Continualization of Probabilistic Programs With Correction

Jacob Laurel[1] and Sasa Misailovic[1]

University of Illinois Urbana-Champaign, Urbana, Illinois 61820, USA
{jlaurel2,misailo}@illinois.edu

1 Appendix

1.1 Programs

Below we show all benchmark programs. Only the model is shown, the remainder of the program is simply an array with the data points and a for loop iterating over each one to update the posterior. In these models the prior is denoted as `param`.

```
param := Uniform(250,350);
p := Beta(277,param);
n := 4000;
np := (n*p);
half := (0.5*n);
Votes := Binomial(n,p);
if (Votes > half){
        win := 1;
}
else {
        win := 0;
};
return win;
```

Election: A model of an election. Taken from [4]. We continualize the Binomial to a Gaussian, and infer the scale parameter, describing the probability of winning.

```
param := DiscUniform(35,55);
ethnicity := Normal(0,10);
colRank := Binomial(param,0.6);
yExp := Binomial(20,0.5);
if (ethnicity > 10){
        colRank := (colRank + 5);
}
expRank := (yExp - colRank);
if ((colRank <= 5) or (expRank > -5)) {
    hire := 1;
}
else {
    hire := 0;
};
return hire;
```

Fairness [1]: A decision making program that chooses whether to hire an individual sampled from a given population distribution. We continualize the Binomial to Gaussian (which was used in the original).

```
education_num := Binomial(30,0.33); // prior on education level
age := Normal(38.5816, 186.0614);
sex := Bernoulli(0.33);
capital_gain := Normal(1077.6488, 54542539.1784);
capital_loss := Normal(87.3038, 162376.9378);
hours_per_week := Normal(40.4374, 152.4589);

N_age := ((age − 17.0) / 62.0);
N_education_num := ((education_num − 3.0) / 13.0);
N_capital_gain := ((capital_gain − 0.0) / 22040.0);
N_capital_loss := ((capital_loss − 0.0) / 1258.0);
N_hours_per_week := (hours_per_week − 4.0) / 73.0);

t := (−0.0106 ∗ N_age) + (−0.0194 ∗ N_education_num) + (−5.7412 ∗ N_capital_gain)
    + (0.0086 ∗ N_capital_loss) + (−0.0123∗N_hours_per_week) + 1.0257;

if (sex > 0) {
     t := t + −0.0043;
}
return t;
```

Fairness SVM Example [1]. An SVM classifier

```
skillA := Poisson(100); //poisson prior on skill
skillB := 100;
perfA1 := Binomial(skillA,0.9);
perfB1 := Binomial(skillB,0.9);
if (perfA1 > perfB1) {
    res := 1;
}
else {
    res := 0;
};
return res;
```

TrueSkill [5]: A popular standard rating system for online gaming matches. We continualize the Poisson to a Gaussian (which was used in the original) and estimate the player's skill given multiple matches.

```
param := Uniform(0,10) //prior on position
which := Uniform(0,1);
if (which < 0.9){
    x := Normal(param,1);
}
else {
    if (which < 0.95){
        x := Uniform(0,10);
    }
    else {
        x := 10;
    };
};
return x;
```

SVE [7]: A Discrete-Continuous mixture modeling a robot localizing itself from observed sensor readings, hence we estimate the position which has a uniform prior. We continualize the constant (the original had a near-constant triangular distribution).

```
param := Beta(1,1)
successes := Binomial(100,param);
return successes;
```

Rate Inference [6]: A Beta Binomial from the Cognitive Science literature, where we must infer the success rate given the number of trials.

```
param := DiscUniform(500,1500);
NumMet := Binomial(param,0.5);
NumInfected := Binomial(NumMet,0.3);
return NumInfected;
```

Discrete Disease Model: a chain double Binomial model (based on [2]) of an infectious disease spread, where we estimate the amount of contacts having observed the number of sick patients.

```
param := DiscUniform(10,50)
PlanktonCount := Binomial(param,0.5);
return PlanktonCount;
```

Plankton: An ecological model taken directly from [3] that highlights the difficulties in estimating n for discrete models of plankton populations.

```
param := Uniform(0.5,1) //prior on the probability of getting question right
percentile := Uniform(0,1);
if (percentile > 0.5) //top half of the class
    k := Binomial(20,param)
else:
    k := Binomial(20,0.5)    //0.5 is guessing probability
return k;
```

Exam: Another Cognitive Science model taken from [6] that has a latent mixture of students (those who studied vs. those who did not) and a different probability of getting a question right for each group (the group who did not study simply has 0.5 chance of getting a T/F question correct, while the group who studied could have their probability lie between 0.5 and 1)

References

1. Albarghouthi, A., D'Antoni, L., Drews, S., Nori, A.V.: Fairsquare: Probabilistic verification of program fairness. Proc. ACM Program. Lang. **1**(OOPSLA), 80:1–80:30 (Oct 2017)

2. Becker, N.: A general chain binomial model for infectious diseases. Biometrics **37**(2), 251–258 (1981)
3. Blumenthal, S., Dahiya, R.C.: Estimating the binomial parameter n. Journal of the American Statistical Association **76**(376), 903–909 (1981)
4. Goodman, N.D., Tenenbaum, J.B., Contributors, T.P.: Probabilistic Models of Cognition (2016)
5. Herbrich, R., Minka, T., Graepel, T.: Trueskill$^{\text{TM}}$: A bayesian skill rating system. In: Proceedings of the 19th International Conference on Neural Information Processing Systems. pp. 569–576. NIPS'06 (2006)
6. Lee, M.D., Wagenmakers, E.J.: Bayesian cognitive modeling: A practical course. Cambridge university press (2014)
7. Sanner, S., Abbasnejad, E.: Symbolic variable elimination for discrete and continuous graphical models. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. pp. 1954–1960. AAAI'12 (2012)