Chapter 2: Bayesian Updating in a Linguistic Framework

The formal characterization of language learning from Yang (2002) consists of a language learning algorithm L, a set S of potential states the learner can be in, and experience from the linguistic environment E. The language learning algorithm L contains specifications for (a) the data intake the learner uses to update beliefs in available hypotheses and (b) the update procedure itself. In this chapter, I will describe the instantiation of the update procedure I will use for the case studies in the following chapters: an adapted form of Bayesian updating. Specifically, I will demonstrate how a standard implementation of this updating procedure (Manning & Schütze, 1999) can be adapted to language learning problems.

2.1 Bayesian Updating: Overview

Bayesian updating is a probabilistic updating procedure that is widely used in natural language processing tasks to update the probabilities of alternate available hypotheses (Manning & Schütze, 1999). Specifically, it calculates the conditional probability of the hypothesis, given the data. Probabilistic reasoning has been shown to be the optimal strategy for solving problems and making decisions given noisy or incomplete information (J. Pearl, 1996). Like many other systems, the linguistic system is often learned from observable data that is highly ambiguous and exception-filled. Thus, a probabilistic component seems necessary to the language learning mechanism.

There is also evidence for the psychological validity of a procedure like Bayesian updating as a method used by adult humans (Tenenbaum & Griffiths, 2001; Cosmides & Tooby, 1996; Staddon, 1988) and infants (Gerken, 2006). Specifically, these studies demonstrate probabilistic convergence on the more restrictive hypothesis compatible with the observable data. This is in line with the Bayesian updating procedure adopted here when there are two hypotheses under consideration that differ in their level of restrictiveness (section 2.1.5).

The main purpose of Bayesian updating is to infer the likelihood of a given hypothesis, given a series of examples as input. The implementation of Bayesian updating depends greatly on the structure of the hypothesis space, since the relation of the hypotheses to each other affects how probability is shifted between the different hypotheses. I will now examine several instances of hypothesis spaces below and their effect on Bayesian updating.

2.1.1 A Simple Case: Two Non-overlapping Hypotheses, Equally Likely

Suppose there are two non-overlapping hypotheses in the set: A and B. By non-overlapping, I mean that the examples in the input will either favor A or favor B unambiguously. There are no examples that signal (or can be accounted for by) both A and B – each hypothesis covers a distinct set of data points. Suppose also that the learner who will be using Bayesian updating has no reason to be biased towards one hypothesis, so the initial probabilities assigned to both A and B are 0.5. These are the prior probabilities associated with each hypothesis.

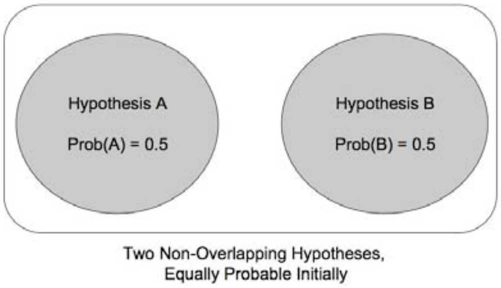


Figure 1. Two non-overlapping hypotheses, equally probable initially. The shading reflects how much probability is associated with each hypothesis.

The learner then encounters some amount of data (say d_1 data points) and uses Bayesian updating to shift the probability mass between A and B to reflect the distribution in the data intake. Each data point will cause the learner to shift the probabilities a small amount until the probability distribution among the hypotheses eventually matches the probability distribution encountered in the intake.

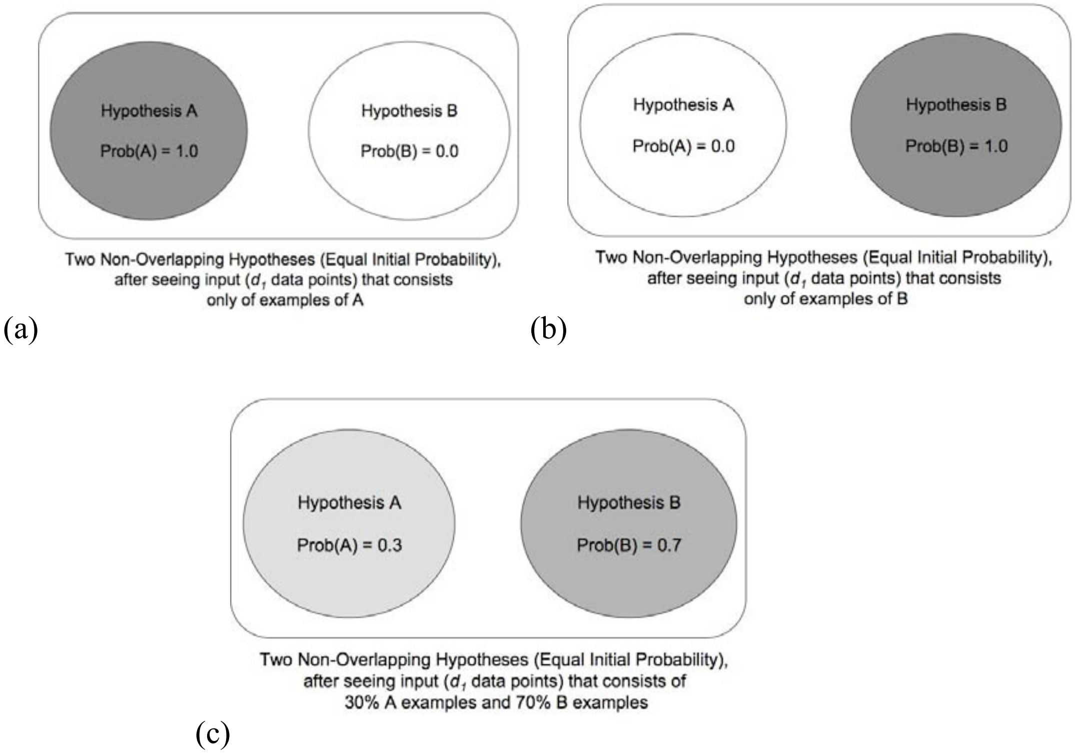

Hypothesis A
Prob(A) = 1.0
Hypothesis B
Prob(B) = 0.0
Two Non-Overlapping Hypotheses (Equal Initial Probability), after seeing input (d_1 data points) that consists only of examples of A
(a)
Hypothesis A
Prob(A) = 0.0
Hypothesis B
Prob(B) = 1.0
Two Non-Overlapping Hypotheses (Equal Initial Probability), after seeing input (d_1 data points) that consists only of examples of B
(b)
Hypothesis A
Prob(A) = 0.3
Hypothesis B
Prob(B) = 0.7
Two Non-Overlapping Hypotheses (Equal Initial Probability), after seeing input (d_1 data points) that consists of 30% A examples and 70% B examples
(c)

Figure 2. Two non-overlapping hypotheses with equal initial probability after seeing various distributions of intake (the total amount is quantified as d_1 data points). The shading reflects how much probability is associated with each hypothesis.

If the data intake consists only of examples of A, the learner will eventually shift the probability so A is 1.0 and B is 0.0 (2a).[2] Conversely, if the data intake consists only of examples of B, the learner will eventually shift the probability so A is 0.0 and B is 1.0 (2b). In each of these cases, the learner shifts all the probability to a single hypothesis, thereby converging on one hypothesis as correct. However, it is possible that the learner will encounter a mixed distribution between A and B in the data intake. If so, the learner will shift the probability to reflect the bias in the perceived distribution since the target state is a probabilistic distribution between A and B. As a concrete example, if the input is consistently 30% A examples and 70% B examples, the learner will eventually shift the probability of A to be significantly less than that of B, reflecting the 30-70 distribution (2c).

2.1.2. A Variant on the Simple Case: Two Non-overlapping Hypotheses, with an Initial Bias for One Hypothesis

Suppose the hypothesis space again has two non-overlapping hypotheses, A and B. However, suppose the learner is biased towards A initially, so A has a higher prior probability associated with it than B does. For example, let the initial probability assigned to A be 0.7, and the initial probability assigned to B be 0.3. This scenario could represent a case where A is the default hypothesis and B is the exceptional (or marked) hypothesis – thus, B has a lower prior probability.

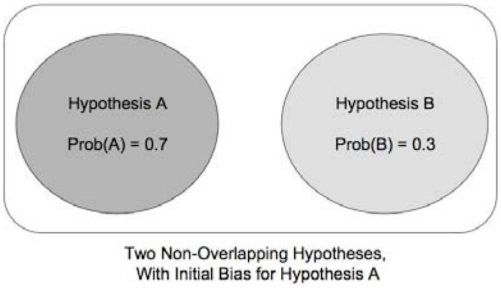
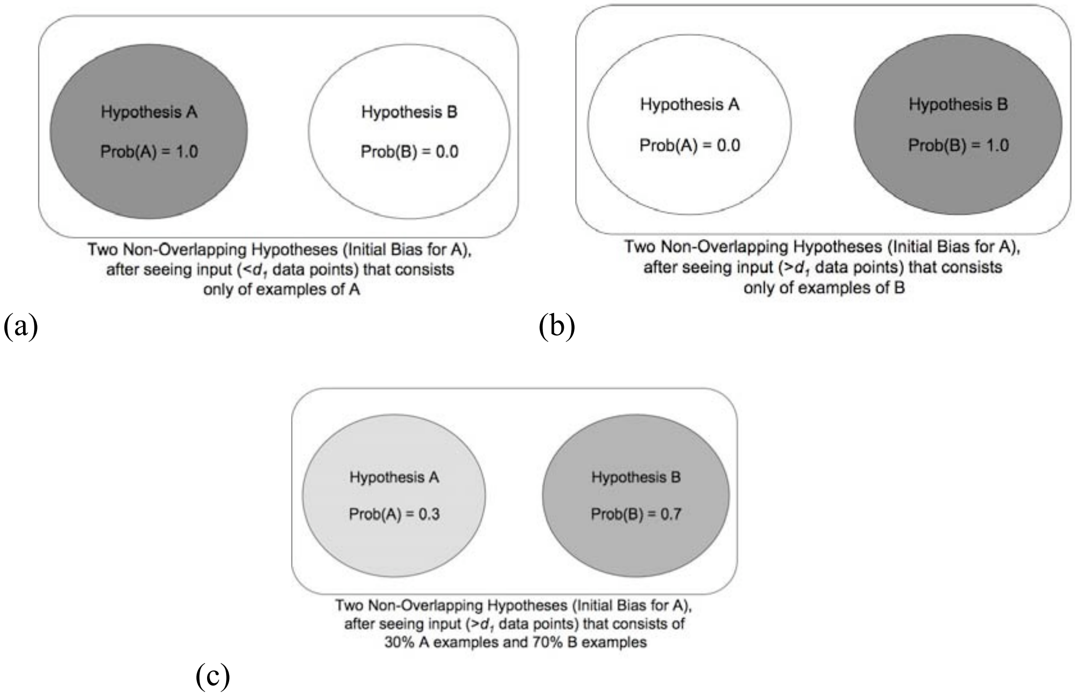

Hypothesis A
Prob(A) = 0.7
Hypothesis B
Prob(B) = 0.3
Two Non-Overlapping Hypotheses,
With Initial Bias for Hypothesis A

Figure 3. Two non-overlapping hypotheses, with an initial bias towards hypothesis A. The shading reflects how much probability is associated with each hypothesis.

The learner then encounters some amount of data and uses Bayesian updating to shift the probability mass between A and B to reflect the distribution in the data intake. As before, a learner encountering all A or all B examples will eventually shift the probability so that one hypothesis is 1.0 while the other is 0.0. However, because the prior probability of A is higher than that of B, it will take a smaller number of A examples to cause the probability of A to reach 1.0 (less than the d_1 data points in the unbiased hypothesis space) (4a). Conversely, since B is the disfavored hypothesis

[2] However, it is possible that the endpoints (0.0 and 1.0) will only be reached in the limit. Still, after encountering overwhelming data in support of one hypothesis over the other, the learner using Bayesian updating will likely be very *near* the endpoints. This point will hold true for all Bayesian updating examples in the remaining sections of this chapter.

initially, it will take a larger number of B examples to cause the probability of B to reach 1.0 (more than the d_1 data points in the unbiased hypothesis space) (4b). If the data intake has a mixed distribution, the same logic applies: a data distribution favoring A will be reflected more quickly in the probabilities the learner assigns to the hypotheses than a data distribution favoring B (4c).



Two Non-Overlapping Hypotheses (Initial Bias for A), after seeing input (<d_1 data points) that consists only of examples of A

(a)

Two Non-Overlapping Hypotheses (Initial Bias for A), after seeing input (>d_1 data points) that consists only of examples of B

(b)

Two Non-Overlapping Hypotheses (Initial Bias for A), after seeing input (>d_1 data points) that consists of 30% A examples and 70% B examples

(c)

Figure 4. Two non-overlapping hypotheses with an initial bias for hypothesis A after seeing various distributions and quantities of intake. The shading reflects how much probability is associated with each hypothesis.

2.1.3 A Less Simple Case: Two Overlapping Hypotheses, Equally Likely

Suppose there are two overlapping hypotheses in the set: A and B. By overlapping, I mean that there are two types of examples, unambiguous and ambiguous. Unambiguous examples either signal A or signal B. Ambiguous examples can be accounted for by both hypotheses. Thus, while each hypothesis has a unique subset of examples associated with it, there is also a subset that can be covered by both hypotheses. Suppose also that the learner has no reason to be biased towards one hypothesis, so the initial probabilities assigned to both A and B are 0.5.

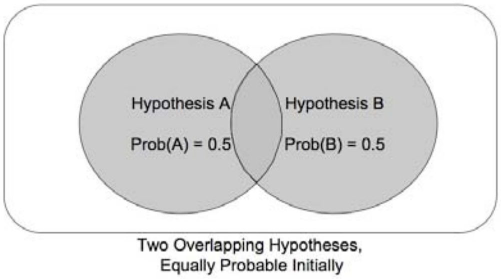


Figure 5. Two overlapping hypotheses, with equal probability initially. The shading reflects how much probability is associated with each hypothesis.

The learner then encounters some amount of data and uses Bayesian updating to shift the probability mass between A and B to reflect the distribution in the data intake. The important consideration is whether a given data point is unambiguous or ambiguous. If unambiguous (for either A or B), the updating will work the same as in the simple non-overlapping case, and the probability will be shifted slightly in favor of the hypothesis the data point is unambiguous for.

However, if the data point is ambiguous, the learning procedure must decide what to do with it. One possibility is to simply ignore the data point – this is the same as applying an unambiguous data filter that updates based only on unambiguous data points. This is a filter that will be explored in detail in chapters 4 and 5. Another possibility is to employ some strategy to deal with the ambiguous data point: use knowledge of the hypothesis space layout to assign partial credit (an approach explored in section 2.1.5 and chapter 3), use an informed guessing strategy (Fodor & Sakas, 2001), or randomly assign the data point to one hypothesis based on the current probabilities of both hypotheses (Yang, 2002). The random assignment method assumes that the effect of such ambiguous data will wash out in the face of the unambiguous data.

If the learner uses some strategy to extract information from an ambiguous data point in the overlapping hypothesis scenario, the learner will need to encounter more *total* data points than in the equivalent non-overlapping hypothesis scenario in order to converge on a hypothesis (more than d_l data points). This is simply a result of using both unambiguous and ambiguous data points to update the probabilities. Interestingly, if the learner uses an unambiguous data filter and ignores ambiguous data points, then we have a learning scenario that is very similar to the non-overlapping scenario: the learner must encounter d_l *unambiguous* data points in order to converge on the correct hypothesis. (In the non-overlapping hypothesis space, all data points are unambiguous.) Still, the total quantity of data points the learner encounters in the overlapping case will be greater than d_l, since the learner encounters both unambiguous and ambiguous data points. However, the only data points that cause any updating are the d_l unambiguous ones.

2.1.4 A Variant of the Less Simple Case: Two Overlapping Hypotheses, with an Initial Bias for One Hypothesis

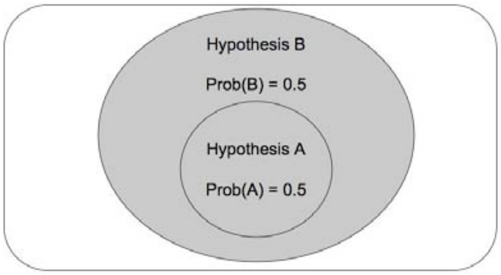
A variant of the overlapping case has biased initial probabilities. For instance, suppose hypothesis A has a prior probability of 0.7 while hypothesis B has a prior probability of 0.3. There are unambiguous examples of A, unambiguous examples of B, and ambiguous examples that can be accounted for by both A and B.

In terms of how the model deals with unambiguous and ambiguous data points, this scenario works the same as the unbiased overlapping scenario described in the previous section. The learner can either ignore the ambiguous data points, or employ some method to attribute them to one hypothesis.

However, as in the biased non-overlapping scenario described before, the number of data points the learner must encounter to converge on a hypothesis depends on how the data intake distribution relates to the prior probability distribution. If the data intake distribution is biased in the same direction as the prior probability distribution (say, 0.8 for A and 0.2 for B), the learner will need to encounter fewer data points to converge on the correct probability distribution. Conversely, if the data intake distribution is biased in the opposite direction from the prior probability distribution (say, 0.2 for A and 0.8 for B), the learner will need to encounter more data points to converge on the correct probability distribution.

2.1.5 An Even Less Simple Case: Two Overlapping Hypotheses in a Subset Relation, Equally Likely

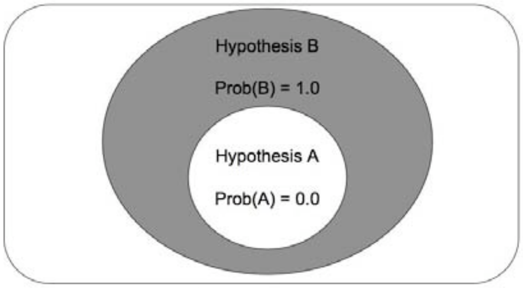
Suppose the hypothesis space again consist of two overlapping hypotheses, but one hypothesis is a subset of the other hypothesis. Let A be a subset of B, so all examples of A are also examples of B (Tenenbaum & Griffiths, 2001; Manzini & Wexler, 1987; Berwick, 1985; Berwick & Weinberg, 1984; Pinker, 1979). That is, while B has unambiguous examples, there are no unambiguous examples for A – all examples covered by hypothesis A can also be covered by hypothesis B. Suppose the initial probabilities assigned to both A and B are 0.5.



Two Overlapping Hypotheses in a Subset Relation,
Equally Probable Initially

Figure 6. Two overlapping hypotheses in a subset relation, with equal probability initially. The shading reflects how much probability is associated with each hypothesis.

Suppose the learner encounters only unambiguous examples for B in the data intake (say, d_2 data points). Eventually, the learner will shift all the probability to B (B = 1.0, A = 0.0).



Two Overlapping Hypotheses in a Subset Relation, after seeing input (d_2 data points) that consists only of examples of B

Figure 7. Two overlapping hypotheses in a subset relation with equal probability initially, after seeing d_2 data points that are unambiguous for hypothesis B. The shading reflects how much probability is associated with each hypothesis.

But what if hypothesis A (the subset hypothesis) is the correct one for the target language? All examples covered by hypothesis A are also covered by hypothesis B – they are thus ambiguous data points. It is *impossible* for the learner to encounter any unambiguous data points for hypothesis A. If the data intake consists only of these ambiguous data points, one might expect the learner to remain at a neutral probability of 0.5 for each hypothesis since these data points are compatible with each hypothesis. The learner would be doomed never to converge on the correct hypothesis, the subset hypothesis A.

One way to save the learner from this fate is to exploit the layout of the hypothesis space. The Bayesian updating procedure can take advantage of the subset-superset relation of the hypotheses to favor hypothesis A when encountering an ambiguous data point. The logic is as follows:

(1) Logic of Favoring the Subset Hypothesis For an Ambiguous Data point
(a) If hypothesis B (the superset hypothesis) was correct, the data intake should contain at least *some* examples covered only in the superset B (i.e. unambiguous B examples).
(b) If only examples covered by the subset A are encountered in the data intake, it becomes more and more unlikely that hypothesis B is correct.
(c) Therefore, the more the learner encounters only data points in the subset A (even though these are ambiguous data points), the more the learner will favor the subset hypothesis A.

A learner taking advantage of this logic will therefore consider a restriction to the subset A more and more probable as time goes on if only subset data points are encountered. This logic can be implemented in the Bayesian updating procedure itself, and has been referred to as the *size principle* (Tenenbaum & Griffiths, 2001).

Essentially, the smaller size of the set of examples covered by hypothesis A benefits hypothesis A when ambiguous examples are encountered. Specifically, the likelihood of encountering these examples given the smaller set covered by A is greater than the likelihood of encountering these examples given the larger set covered by B. So, A is slightly favored when encountering an ambiguous example covered in its subset.[3] After a sufficient number of ambiguous examples in the data intake (and, importantly for the basic version of the size principle, *no* unambiguous examples of the superset B), A will be highly favored.

We note that there is a disparity between the quantity of data points required to converge on B when using unambiguous data points as compared to the quantity required to converge on A using ambiguous data points. In particular, if the learner requires d_2 data points to reach probability p for B when encountering unambiguous B data points, the learner will require *more* than d_2 data points to reach p for A when encountering ambiguous data points. This is because the size principle allows A to only be *slightly* favored for an ambiguous data point while B is *exclusively* favored for an unambiguous B data point, though the actual amount of favoring depends on the relative sizes of A and B.

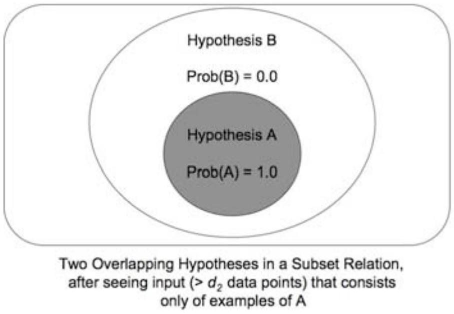

Hypothesis B
Prob(B) = 0.0
Hypothesis A
Prob(A) = 1.0
Two Overlapping Hypotheses in a Subset Relation, after seeing input (> d_2 data points) that consists only of examples of A

Figure 8. Two overlapping hypotheses in a subset relation with equal probability initially, after seeing more than d_2 data points that are examples of A. The learner uses the size principle to converge on hypothesis A. The shading reflects how much probability is associated with each hypothesis.

If the data intake has a mixed distribution (both unambiguous B examples and ambiguous examples), the unambiguous B examples will have more effect on the learner's probability distribution than the ambiguous examples that slightly favor A. Both types of data points, however, will contribute to the final probability the learner converges on. Again, the number of data points required to converge on the final probability will be greater in this case (more than d_2 data points) than if only unambiguous B examples were encountered and the correct hypothesis was B exclusively.

[3] The amount A is favored depends on the relative sizes of A and B, which the learner must already know (perhaps as a separate prior) or empirically derive from the data. The smaller A is compared to B, the more A is favored given an ambiguous data point.

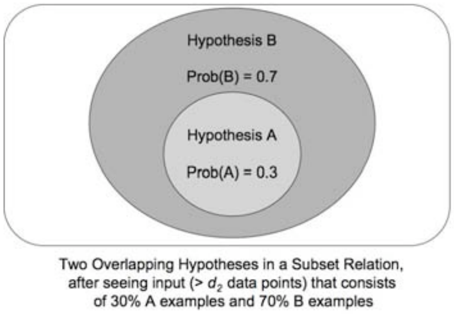

Hypothesis B
Prob(B) = 0.7
Hypothesis A
Prob(A) = 0.3
Two Overlapping Hypotheses in a Subset Relation, after seeing input (> d_2 data points) that consists of 30% A examples and 70% B examples

Figure 9. Two overlapping hypotheses in a subset relation with equal probability initially, after seeing more than d_2 data points that are a mix of unambiguous B examples and ambiguous examples in the subset A. The learner uses the size principle to converge on the probability that reflects the distribution observed in the input. The shading reflects how much probability is associated with each hypothesis.

It is important to note that exploiting the hypothesis space layout using the heuristic of the size principle is a non-trivial contribution to the learning problem for hypotheses arrayed in a subset-superset relationship. Though it is a heuristic and so not guaranteed to succeed for all cases, it nonetheless has an advantage over approaches that do not exploit the hypothesis space layout. Specifically, if only subset data are encountered, it will converge on the subset.

Suppose, however, that the learner did not use a heuristic like the size principle for learning. An instantiation of learning like this that still retains the advantages of probabilistic learning is the Naïve Parameter Learner (Yang, 2002), and the rate at which the learner shifts probabilities is represented by a parameter, gamma. A more conservative learner will have a smaller gamma, while a more liberal learner will have a larger gamma. For a data point, the Naïve Parameter Learner (NP learner) chooses one hypothesis and determines if the data point is compatible with it. If so, that hypothesis is rewarded while the remaining ones are punished; if not, it is punished while the remaining ones are rewarded. The update equations are given in (2), assuming two hypotheses, G1 and G2 (from Yang (2002)).

(2) Update equations for the NP learner for a hypothesis space with two hypotheses, G1 and G2, given a data point *d* and testing G1 against *d*

(a) If G1 is compatible with *d*,

$p_{G1} = p_{G1} + \text{gamma}*(1 - p_{G1})$

$p_{G2} = (1\text{-gamma})*p_{G2}$

(b) If G1 is not compatible with *d*,

$p_{G1} = (1\text{-gamma})*\ p_{G1}$

$p_{G2} = \text{gamma} + (1\text{-gamma})*p_{G2}$

To give a concrete example, suppose $p_{G1} = p_{G2} = 0.5$, and gamma = 0.005. Suppose data point *d* is encountered. The learner will test G1 with a 50% chance, and G2 with a 50% chance. Suppose the learner tests G1, and G1 is compatible with *d*. Then, the updated $p_{G1} = 0.5 + 0.005(1\text{-}0.5) = .5025$. The updated $p_{G2} = (1\text{-}0.005)*0.5 = 0.4975$.

As another example, suppose again that $p_{G1} = p_{G2} = 0.5$, and gamma = 0.005. Suppose data point *d* is encountered, and the learner tests G1 and finds it is not compatible with *d*. Then, the updated $p_{G1} = (1-0.005)*0.5 = 0.4975$, and the updated $p_{G2} = 0.005 + (1-0.005)*0.5 = 0.5025$.

As these two examples show for a hypothesis space that consists only of two hypotheses, when one hypothesis is punished by a certain amount, the other is rewarded by that same amount. If there were more than 2 hypotheses, the amount the tested hypothesis (G1) is punished/rewarded (gamma) would be distributed among the alternative hypotheses (G2…Gn).

The NP learner is implicitly driven by the availability of unambiguous data for one hypothesis – the alternative hypothesis is punished whenever it is used to interpret such unambiguous data points. Yet, if all data come from the subset hypothesis, then there will be no unambiguous data to punish the superset hypothesis. The NP learner encounters only ambiguous data, and is actually driven to convergence on *either* hypothesis, given sufficient data. This is shown in figure 10, assuming a hypothesis space where G1 is a subset of G2, and learning rates represented by gamma = 0.001 to 0.005, given 100,000 data points. The more liberal the learner is, the more likely the learner is to converge to one hypothesis or the other. Importantly, there is no guarantee that the learner will converge on the subset hypothesis, even though all data points come from the subset hypothesis.

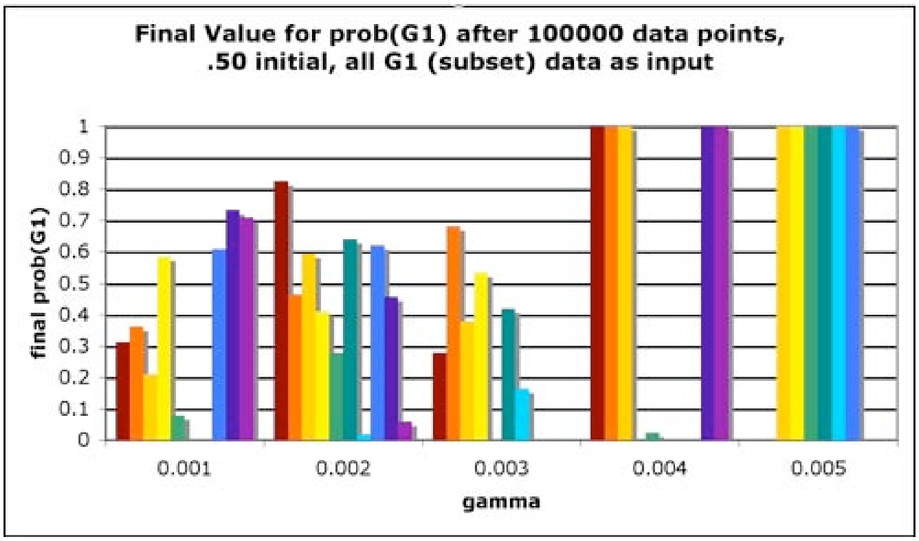


Figure 10. The NP learner, given ambiguous data from only the subset hypothesis, G1. This shows the results of 10 learners for each value of gamma, where gamma represents how conservative/liberal learning is. The NP learner has a tendency to converge to one hypothesis or the other, but is just as likely to converge to the subset G1 as the superset G2.

So, for learning cases where the hypotheses have a subset-superset relation to each other, approaches that do not exploit the hypothesis space layout will have difficulty converging on the subset hypothesis. The heuristic of the size principle provides a way to use this information to bias the learner towards the correct hypothesis.

2.1.6 Hypothesis Spaces for Language Learning

As we have seen, the layout of the hypothesis space and the relations between the hypotheses greatly affect how Bayesian updating uses the data intake to shift probability between alternate hypotheses. Crucially for Bayesian updating to be able to function, the hypothesis space must already be specified (cf. Tenenbaum, Griffiths, & Kemp (2006) for theory-based Bayesian models that emphasize this point). Otherwise, the Bayesian updating procedure has nothing over which to operate. In short, if the learner has no options to select from, Bayesian updating cannot help. A Bayesian updating procedure dovetails with a defined hypothesis space; it does not replace it.

For language learning, a simple interpretation in the parametric framework of the generative tradition (Chomsky, 1981) is that there is a hypothesis space associated with each parameter, and alternative hypotheses within a given hypothesis space correspond to opposing values for linguistic parameters. For instance, suppose we examine the syntactic parameter of Verb-Second movement. A language with Verb-Second movement (such as German) will move the tensed Verb to the second phrasal position in the main clause; a language without Verb-Second movement (such as English) will not. The Verb-Second hypothesis space thus contains the hypotheses Verb-Second-Movement and No-Verb-Second-Movement. A learner of either German or English will encounter data points from the target language and use the data intake to converge on the appropriate hypothesis for that language.

In the remaining chapters, we will examine different hypothesis spaces in different domains of linguistics. Chapter 3 explores a language learning problem that spans syntax and semantics: English anaphoric *one*. Both the syntactic and semantic hypothesis spaces for English anaphoric *one* contain two overlapping hypotheses in a subset-superset relation, and these hypotheses are equally probable initially.[4]

Chapter 4 investigates a language learning problem in Old English syntax where the target state is a probabilistic distribution between two hypotheses, Object-Verb order and Verb-Object word order, that changes over time. The hypotheses are overlapping – that is, there are both unambiguous data points for each hypothesis and ambiguous data points. Both hypotheses are equally probably initially.

Chapter 5 studies the language learning problem of English metrical phonology, which is a data set plagued by noisy and contradictory data. There are nine separate interacting parameters, each with their own hypothesis spaces. Each hypothesis space contains two hypotheses that are overlapping, and these hypotheses are equally probable initially.

[4] Note that it is an assumption of the model that these hypotheses are equiprobable initially, rather than a derivation from theoretical work or an observation from experimental work.

2.2 Bayesian Updating: General Implementation for Language Learning in a Hypothesis Space with Two Hypotheses

I will now describe how the mathematical framework of Bayesian updating (Manning & Schütze, 1999) can be adapted to a language learning hypothesis space with two non-overlapping hypotheses, A and B.[5] The only data points a learner encounters will be unambiguous for either A or B. Note that we can use this same procedure for an overlapping hypothesis space (having both unambiguous data points and ambiguous data points) if the learner employs an unambiguous data filter that ignores the ambiguous data points. In this scenario, the only data points the learner uses to update the hypothesis probabilities are the unambiguous data points, which signal either A or B.

I will then briefly sketch how to modify the Bayesian update functions to account for an overlapping hypothesis space where the hypotheses are in a subset-superset relation. The details of this modification will be described more thoroughly in chapter 3, since the specific modifications are dependent on properties of the hypotheses themselves.

2.2.1 Updating with Unambiguous Data in a Hypothesis Space with Two Hypotheses

Suppose the hypothesis space consists of two hypotheses, A and B. Let the probability of hypothesis A be p_A and the probability of hypothesis B be p_B. Below, I describe how to update p_A. Before updating, p_A represents the prior probability of A; after updating, p_A represents the posterior probability of A. The calculation of p_B is straightforward once p_A is known, since $p_B = 1 – p_A$, given that there are only two hypotheses in the hypothesis space and only one of them can be correct for any given data point.

I assume that the learner extracts information only from the current data point, and uses the information from this data point to update the probabilities of the hypotheses. Thus, the sequence length for the language learning Bayesian update function is 1. Importantly, the learner does not store data points and subsequently conduct analyses across sequences of stored data points. So, the learner is not required to remember past data points in their raw form (i.e. as utterances), which I believe is a favorable quality for a model that aims to be psychologically realistic.

[5] Of course there are several alternative approaches for the updating procedure. For instance, one might try likelihood ratios (Neyman, J. & Pearson, E., 1928) to shift probability between hypotheses, given a data point. However, likelihood ratios require a prior knowledge of the success of the test used to identify the property of interest. Mapping this to the language learning problem, the learner would need to know the success of whatever method is used to identify unambiguous data for identifying *actual* unambiguous data. To know this, the learner must know what actual unambiguous data is. To know that, the learner would need to already know the system, so as to accurately determine what unambiguous data for it is. This, however, defeats needing to learn the system in the first place.

A more promising alternative is LaPlace's rule of succession (Manning & Schütze, 1999) which normalizes the number of previous successes (e.g. data points identified as unambiguous) against the total number of data points observed. Though similar to the adaptation of Bayesian updating used in this dissertation, it does not rely on a parameter corresponding to the period of fluctuation a learner is allowed. The benefit of this parameter (*t*) is discussed in section 2.3.3.

Because there are exactly two hypotheses in the hypothesis space, I use a binomial distribution to approximate a learner's expectation of the data distribution to be encountered. The binomial distribution is centered at p_A, so the learner's expectation is about the quantity of A data points that should be encountered in the data intake.

The binomial distribution is normally used to represent the likelihood of seeing *r* data points out of *t* total with some property. For example, if these are coin flip data points, the property might be "is heads". There are only two choices for each data point: the property is either present or absent. If these are coin flip data points, the coin is either heads or it isn't (specifically, it's tails). For the hypothesis space we are considering, the data point is either an example of A, or it isn't (specifically, it's an example of B). The highest confidence is assigned to the distribution where *r* A data points are observed our of *t* total: $r = t*p_A$. Recall that the binomial distribution is centered at p_A, and so the learner is most confident that the probability of seeing an A data point is p_A. So, *r* is the most probable number of A data points expected out of *t* total, given the current probability of hypothesis A, p_A.

As an example, suppose p_A is 0.5, as it is in the initial state in an unbiased hypothesis space before the learner has encountered any data points. The binomial distribution is centered at 0.5, which we can interpret as the learner having the most confidence that half the total data points encountered will be A data points. Specifically, the learner will expect $r = t*0.5$ data points to be A data points.

To update p_A after seeing a single unambiguous A data point *a*, we can follow Manning & Schütze's (1999) Bayesian updating algorithm and calculate the maximum of the *a posteriori* (MAP) probability. The a posteriori probability is the probability that p_A is the correct probability to center the binomial distribution at after seeing an unambiguous data point A; p_A represents the expected probability of encountering an A data point. We maximize this probability because we are using a probability distribution (specifically, the binomial distribution) to approximate the learner's expectation about the data distribution to be encountered. We want the maximum a posteriori probability that comes from using this probability distribution.

We represent the a posteriori probability as $\text{Prob}(p_A| a)$[6], and calculate it using Bayes' rule:

$$(3)\ \text{Prob}(p_A \mid a) = \frac{\text{Prob}(a \mid p_A) * \text{Prob}(p_A)}{\text{Prob}(a)}$$

We can now examine individual pieces of the right hand side equation. $\text{Prob}(a \mid p_A)$ is the probability of encountering the unambiguous A data point *a*, given that p_A is the correct probability to center the binomial distribution at. For a single instance (i.e. for the single data point *a*), the probability of encountering 1 instance of *a* for 1 observation from the binomial distribution centered at p_A is

[6] $\text{Prob}(p_A| a)$ is actually intended, rather than $\text{Prob}(A| a)$. This is because we are attempting to calculate the probability that p_A is the correct probability to center the binomial distribution at, given data point *a*. So, $\text{Prob}(p_A \mid a)$ can be thought of as shorthand for Prob(p_A is the correct center for binomial distribution that will match the distribution in the learner's intake | *a*).

$\binom{1}{1} * p_A{}^1 * (1-\ p_A)^{1\text{-}1}$, which is p_A.

Prob(p_A) is the probability that p_A is the correct probability to center the binomial distribution at, i.e. that the learner should be most confident that an A data point will be encountered with probability p_A. Recall that a binomial distribution centered at p_A will assign the highest confidence to the situation where r = (p_A*t) A data points are encountered out of t total. We can instantiate Prob(p_A) as the probability of encountering r A data points out of t total in a binomial distribution for *all* values of r, from 0 to t.[7]

(4) $\text{Prob}(p_A) \ = \ \binom{t}{r} * p_A{}^r * (1-\ p_A)^{t\text{-}r}$ (for each r, $0 \le\ r\ \le\ t$)

Substituting these pieces back into equation (3) for the a posteriori probability yields (5):

(5) $\text{Prob}(p_A \mid a) \ = \ \dfrac{p_A \ * \ \binom{t}{r} * p_A{}^r * (1\text{-}p_A)^{t\text{-}r}}{\text{Prob}(a)}$ (for each r, $0\ \le\ r\ \le\ t$)

We can now calculate the MAP probability by finding the maximum of this equation. To do this, we take the derivative with respect to p_A, set it equal to 0, and solve for p_A.

(6) Calculating the MAP probability

$$\frac{d}{dp_A}(\text{Prob}(p_A \mid a) \ = \ \frac{d}{dp_A}(\frac{p_A \ * \ \binom{t}{r} * p_A{}^r * (1\text{-}p_A)^{t\text{-}r}}{\text{Prob}(a)}) \ = \ 0$$

$$\frac{d}{dp_A}(\frac{p_A \ * \ \binom{t}{r} * p_A{}^r * (1\text{-}p_A)^{t\text{-}r}}{\cancel{\text{Prob}(a)}}) \ = \ 0 \text{ (since Prob}(a)\text{ is a constant w.r.t. } p_A)$$

$$p_A \ = \ \frac{r+1}{t+1}$$

Recall that r is the previous expected number of A data points encountered out of t data points total. Hence, r = $p_{A\ \text{old}}$*t. Therefore, we write the update function for p_A after encountering unambiguous A data point a as (7a).

(7a) Update function for p_A after seeing unambiguous A data point a

$$p_A = \frac{p_{A\ \text{old}} * t + 1}{t+1}$$

[7] Note that approximating Prob(p_A) this way is a non-standard assumption. However, it yields update equations with psychologically desirable properties that other more standard assumptions do not.

An intuitive interpretation of this update function is that the numerator represents the learner's confidence that the encountered unambiguous A data point *a* is a result of the A hypothesis being correct; the denominator represents the total data encountered so far. Thus, 1 is added to the numerator because the learner is fully confident that the unambiguous data point *a* indicates the A hypothesis is correct; 1 is added to the denominator because a single data point has been encountered.

As we observed before, given that there are only two hypotheses in the hypothesis space, we can calculate the new p_B after seeing an unambiguous A data point *a* as $p_B = 1.0 - p_A$.

(7b) Update function for p_B after seeing unambiguous A data point *a*

$$p_B \ = \ 1 - \ p_A = \ 1 - \frac{p_{A\text{ old}} * t + 1}{t + 1}$$

Now, we can also derive the update functions for p_A and p_B after seeing an unambiguous B data point *b*. The derivation of the update function for p_B after seeing *b* is identical to the derivation of the update function for p_A after seeing *a*, and leads to equation (8).

(8) $$p_B = \frac{p_{B\text{ old}} * t + 1}{t + 1}$$

Again, since there are only two hypotheses in the hypothesis space, $p_B = 1.0 - p_A$. So, if we wish to track the value of p_A, we can substitute this into equation (7) and derive the update function for p_A after an unambiguous B data point *b* is encountered.

(9)

$$p_B = \frac{p_{B\text{ old}} * t + 1}{t + 1}$$

$$(1 - p_A) \ = \ \frac{(1 - p_{A\text{ old}}) * t \ + \ 1}{t + 1}$$

$$p_A \ = \ 1 - \ \frac{(1 - p_{A\text{ old}}) * t \ + \ 1}{t + 1} = \ \frac{t + 1 \ - \ (t \ - \ p_{A\text{ old}} * t \ + \ 1)}{t + 1}$$

$$p_A \ = \ \frac{p_{A\text{ old}} * t}{t + 1}$$

This update equation is identical to (7a), except that 0 is added to the numerator instead of 1. This reflects the intuitive notion that the learner should have no confidence that the A hypothesis generated the unambiguous B data point *b* just encountered.

2.2.2 Updating with Ambiguous Data in a Hypothesis Space with Two Hypotheses

We have just seen how to derive the update functions for when an unambiguous data point is encountered. Suppose, however, that the learner encounters an ambiguous data point and does not impose a filter that ignores such data for the purposes of updating. Since this data point is ambiguous between hypotheses A and B, the value added to the numerator should be a reflection of the learner's confidence that the data point indicates each of these hypotheses.

I now focus on the update of p_A (recalling, of course, that we can easily derive p_B as $1 - p_A$). If an unambiguous A data point is encountered, 1 is added to the numerator to indicate full confidence in A (and no confidence in B). Conversely, if an unambiguous B data point is encountered, 0 is added to the numerator to indicate no confidence in A (and full confidence in B). So, if a data point is ambiguous between the two hypotheses, a value greater than 0 and less than 1 should be added to the numerator. If the value added is 0.5, this would reflect no bias for either hypothesis (a truly ambiguous data point); if the value added is closer to 1, this would reflect a bias for the A hypothesis; if the value added is closer to 0, this would reflect a bias for the B hypothesis. As an example, if the learner has reason to favor A (perhaps because A and B are in a subset-superset relation with A as the subset), the value added would be greater than 0.5 but less than 1. The exact value would depend on the relative size of the sets of examples covered by A and B.

(10) Hypothetical update function for p_A after encountering an ambiguous data point, A is a subset of B, and so there is bias for hypothesis A

$$p_A = \frac{p_{A\ old} * t + m}{t + 1},\ 0.5 < m < 1$$

It is important to note that ignoring ambiguous data is *not* equivalent to adding 0.5 to the numerator when encountering an ambiguous data point. One might presume this since we interpreted the addition of 0.5 to the numerator as having no bias for either hypothesis. The crucial difference is in the invocation of the update function: if the ambiguous data point is ignored, no updating occurs; if the ambiguous data point is used, the update function is invoked. This has important consequences if the learner employs the strategy of adding 0.5 to the numerator when encountering an ambiguous data point. Each ambiguous data point will cause an update that will drive p_A closer to 0.5.

As an example, suppose the input stream contains 10% unambiguous A data points and 90% ambiguous data points. If the learner imposes an unambiguous data intake filter, the learner will only update p_A for 10% of the data points encountered and will always add 1 to the numerator. This results in a p_A that is significantly greater than 0.5 (though possibly still less than 1). Conversely, if the learner updates for both unambiguous and ambiguous data points, the update function is always invoked; 10% of the time, p_A is pushed closer to 1.0 but 90% of the time p_A is pushed back towards 0.5. This results in a p_A that is significantly closer to 0.5 than the p_A obtained by using only unambiguous data to update. In short, the learner is less likely to converge on the correct hypothesis, A.

2.2.3 About *t*

The update functions just derived depend on two parameters: the prior probability, $p_{A\ old}$, and the total amount of data expected during the learning period, *t*. Expecting the learner to already know the prior probability seems reasonable, as it is the most recent value the learner has calculated using the update function. Expecting the learner to already know the total amount of data during the learning period, however, may seem farfetched. Yet, the underlying concept behind *t* can also be interpreted as the amount of change a real learner's brain is allowed to undergo before settling into the final state. This would be a biologically given constraint. In my simulations, this amount is simply quantified as the total amount of data available as intake to the learner (i.e. the learner can use *t* data points of data to update the probabilities assigned to the different hypotheses).

The role of *t* in the update functions is to determine how much the probability should be shifted, given a single data point. If *t* is small, a single data point shifts the probability a great deal. This is a direct result of the fact that a small *t* means the expected data set will be small, and so only a small number of changes are allowed. Thus, the learner shifts the probability more liberally in an attempt to get to an appropriate target state before *t* runs out. Conversely, if *t* is large, a single data point shifts the probability a lesser amount. This is a direct result of the fact that a large *t* means the expected data set will be large, and so a large number of changes are allowed. The learner in this case can afford to be more conservative when shifting probability because there are more chances to shift the probability before *t* runs out.

Importantly (and perhaps surprisingly), the value of *t* is essentially arbitrary: the final probability the learner settles on is independent of the size of *t*, provided *t* is not *too* small. The reason for this stability is that the behavior of the learner is dependent on the probability distribution of the data. As long as *t* is large enough for the learner to observe a reasonably accurate sample of the probability distribution in the data intake, the learner will converge on a final probability that is the same across different values of *t*. If *t* is small, each data point has a larger impact; if *t* is large, each data point has a smaller impact. The final probability, however, does not change. This will be demonstrated with an explicit example in the chapter 3. [8]

The parameter *t* can also capture the notion of "critical period" or "period of fluctuation", where learning of particular aspects of the linguistic system ceases abruptly after some maturational point. Specifically, after the learner has encountered *t* amount of data in the intake, no more updating is possible. The probabilities for the

[8] Note that this is different from saying that that *t* must be empirically determined for each learning problem. It does not matter what *t* is for a given learning problem– the bias in the distribution is what drives the learner one way or the other. The value of *t* simply quantifies the amount a given data points alters the learner's associated probabilities for each hypothesis.

hypotheses are set, and future data points encountered have no effect. In short, the data intake for this hypothesis space is then zero, no matter what the available input is. This maps directly to the idea of a cut-off point for language learning, after which no further input can influence the learner's linguistic hypotheses.

Equipped with these relations between the period of fluctuation, *t*, and the data intake, I can speculate on the time course of parameter-setting for individual parameters. In this model, the period of fluctuation is defined by *t*: the size of *t* determines the length of the period of fluctuation. If we link *t* to the amount of change a real learner's brain is allowed to undergo and so view *t* as a biologically given constraint, we might expect that *t* should be invariant across different parameters. If all parameters have the same period of fluctuation (as defined by *t*), we should expect all parameters to be set at the same time. Yet, there is ample evidence that this is not the case. How do we reconcile this with our view of *t*?

The answer lies in the relation between *t*, data intake, and the filtering component of the learning theory. The period of fluctuation is defined by a constant value of *t*, but *t* is defined over the quantity of data points in the *intake* - not just in the available input. The proportion of input that is used as intake can vary from parameter to parameter, based on the filters used to define intake. High proportions of intake from input will allow the quantity of intake to accumulate more quickly over time; low proportions of intake from input will cause the quantity of intake to accumulate more slowly over time. The more quickly intake is accumulated over time, the faster the learner reaches the data intake limit of *t*. So, this view predicts that parameters that accumulate data intake more quickly will be set earlier than parameters that accumulate data intake more slowly.

As a concrete example, suppose learners implement an unambiguous data filter that causes the data intake to consist only of unambiguous data. The time course of parameter-setting should then depend on the quantity of unambiguous data available in the input. Yang (2004) provides a summary of evidence from experimental studies that suggests this is precisely what happens for certain syntactic parameters, including the information in table 2.1.[9] Syntactic parameters with a larger proportion of unambiguous data in the input are acquired earlier while syntactic parameters with a smaller proportion of unambiguous data in the input are acquired later.

[9] This is no longer true if the learner uses ambiguous data as well, unless the combination of unambiguous and ambiguous data used yields these same correlations. Again, one possibility is there is a correlation between the data intake (*useable* data) and the time course of acquisition. In that case, *t* would again represent the amount of change allowed, but useable ambiguous data "uses up" some of *t* (in addition to unambiguous data using up some of *t*). It is even possible that unambiguous data would use up more of *t* than useable ambiguous data would, perhaps in proportion to the amount of perceived ambiguity: the more unambiguous the data is, the more *t* is used up since the learner is more confident that the data is informative.

Parameter	Target Language	Unamb Data Frequency	Time of Acquisition
Verb-Raising[a]	French	7 %	1;8 (Pierce, 1992)
Obligatory Subject[b]	English	1.2%	3;0 (Valian, 1991)
Verb-Second[c]	German/Dutch	1.2%	3;0-3;2 (Clahsen, 1986)
Scope-Marking[d]	English	0.2%	4;0+ (Thornton & Crain, 1994)

Table 2.1. The effect of data intake accumulation on parameter-setting. Assuming an unambiguous data filter, syntactic parameters that have a higher proportion of input used as intake are the parameters that are acquired earlier.

Looking at the data in table 2.1, we can see the relation between the frequency of unambiguous data in the learner's input and the time of acquisition. We look first at Verb-Raising. In languages like French, the tensed verb moves before adverbs negation and adverbs ('*Jacques voit souvent/pas Simone*'; 'Jack sees often/not Simone'), in contrast to languages like English ('Jack often sees Simone') (1a). Unambiguous data signaling Verb-Raising comprise about 7% of the input, and children appear to have knowledge of Verb-Raising quite early.

We turn then to the Obligatory Subject. In languages like English, a subject is required ('He saw Rafael', 'It is raining'), while in languages like Spanish, the subject is optional ('*(Él) vio a Rafael', 'Llueve'*; '(He) saw Rafael', 'Rains'). Unambiguous data for Obligatory Subject is much less frequent than Verb-Raising data, and the time of acquisition is also later than that of Verb-Raising.

We can look to Verb-Second as well. In languages like German and Dutch, the tensed Verb in the main clause is moved to the second phrasal position, following one phrase of any type ('*Ich liebe die Katzen', 'Die Katzen liebe ich'; '*I-Subj love cats-Obj', 'Cats-Obj love I-Subj'). Unambiguous data for Verb-Second appear approximately as frequently as unambiguous data for Obligatory Subject, and the time of acquisition is also approximately equivalent.

There is also evidence from Scope-Marking. In German, Hindi, and other languages, long-distance *wh*-questions leave intermediate copies of *wh*-markers ('*Wer glaubst du wer Recht hat?'*, 'Who think you who right has?', Who do you think has the right?). For English children to know that English does not use this option, long distance *wh*-questions must be heard in the input, a type of data that is very infrequent in the available input to children. And indeed, the time of acquisition is much later.

In summary, children could learn from a fixed quantity of *relevant* data points, irrespective of parameter, and this would accord with experimental evidence. The quantity is constant across all parameters (*t*), but the availability of relevant data (intake) is not constant across all parameters. This yields different time courses of parameter-setting.

2.3 Summary of Bayesian Updating Adapted to a Linguistic Framework

I have now described the mathematical framework I will employ in the subsequent chapters to explore different case studies in language learning. In addition, I have sketched how values integral to the mathematical framework can be mapped to already existing concepts in the language learning literature. This framework will be the basis for the updating procedure used by the learner to shift probability between competing hypotheses. I reiterate that this updating procedure is domain-general, and is applicable across linguistic domains (and other cognitive domains). However, the representations assumed for the hypothesis space and the filters tested in each case study will be domain-specific. The separation of a learning theory into three distinct parts allows us to merge domain-specific components with domain-general components and thus have a theory that is both.