Automating variational inference for statistics and data mining

Tom Minka
Machine Learning and Perception Group
Microsoft Research Cambridge

IMPS 2009
A common situation

- You have a dataset
- Some models in mind
- Want to fit many different models to the data
Model-based psychometrics

\[ y_{ij} \sim f(y | \alpha_i, \beta_j, \theta) \]

- Subjects \( i = 1, \ldots, N \)
- Questions \( j = 1, \ldots, J \)
- \( \alpha_i = \) subject effect
- \( \beta_j = \) question effect
- \( \theta = \) other parameters
The problem

• Inference code is difficult to write
• As a result:
  – Only a few models can be tried
  – Code runs too slow for real datasets
  – Only use models with available code
• How to get out of this dilemma?
Infer.NET: An inference compiler

• You specify a statistical model
• It produces efficient code to fit the model to data
• Multiple inference algorithms available:
  – Variational message passing
  – Expectation propagation
  – Gibbs sampling (coming soon)
• User extensible
Infer.NET: An inference compiler

• A compiler, not an application
• Model can be written in any .NET language (C++, C#, Python, Basic, …)
  – Can use data structures, functions of the parent language (jagged arrays, if statements, …)
• Generated inference code can be embedded in a larger program
• Freely available at:

  http://research.microsoft.com/infernet
Papers using Infer.NET


• David Stern, Ralf Herbrich, Thore Graepel, “Matchbox: Large Scale Online Bayesian Recommendations”, WWW 2009

Variational Bayesian inference

• True posterior is approximated by a simpler distribution (Gaussian, Gamma, Beta, ...)  
  – “Point-estimate plus uncertainty”  
  – Halfway between maximum-likelihood and sampling
Variational Bayesian inference

• Let variables be \( x_1, \ldots, x_V \)

• For each \( x_v \), pick an approximating family \( q(x_v) \) (Gaussian, Gamma, Beta, \( \ldots \))

• Find the joint distribution \( q(x) = \prod_v q(x_v) \) that minimizes the divergence

\[
KL(q(x) \parallel p(x \mid data))
\]

(or other error measure)
Variational Bayesian inference

• Well-suited to large datasets, sequential processing (in style of Kalman filter)
• Provides Bayesian model score
Implementation

- Convert model into factor graph
- Pass messages on the graph until convergence

$$p(y | x) = p(y_1 | x_1, x_2) p(y_2 | x_1, x_2)$$
Further reading


Example: Cognitive Diagnosis Models (DINA,NIDA)

• $y_{ij} = 1$ if student $i$ answered question $j$ correctly (observed)
• if question $j$ requires skill $k$ (known)
• $q_{jk} = 1$ if student $i$ has skill $k$ (latent)

$hasSkill_{ik} = 1$
$hasSkill_{ik} \sim Bernoulli (pSkill_k)$

**DINA model:** $K+2J$ parameters

$hasSkills_{ij} = \prod_k hasSkill_{ik}^{q_{jk}}$ (student possesses all skills for question)

$p(y_{ij} = 1) = (1 - slip_j)^{hasSkills_{ij}} guess_j^{1-hasSkills_{ij}}$

**NIDA model:** $K+2K$ parameters

$exhibitsSk ill_{ik} = (1 - slip_k)^{hasSkill_{ik}} guess_k^{1-hasSkill_{ik}}$

$p(y_{ij} = 1) = \prod_k exhibitsSk ill_{ik}^{q_{jk}}$
Graphical model
(per student)

Linkage depends on the Q matrix
Prior work

• Junker & Sijtsma (2001), Anozie & Junker (2003) found that MCMC was effective but slow to converge
• Ayers, Nugent & Dean (2008) proposed clustering as fast alternative to DINA model
• What about variational inference?
DINA, NIDA models in Infer.NET

• Each model is approx 50 lines of code
• Tested on synthetic data generated from the models
  – 100 students, 100 questions, 10 skills
  – Random question-skill matrix
  – Each question required at least 2 skills
• Infer.NET used Expectation Propagation (EP) with Beta distributions for parameter posteriors
  – Variational Message Passing gave similar results on DINA, couldn’t be applied to NIDA
Comparison to BUGS

• EP results compared to 20,000 samples from BUGS

• For estimating posterior means, EP is as accurate as 10,000 samples, for same cost as 100 samples
  – i.e. 100x faster
DINA model on DINA data

Gibbs
EP (Infer.NET)
Truth
NIDA model on NIDA data

Gibbs
EP (Infer.NET)
Truth
Model selection

- Correct generative model is chosen in each case
- DINA model is better at fitting NIDA data than vice versa
Code for DINA model

```csharp
using (Variable.ForEach(student)) {
    using (Variable.ForEach(question)) {
        VariableArray<bool> hasSkills = Variable.Subarray(hasSkill[student], skillsRequiredForQuestion[question]);
        Variable<bool> hasAllSkills = Variable.AllTrue(hasSkills);
        using (Variable.If(hasAllSkills)) {
            responses[student][question] = !Variable.Bernoulli(slip[question]);
        }
        using (VariableIfNeeded(hasAllSkills)) {
            responses[student][question] = Variable.Bernoulli(guess[question]);
        }
    }
}
```
Code for NIDA model

```csharp
using (Variable.ForEach(skillForQuestion)) {
    using (Variable.If(hasSkills[skillForQuestion])) {
        showsSkill[skillForQuestion] = !Variable.Bernoulli(slipSkill[skillForQuestion]);
    }
    using (Variable.IfNot(hasSkills[skillForQuestion])) {
        showsSkill[skillForQuestion] = Variable.Bernoulli(guessSkill[skillForQuestion]);
    }
}
responses[student][question] = Variable.AllTrue(showsSkill);
```
Example: Latent class models for diary data

Diary data

• Patients assess their emotional state over time (Rijmen et al 2008, PMKA)
• \( y_{itj} = 1 \) if subject \( i \) at time \( t \) feels emotion \( j \) (observed)

Basic Hidden Markov model:

• \( z_{it} \in \{1, \ldots, S\} \) is hidden state of subject \( i \) at time \( t \) (latent)

\[ z1 \rightarrow z2 \rightarrow z3 \rightarrow z4 \]

\( S^2 \) transition parameters

\( JS \) observation parameters
Prior work

• Rijmen et al (2008) used maximum-likelihood estimation of HMM parameters
  – model selection was an open issue
• Which model gets highest score from variational Bayes?
HMM in Infer.NET

- Model is approx 70 lines of code
- Can vary:
  - number of latent classes (S)
  - whether states are independent or Markov

![Graph showing log p(Data) vs Number of latent classes]

Best model is Markov with 12 latent classes
Hierarchical HMM

- Real data has more structure than HMM
- 32 subjects were observed over 7 days, having 9 observations per day
  - Basic HMM treated each day independently
- Rijmen et al (2008) proposed switching between different HMMs on different days (hierarchical HMM)
  - more model selection issues
Hierarchical HMM in Infer.NET

• Model is approx 100 lines of code
• Can additionally vary:
  – number of HMMs (1,3,5,7,9)
  – whether days are independent or Markov
  – whether transition params depend on day
  – whether observation params depend on day
• Best model among 400 combinations (2 hours using VMP):
  – 5 HMMs, each having 5 latent states
  – Observation params depend on day, but transition params do not
Summary

- Infer.NET allowed 4 custom models to be implemented in a short amount of time
- Resulting code was efficient enough to process large datasets, compare many models
- Variational inference is potential replacement for sampling in DINA, NIDA models

http://research.microsoft.com/infernet
Acknowledgements

• Rest of Infer.NET team:
  – John Winn, John Guiver, Anitha Kannan
• Beth Ayers, Brian Junker (DINA,NIDA models)
• Frank Rijmen (Diary data)