Writing Scientific Papers and Software

Cheng Soon Ong
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—A critical part of scientific discovery is the communication of research findings to peers or the general public. Mastery of the process of scientific communication improves the visibility and impact of research. While this guide is a necessary tool for learning how to write in a manner suitable for publication at a scientific venue, it is by no means sufficient, on its own, to make its reader an accomplished writer. This guide should be a starting point for further development of writing skills.**

I. Introduction

The aim of writing a paper is to infect the mind of your reader with the brilliance of your idea [1]. The hope is that after reading your paper, the audience will be convinced to try out your idea. In other words, it is the medium to transport the idea from your head to your reader's head. In the following section, we show a common structure of scientific papers and briefly outline some tips for writing good papers in Section **??**.

At that point, it is important that the reader is able to reproduce your work [2], [3], [4]. This is why it is also important that if the work has a computational component, the software associated with producing the results are also made available in a useful form. Several guidelines for making your user's experience with your software as painless as possible is given in Section **??**.

This brief guide is by no means sufficient, on its own, to make its reader an accomplished writer. The reader is urged to use the references to further improve his or her writing skills.

II. Models and Methods

Describe your idea and how it was implemented to solve the problem. Survey the related work, giving credit where credit is due.

III. Results

Show evidence to support your claims made in the introduction.

IV. Discussion

Discuss the strengths and weaknesses of your approach, based on the results. Point out the implications of your novel idea on the application concerned.

V. Summary

Summarize your contributions in light of the new results.

VI. The Post-Processing Phase

In the previous step we have obtained, for each pixel (i, j), an estimate $p_{ij} \in [0, 1]$ of whether this pixel of the (high-resolution) ground truth is white. We interpret this number as a probability. From this, we would like to output a black or white value for each 16×16 patch; we call this process *rounding* or *post-processing*.

A. The simple algorithm

A simple method is just to choose a threshold t (typically $t = 0.25$ since this is how the ground truth is generated) and, for each patch, take the mean of the estimated values $m = \frac{\sum p_{ij}}{16 \cdot 16}$. Then, output white iff $m > t$. The parameter t can be optimized to get good validation scores.

B. Integer Programming

We use a more involved approach. First, for each patch, we estimate its likelihood $\ell \in [0, 1]$. Intuitively, we want it to satisfy that if all pixels have $p \approx 0.25$, then $\ell \approx 0.5$ (since then we are not sure whether to take the patch). The same should hold if we have a 0.25 fraction of pixels with $p = 1$ and the rest have $p = 0$.

One method to do so is to compute the mean m and then set $\ell(m)$ to be a piecewise linear function with $\ell(0) = 0, \ell(t) = 0.5, \ell(1) = 1$.

Another one is to assume that each pixel is an *independent* p-biased coin flip. Then, we can use dynamic programming to compute, for each $k \in \{0, ...16 \cdot 16\}$ the exact probability that k pixels are white. We take the probability that the number of white pixels is above the threshold.

In practice, we choose the first method because of its simplicity and speed.[1]

Now that we have the likelihood ℓ, we assume that patches are independent. We would like to pick the maximum-likelihood set of patches.

Let V be the set of patches indexed by v; thus we want:

$$\text{argmax}_{X \subseteq V} \ell(X) = \text{argmax}_{X \subseteq V} \prod_{v \in X} \ell_v \prod_{v \in V \setminus X} (1 - \ell_v)$$

taking logarithms:

$$\begin{aligned}
\text{argmax}_{X \subseteq V} \log \ell(X) &= \text{argmax}_{X \subseteq V} \sum_{v \in X} \log \ell_v + \sum_{v \in V \setminus X} \log(1 - \ell_v) \\
&= \text{argmax}_{X \subseteq V} \sum_{v \in V} \log(1 - \ell_v) + \sum_{v \in X} \log(\ell/(1 - \ell_v)) \\
&= \text{argmax}_{X \subseteq V} \sum_{v \in X} \log(\ell/(1 - \ell_v))
\end{aligned}$$

[1]The dynamic programming uses about 2-3 seconds per image, which makes it difficult to try many hyperparameters.

ACKNOWLEDGEMENTS

REFERENCES

[1] S. P. Jones, "How to write a great research paper," 2008, microsoft Research Cambridge.

[2] M. Schwab, M. Karrenbach, and J. Claerbout, "Making scientific computations reproducible," *Computing in Science and Engg.*, vol. 2, no. 6, pp. 61–67, 2000.

[3] J. B. Buckheit and D. L. Donoho, "Wavelab and reproducible research," Stanford University, Tech. Rep., 2009.

[4] R. Gentleman, "Reproducible research: A bioinformatics case study," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005. [Online]. Available: http://www.bepress.com/sagmb/vol4/iss1/art2