Programming Language Support for Probabilistic Associative Memory

Dept. of CIS - Senior Design 2014-2015*

Fan Yin
fanyin@seas.upenn.edu

Haolin Lu
haolinlu@seas.upenn.edu

Yukuan Zhang
yukuan@seas.upenn.edu

ABSTRACT

Our goal is to create a library (OPaml) that provides an interface and implementation for programming with a probablistic associative memory.

An associative memory allows queries by content rather than by address or location. In addition, as with standard memory, associative memory remembers every value stored.

On the other hand, probabilistic associative memory returns a value based on a probability model for a given memory query. This allows it to use less space than regular associative memories, making it useful for applications that prioritize low memory footprints over fidelity.

1. INTRODUCTION

The standard way to access a stored value in memory is by address. This involves finding the physical position of that address in memory and returning the stored value. Associative memory allows searching for a values based on the stored content. An analagous structure in programming languages is the hashtable, which returns an output value based on an input key.

Associative memories are useful for when one needs to query memory based on some input data or tag. For instance, router firmware will often employ associative memories to store tables of IP addresses, which serve as the "next hop" for incoming packets. Since routers looking for the "next hop" have a specific, standard query format – e.g. an IP address of the form www.xxx.yyy.zzz – it makes sense to store the routing tables based around this input string.

With proabilistic memories, for a given query, we return the most "likely" candidate based on some specified probability model of the values stored in memory. Since our queries work directly with a probability model, it would offer more fluidity with programming that involves statistical modeling or simulations. Returning values based on a probability also saves memory because not every value has to be stored. The memory mechanism can extrapolate or interpolate a response to a query based on values previously inserted into memory.

We also strive to add support for structured data in our probabilistic associative memory. In a standard associative memory scheme, a similarity query is done by comparing input bits to the values stored in memory. However, this does not allow for good similarity comparisons between trees or lists, as the memory is not aware of the underlying structure. As a result, many applications on structured data will first flatten the structure, do the calculations it needs, then reconstruct the original structure of that data. With a built-in type system that supports structured data, the user will be able to directly manipulate the data in memory while keeping the structure intact.

- Types and Structured Data – more easily determine similarity queries. Data structure aware of structured data that it's holding.

2. RELATED WORK

Perhaps the most important section of your proposal is *related work*. Here you demonstrate that you have read and understand what others in the field have done. This ensures you (1) know the state-of-the-art, (2) are not re-doing others work, and (3) you know the performance levels you must achieve to make a contribution. As you discuss each related work, make note of how each has advanced the field. Keep in mind that this section should not read like a regular research paper you write for other classes. In other words, you should not just discuss related work for the sake of having a related work section; rather, tell a story about the state-of-the-art of the field and where your work fits in.

This section should have in-line citations to your bibliography (really all sections should have citations, but we expect them to be most dense in this section). We are going to require that your proposal has at least 6 references. Fortunately, LaTeX makes citations easy. Your TA has had no difficulty, as the work of Ivanov *et al.* [2] demonstrates. Need help with LaTeX? Be sure to check out [3] and [1], two helpful on-line resources.

What defines a good resource? Wikipedia is **NOT** a good resource. We would like to see references from academic journals/conferences (ACM, IEEE, etc.). We realize not everyone is doing pure research and for students with 'implementation' projects such sources may be rare. No matter the case, your sources need to be reputable.

Let us return to your factorization proposal. You should put out the earliest related work; naïve methods like trial divison and the Sieve of Eratosthenes, but state they are of no modern relevance. Then discuss modern methods like the Quadratic Sieve and General Number Field Sieve. Note the humongous time and memory bounds of these algorithms. But wait! You are going to propose a better way . . .

3. PROJECT PROPOSAL

Now is the time to introduce your proposed project in all of its glory. Admittedly, this is not the easiest since you probably have not done much actual research yet. Even so,

*Advisor: Steve A. Zdancewic (stevez@cis.upenn.edu).

setting and realizing realistic research goals is an important skill. Begin by summarizing what you are going to do and the expected benefit it will bring.

3.1 Anticipated Approach

Having summarized *what* you are going to do, its time to describe *how* you plan to do it. Our factorization example does not work so well here (it is likely impossible to realize) – so let us suppose you are going to create a service that takes a cell-phone picture of a building and returns via text-message, the name of that building[1].

In this case you might want to talk about establishing a server to receive pictures via MMS. Once the picture is received, you will run an edge extraction algorithm over it. Then, similarity between the submitted picture and those stored (and tagged) in a MySQL database will be computing using algorithm XYZ. Finally, the tag of the most similar image will be returned to the user. Do not bore the reader with trivial details, but give them an overview; a block-flow diagram would prove helpful (and is required).

3.2 Technical Challenges

In this subsection note where you anticipate having novel difficulty. Maybe you have never setup a MySQL database or even used SQL before at all – yes, that is a challenge – but not one readers care about. More novel would be the fact that many buildings on Penn's campus look similar and your classifier may be inaccurate in such instances. The purpose of this section is two-fold: 1) you will think about which parts of your project would require the most time and effort and 2) you will convince the reader that this is a project worth undertaking.

3.3 Evaluation Criteria

Suppose you have implemented your approach and it is functioning. Now how are you going to convince readers your approach is better than what exists? In the factorization example, you could just compare run-times between algorithms run on the same input. The image recognition example might use a percentage of accurate classifications. Other fields may have established testing benchmarks.

No matter the case, you need to prove you have contributed to the field. This will be easier for some than others. In particular, those with 'sensory' projects involving visual or sonic elements need to think this point through – objective measures are always better than subjective ones.

4. RESEARCH TIMELINE

Finally, we would like you to speculate about the pace of your research progress. This section need not be lengthy, we would just like you to specify some milestones so we can gauge your progress during our intermediate interviews. Let us follow through with our image recognition example:

- ALREADY COMPLETED: Preliminary reading. Began implementation of image-recognition algorithm.
- PRIOR-TO THANKSGIVING : Photograph buildings for DB. Make algorithm more efficient, tune parameters.
- PRIOR-TO CHRISTMAS : Create server-MMS interface. Expand tagged DB collection.
- COMPLETION TASKS : Verify implementation is bug-free. Conduct accuracy testing. Complete write-up.
- IF THERE'S TIME : Investigate image pre-processing techniques to improve accuracy.

[1] Do not use this idea – someone did it in a previous year.

5. REFERENCES

[1] The Comprehensive TeX Archive Network (CTAN). A (not so) short introduction to LaTeX2e. `http://www.ctan.org/tex-archive/info/lshort/english/`.

[2] Radoslav Ivanov, Miroslav Pajic, and Insup Lee. Attack-resilient sensor fusion. In *DATE'14: Design, Automation and Test in Europe*, 2014.

[3] Wikibooks. LaTeX. `http://en.wikibooks.org/wiki/LaTeX`. Note: Students should not cite Wikis!

APPENDIX

A. OTHER SPECIFICS

Your proposal need not have appendices like this section and the next but we still have info to share:

1. PROPOSAL LENGTH: We require that your proposal be 4–5 pages in length, bibliography included. Be careful, LaTeX and our style-file in particular are *extremely* space efficient. An 9-page MS-Word document could easily become a 5-page LaTeX one.
2. PLAGARISM: **DO NOT** plagarize. If you are caught, you will fail the class (*i.e.*, not graduate), or worse.

B. LaTeX EXAMPLES

At this point, the proposal specification is complete. From here on out, we are just going to show off some commonly used LaTeX technique. Be sure to look at the 'code behind' and see Tab. 1, Eqn. 1 and Fig. 1 for the output! Keep in mind that the appendix is usually not a good place for your figures. Place them where you need them and remember to refer to them in the body of your text; otherwise, the reader will keep reading and will miss them!

$$M(p) = \int_0^\infty (1 + \alpha x)^{-\gamma} x^{p-1} dx \tag{1}$$

User Type	Cleanup%	Honesty%
Good	90-100%	100%
Purely Malicous	0-10%	0%
Malicious Provider	0-10%	100%
Feedback Malicous	90-100%	0%
Disguised Malicous	50-100%	50-100%
Sybil Attacker	0-10%	Irrelevant

Table 1: Example Table

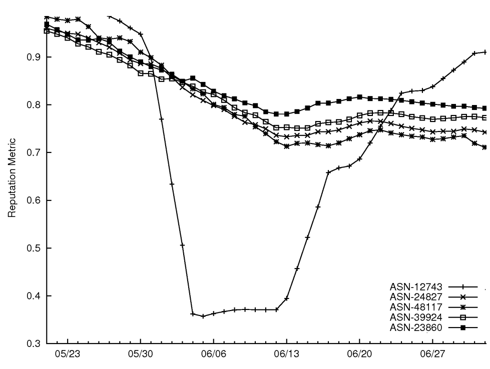

Figure 1: Example Figure/Graph