntu operating system with Apache and your web application installed. Images are used to create Docker containers.

Docker provides a simple way to build new images or update existing images, or you can download Docker images that other people have already created.
Built by you or other Docker users
Stored in the Docker Hub or your local Registry
Images Tags
Images are specified by repository:tag The same image my have multiple tags The default tag is lastest Look up the repository on Docker to see what tags are available 2.2 Docker Containers Docker containers are the run component of Docker.
Docker containers are similar to a directory. A Docker container holds everything that is needed for an application to run. Each container is created from a Docker image. Docker containers can be run, started, stopped, moved, and deleted. Each container is an isolated and secure application platform.
2.3 Registry and Repository Docker registries are the distribution component of Docker.
Docker registries
Docker registries hold images. These are public or private stores from which you upload or download images. The public Docker registry is provided with the Docker Hub. It serves a huge collection of existing images for your use. These can be images you create yourself or you can use images that others have previously created. Docker Hub
Docker Hub is the public registry that contains a large number of images available for your use.
2.3 Docker Networking Docker use DNS instead etc/hosts 3. Docker Orchestration Three tools for orchestrating distributed applications with Docker Docker Machine Tool that provisions Docker hosts and installs the Docker Engine on them Docker Swarm Tool that clusters many Engines and schedules containers Docker Compose Tool to create and manage multi-container applications Installation Docker only supports CentOS 7, Windows 7.1, 8/8.1, Mac OSX 10.8 64bit
Windows, CentOS
Usage Lab: Search for Images on Docker Hub Installation: Ubuntu Installation ubuntu 14.04
Prerequisites
apt-get update apt-get install apt-transport-https ca-certificates
apt-key adv –keyserver hkp://p80.pool.sks-keyservers.net:80 –recv-keys 58118E89F3A912897C070ADBF7622
Edit /etc/apt/sources.list.d/docker.list
deb https://apt.dockerproject.org/repo ubuntu-trusty main apt-get update apt-get purge lxc-docker apt-cache policy docker-engine Install docker
apt-get update apt-get install -y –force-yes docker-engine service docker start
Run Docker
docker run hello-world docker info docker version Installation https://www.docker.com/products/docker-toolbox
Go to the Docker Toolbox page. Click the installer link to download. Install Docker Toolbox by double-clicking the installer. Docker: CentOS 7 Installation
Install Docker
make sure your existing yum packages are up-to-date. sudo yum -y update
add docker repo sudo tee /etc/yum.repos.d/docker.repo «-'EOF' [dockerrepo]
name=Docker Repository baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/enabled = 1gpgcheck = 1gpgkey = https : //yum.dockerproject.org/gpgEOF$

install the Docker package. sudo yum install -y docker-engine
start the Docker daemon sudo service docker start
verify sudo docker run hello-world Install Docker Compose
sudo yum install epel-release sudo yum install -y python-pip
sudo pip install docker-compose Docker 1.10.3
Kitematic Port Forwarding Open Virtualbox
Volumes Install Docker for Windows
Self Paced Training, Docker Fundamentals
Understand the architecture
Understand the architecture
Docker Compose and Networking
https://www.docker.com/
Orchestrating Docker with Machine, Swarm and Compose

10.2 Dev

Create New Image Create new image by commit docker commit <container ID> <yourname>/curl:1.0
Build image docker build -t ubuntu/test:1.0 . docker build -t ubuntu/test:1.0 –no-cache . Import/Export Container 1 2 Export the contents of a container's filesystem as a tar archive
docker export [OPTIONS] CONTAINER docker export red$_{p}anda > latest.tar$
docker import file|URL|- [REPOSITORY[:TAG]] docker import http://example.com/exampleimage.tgz
cat exampleimage.tgz | docker import - exampleimagelocal:new Save/Load Image 3 4 Save one or more images to a tar archive
docker save [OPTIONS] IMAGE [IMAGE...] docker save busybox > busybox.tar.gz
docker load [OPTIONS] docker load < busybox.tar.gz Push Images to Docker Hub Login to docker hub docker login –username=yourhubusername –email=youremail@company.com
push docker docker push [repo:tag]
tag an image into a repository docker tag [OPTIONS] IMAGE[:TAG] [REGISTRYHOST/][USERNAME/]NAME[:TAG] Windows 5 Useful Docker Tips and Tricks on Windows
Commandline Reference: export
Commandline Reference: import
Commandline Reference: save
Commandline Reference: load

10.3 Dockerfile

Build a song with Dockerfile
A dockerfile is a configuration file that contains instructions for building a Docker image.
Provides a more effective way to build images compared to using docker commit
Easily fits into your continuous integration and deployment process. Command Line
'FROM' instruction specifies what the base image should be FROM java:7

'RUN' instruction specifies a command to execute RUN apt-get update apt-get install -y curl vim openjdk-7-jdk
CMD Defines a default command to execute when a container is created CMD ["ping", "127.0.0.1", "-c", "30"]
ENTRYPOINT Defines the command that will run when a container is executed ENTRYPOINT ["ping"] Volumes
Configuration Files and Directories**
COPY <src>... <dest> COPY hom* /mydir/ adds all files starting with "hom" COPY hom?.txt /mydir/ ? is replaced with any single character, e.g., "home.txt" Networking
Ports still need to be mapped when container is executed EXPOSE Configures which ports a container will listen on at runtime FROM ubuntu:14.04 RUN apt-get update RUN apt-get install -y nginx
EXPOSE 80 443
CMD ["nginx", "-g", "daemon off;"] 2.4 Linking Containers Example: Web app container and Database container
Linking is a communication method between containers which allows them to securely transfer data from one to another
Source and recipient containers Recipient containers have access to data on source containers Links are established based on container names Create a Link Create the source container first Create the recipient container and use the –link option
Best practice - give your containers meaningful names
Create the source container using the postgres docker run -d –name database postgres
Create the recipient container and link it docker run -d -P –name website –link database:db nginx Uses of Linking
Containers can talk to each other without having to expose ports to the host Essential for micro service application architecture Example: Container with the Tomcat running Container with the MySQL running Application on Tomcat needs to connect to MySQL Operating Systems for Docker 1 2 Operating System Features CoreOS Service Discovery, Cluster management, Auto-updates CoreOS is designed for security, consistency, and reliability. Instead of installing packages via yum or apt, CoreOS uses Linux containers to manage your services at a higher level of abstraction. A single service's code and all dependencies are packaged within a container that can be run on one or many CoreOS machines The New Minimalist Operating Systems
Top 5 operating systems for your Docker infrastructure

10.4 Container

Lifecycle docker create creates a container but does not start it. docker rename allows the container to be renamed. docker run creates and starts a container in one operation. docker rm deletes a container. docker update updates a container's resource limits. Starting and Stopping docker start starts a container so it is running. docker stop stops a running container. docker restart stops and starts a container. docker pause pauses a running container, "freezing" it in place. docker unpause will unpause a running container. docker wait blocks

until running container stops. docker kill sends a SIGKILL to a running container. docker attach will connect to a running container. Info docker ps shows running containers. docker logs gets logs from container. (You can use a custom log driver, but logs is only available for json-file and * journald in 1.10). docker inspect looks at all the info on a container (including IP address). docker events gets events from container. docker port shows public facing port of container. docker top shows running processes in container. docker stats shows containers' resource usage statistics. docker diff shows changed files in the container's FS. Import / Export docker cp copies files or folders between a container and the local filesystem. docker export turns container filesystem into tarball archive stream to STDOUT. Example
docker cp foo.txt mycontainer:/foo.txt docker cp mycontainer:/foo.txt foo.txt
Note that docker cp is not new in Docker 1.8. In older versions of Docker, the docker cp command only allowed copying files from a container to the host
Executing Commands docker exec to execute a command in container. To enter a running container, attach a new shell process to a running container called foo
docker exec -it foo /bin/bash. Suggest Readings: docker cheatsheet

10.5 Compose

Compose a symphony
Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a Compose file to configure your application's services. Then, using a single command, you create and start all the services from your configuration. To learn more about all the features of Compose see the list of features. Docker Compose Describes the components of an application
Box your component
.component/ docker-compose.yml app-1/ app/ file-1 file-2 Dockerfile run.sh app-2/ app/ file-1 file-2 Dockerfile run.sh Example docker-compose.yml example from docker-birthday-3 3
version: "2"
services: voting-app: build: ./voting-app/. volumes: - ./voting-app:/app ports: - "5000:80" networks: - front-tier - back-tier
result-app: build: ./result-app/. volumes: - ./result-app:/app ports: - "5001:80" networks: - front-tier - back-tier
worker: image: manomarks/worker networks: - back-tier
redis: image: redis:alpine container$_name : redisports : ["6379"]networks : -back - tier$
db: image: postgres:9.4 container$_name : dbvolumes : -"db-data : /var/lib/postgresql/data"networks : -back - tier$
volumes: db-data:
networks: front-tier: back-tier: Networks network$_mode, ports, external_hosts, link, net$
network$_mode : "bridge"network_mode : "host"network_mode : "service : [service_name]"network_mode : "container : [containername/id]"VolumesOpsrebuildyourimagesdocker-composebuild$
build or rebuild services docker-compose build [SERVICE...] docker-compose build –no-cache database
start all the containers docker-compose up -d docker-compose up

stops the containers docker-compose stop [CONTAINER] Keep a container running in compose 2
You can use one of those lines
command: "top" command: "tail -f /dev/null" command: "sleep infinity" Docker Compose Files Version 2
github issue, Keep a container running in compose
docker-compose.yml

10.6 Swarm

Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host. Because Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts 1
Architecture 2
Swarm Manager: Docker Swarm has a Master or Manager, that is a pre-defined Docker Host, and is a single point for all administration. Currently only a single instance of manager is allowed in the cluster. This is a SPOF for high availability architectures and additional managers will be allowed in a future version of Swarm with 598.
Swarm Nodes: The containers are deployed on Nodes that are additional Docker Hosts. Each Swarm Node must be accessible by the manager, each node must listen to the same network interface (TCP port). Each node runs a node agent that registers the referenced Docker daemon, monitors it, and updates the discovery backend with the node's status. The containers run on a node.
Scheduler Strategy: Different scheduler strategies (binpack, spread, and random) can be applied to pick the best node to run your container. The default strategy is spread which optimizes the node for least number of running containers. There are multiple kinds of filters, such as constraints and affinity. This should allow for a decent scheduling algorithm.
Node Discovery Service: By default, Swarm uses hosted discovery service, based on Docker Hub, using tokens to discover nodes that are part of a cluster. However etcd, consul, and zookeeper can be also be used for service discovery as well. This is particularly useful if there is no access to Internet, or you are running the setup in a closed network. A new discovery backend can be created as explained here. It would be useful to have the hosted Discovery Service inside the firewall and 660 will discuss this.
Standard Docker API: Docker Swarm serves the standard Docker API and thus any tool that talks to a single Docker host will seamlessly scale to multiple hosts now. That means if you were using shell scripts using Docker CLI to configure multiple Docker hosts, the same CLI would can now talk to Swarm cluster and Docker Swarm will then act as proxy and run it on the cluster.
Create Swarm Simple Swarm Swarm with CentOS Swarm with Windows Swarm and Compose 5 Create docker-compose.yml file
Example
wget https://docs.docker.com/swarm/swarm$_at_scale/docker-compose.ymlDiscovery$4
Scheduling 3 Docker Swarm: CentOS Prerequisites Docker 1.10.3 CentOS 7
Create cluster 1 Receipts: consult
Configure daemon in each host 6

Edit /etc/systemd/system/docker.service.d/docker.conf
[Service] ExecStart= ExecStart=/usr/bin/docker daemon -H fd:// -D -H tcp://<node$_ip$ >: $2375 - -cluster - store = consul : // < consul_ip >: 8500 - -cluster - advertise =< node_ip >: 2375 Restartdocker$
systemctl daemon-reload systemctl restart docker Verify docker run with config
ps aux | grep docker | grep -v grep Run cluster
In discovery host
run consul docker run -d -p 8500:8500 –name=consul progrium/consul -server -bootstrap In swarm manage host
run swarm master docker run -d -p 2376:2375 swarm manage consul://<consul$_ip$ >: $8500/swarmInswarmnodehost$
open 2375 port firewalld-cmd –get-active-zones firewall-cmd –zone=public –add-port=2375/tcp –permanent firewall-cmd –reload
run swarm node docker run -d swarm join –addr=<node$_ip$ >: $2375consul : // < consul_ip >: 8500/swarmQuestion : Whymanagerandconsulcannotruninthesamehost?(becausedocker-proxy?)$
Verify 2
manage docker -H :2376 info export DOCKER$_HOST = tcp : // < ip >: 12375dockerinfodockerrun - d - Predis$
debug a swarm docker swarm –debug manage consul://<consul$_ip$ >: $8500DockerSwarm : SimpleSwarmSimpleSwarminonenodewithtoken1Changefilevi/etc/default/docker$
DOCKER$_OPTS = "-Htcp : //0.0.0.0 : 2375-Hunix : ///var/run/docker.sock"$
Restart docker service docker restart
create swarm token docker run –rm swarm create > token$_id$
Run swarm node docker run -d swarm join –addr=ip:2375 token://token$_iddockerps$
Run swarm manager docker run -d -p 12375:2375 swarm manage token://token$_id$
Swarm info docker -H :12375 info
export DOCKER$_HOST = tcp : //ip : 12375dockerinfodockerrun-d-PredisDockerSwarm : WindowsPrerequisitesDocker1.10.3GiithiuvchythDockerSwarm$
Docker-swarm, Cannot connect to the docker engine endpoint
Docker Swarm Part 3: Scheduling
Docker Swarm Part 2: Discovery
Deploy the application
Configuring and running Docker on various distributions

10.7 UCP

Docker Universal Control Plane
Docker Universal Control Plane provides an on-premises, or virtual private cloud (VPC) container management solution for Docker apps and infrastructure regardless of where your applications are running.
[embed]https://www.youtube.com/watch?v=woDzgqtZZKc[/embed]

10.8 Labs

Lab 1:
1. Create a container from an Ubuntu image and run a bash terminal docker run -it ubuntu /bin/bash 2.Inside the container install curl apt-get install curl

3. Exit the container terminal Ctrl-P 4. Run ‘docker ps -a‘ and take not of your container ID 5. Save the container as a new image. For the repository name use <yourname>/curl. Tag the image as 1.0 6. Run ‘docker images‘ and verify that you can see your new image Lab 2: Run Ubntu
FROM ubuntu:14.04 RUN apt-get update apt-get install -y curl vim 1. Go into the test folder and open your Dockerfile from the previous exercise
2. Add the folloing line to the end CMD ["ping", "127.0.0.1", "-c", "30"]
3. Build the image docker build -t <yourname>/testimage:1.1 .
4. Execute a container from the image and observe the output docker run <yourname>/testimage:1.1
5. Execute another container from the image and specify the echo command docker run <yourname>/testimage:1.1 echo "hello world"
6. Observe how the container argument overrides the CMD instruction Lab:
1) In your home directory, create a folder called test 2) In the test folder, create a file called ‘Dockerfile‘ 3) In the file, specify to use Ubuntu 14.04 as the base image FROM ubuntu:14.04 4) Write an instruction to install curl and vim after an apt-get update RUN apt-get update apt-get install -y curl vim 5) Build an image from Dockerfile. Give it the repository <yourname>/testimage and tag it as 1.0 docker build -t johnnytu/testimage:1.0 . 6) Create a container using your newly build image and verify that curl and vim are installed. Lab: Run a container and get Terminal Access
* Create a container using the ubuntu image and connect to STDIN and a terminal docker run -i -t ubuntu /bin/bash * In your container, create a new user using your first and last name as the username addusername username * Exit the container exit * Notice how the container shutdown * Once again run: docker run -i -t ubuntu /bin/bash * Try and find your user ‘cat /etc/passwd‘ * Noice that it does not exist. Lab: Run a web application inside a container
The -P flag to map container ports to host ports docker run -d -P tomcat:7 docker ps check your image details runing go to <server url>:<port number> verify that you can see the Tomcat page Lab Create a Docker Hub Account
Lab Push Images to Docker Hub
Login to docker hub $dockerlogin--username = yourhubusername--email = youremail@company.comPassword : WARNING : logincredentialssavedinC : \Omega docker.jsonLoginSucceeded$
Use ‘docker push‘ command docker push [repo:tag]
Local repo must have same name and tag as the Docker Hub repo
Used to rename a local image repository before pushing to Docker Hub docker tag [image ID] [repo:tag] docker tag [local repo:tag] [Docker Hub repo:tag]
> tag image with ID (trainingteam/testexample is the name of repository on Docker hub) docker tag edfc212de17b trainingteam/testexample:1.0
> tag image using the local repository tag docker tag johnnytu/testimage:1.5 trainingteam/testexample

10.9 GUI

Environment
Ubuntu 14.04 Docker Engine 1.11 note: I can’t make it run on CentOS.
Dockerfile 1 [code]
Base docker image FROM debian:sid MAINTAINER Jessica Frazelle jess@docker.com

ADD https://dl.google.com/linux/direct/google-talkplugin$_current_amd64.deb /src/google-talkplugin_current_amd64.deb$
Install Chrome RUN echo 'deb http://httpredir.debian.org/debian testing main' » /etc/apt/sources.list apt-get update apt-get install -y ca-certificates curl hicolor-icon-theme libgl1-mesa-dri libgl1-mesa-glx libv4l-0 -t testing fonts-symbola –no-install-recommends curl -sSL https://dl.google.com/linux/linux$_signing_key.pub|apt-keyadd- echo"deb[arch=amd64]http://dl.google.com/linux/chrome/deb/stablemain" > /etc/apt/sources.list.d/google.list apt-getupdateapt-getinstall-y google-chrome-stable --no-install-recommends dpkg-i'/src/google-talkplugin_current_amd64.deb' apt-getpurge--auto-remove-ycurl rm-rf/var/lib/apt/lists/ rm-rf/src/.deb$
COPY local.conf /etc/fonts/local.conf
RUN rm /etc/apt/sources.list.d/google-chrome.list
RUN apt-get update RUN apt-get install -y libcanberra-gtk-module RUN apt-get install -y dbus libdbus-1-dev libxml2-dev dbus-x11
Autorun chrome ENTRYPOINT ["google-chrome"] CMD ["–user-data-dir=/data"] [/code]
Build Image [code] docker build -t chrome . [/code]
Docker Run [code] xhost + docker run -it –net host –cpuset-cpus 0 -v /tmp/.X11-unix:/tmp/.X11-unix -e DISPLAY=unix$DISPLAY -v$HOME/chrome/data/Downloads:/root/Downloads -v $HOME/chrome/data/:/data -v/deb/shm:/dev/shm --device/dev/sndchrome- -user-data-dir=/datawww.google.com[/code]$
Known Issue Issue 1: Failed to load module "canberra-gtk-module" 2
[code] Gtk-Message: Failed to load module "canberra-gtk-module" [/code]
Issue 2: Chrome crash 3
Solution: add -v /dev/shm:/dev/shm

- https://hub.docker.com/r/jess/chrome/ /dockerfile/
- http://fabiorehm.com/blog/2014/09/11/running-gui-apps-with-docker/comment-2515573749

Chương 11

Lean

View online http://magizbox.com/training/lean/site/
Lean startup is a method for developing businesses and products first proposed in 2008 by Eric Ries. Based on his previous experience working in several U.S. startups, Ries claims that startups can shorten their product development cycles by adopting a combination of business-hypothesis-driven experimentation, iterative product releases, and what he calls validated learning. Ries' overall claim is that if startups invest their time into iteratively building products or services to meet the needs of early customers, they can reduce the market risks and sidestep the need for large amounts of initial project funding and expensive product launches and failures.

11.1 Lean Canvas

11.2 Workflow

Life's too short to build something nobody wants. What separates successful startups from unsuccessful ones is that they find a plan that works before running out of resources.
Build-Measure-Learn The fundamental activity of a startup is to turn ideas into products, measure how customers respond, and then learn whether to pivot or persevere. All successful startup processes should be geared to accelerate that feedback loop.
Step 1: Document your Plan A. Writing down the initial vision using Lean Canvas and then sharing it with at least one other person
Step 2: Identify the Riskiest Parts Formulate Falsifiable Hypotheses
Falsifiable Hypothesis = [Specific Repeatable Action] will [Expected Measurable Action] The biggest risk is building something nobody wants. The risks can be divided in:
Stage 1: Problem/Solution Fit. Do I have a problem worth solving?
Answer these questions:
Is it something customers want? (must-have) Will they pay for it? If not, who will? (viable) Can it be solved? (feasible) Stage 2: Product/market Fit. Have I built something people want?"

Qualitative metrics (interviews) Quantitative metrics for measuring product/-market fit. Stage 3: Scale. How do I accelerate growth?
After product/market fit, raise a big round of funding to scale the business. Traction is needed! Seek External advice and ask specific questions.
Systematically test your Plan. Based on the scientific method run experiments. First build hypothesis that can be easily disproved and use artifacts in front of customers to "measure" their response using qualitative and quantitative data to derive "learning" to validate or refute the hypothesis.
The Lean Startup is also about being efficient. In all the stages we can take actions to be more efficient: eliminate don't-needs at the MVP, deploy continuously, make feedback easy for customers, don't push for features, etc.
The Lean Startup methodology is strongly rooted in the scientific method, and running experiments is a key activity. Your job isn't just building the best solution, but owning the entire business model and making all the pieces fit.

11.3 Validated Learning

Startups exist not to make stuff, make money, or serve customers. They exist to learn how to build a sustainable business. This learning can be validated scientifically, by running experiments that allow us to test each element of our vision.

Tài liệu tham khảo

Goodfellow, Ian / Bengio, Yoshua / Courville, Aaron (2016): *Deep Learning.* , MIT Press.

Chỉ mục

bài toán chờ xe buýt, 20

kì vọng, 16

phân phối đều, 20

phương sai, 17

Ghi chú

19/01/2018 Nay tìm được khóa CS 109 của bác Chris Piech quá hay . 5
17/01/2018 Lòng vòng thế nào hôm nay lại tìm được blog của bạn Đỗ Minh Hải, rất hay 18
01/11/2017 Không biết mình có phải làm nghiên cứu không nữa? Vừa kiêm phát triển, vừa đọc paper mỗi ngày. Thôi, cứ (miễn cưỡng) cho là nghiên cứu viên đi. 24