Modeling Protein-Protein Interactions in Biomedical Abstracts with Latent Dirichlet Allocation

David Andrzejewski
CS 838 - Final Project

June 14, 2006

Abstract

A major goal in biomedical text processing is the automatic extraction of protein interaction information from scientific articles or abstracts. We approach this task with a topic-based generative model. Under the model, sentences in biomedical abstracts can be generated by either an 'interaction' topic if they contain or discuss interacting proteins or a 'background' topic otherwise. This structure is implemented as a Latent Dirichlet Allocation (LDA) model. The model structure was previously developed as part of work with Mark Craven and Jerry Zhu. During this project, parameter inference equations and algorithms were derived. Future work will consist of implementation and experimental testing.

1 Introduction

Proteins are biomolecules made up of amino acids which occupy a central role in cellular biology. After water, they make up the next highest proportion of cellular weight [8]. Interactions between proteins are very important in many vital biological processes. Because of this, protein-protein interaction information can be very useful for both biological scientists and computational systems designed to analyze biological data. This information can be found in a structured format in protein-protein interaction databases like the Database of Interacting Proteins (DIP) [10]. These databases are populated by human readers who read the relevant research articles and then enter the interaction data into the database. This manual entry step can be a severe bottleneck in such a system, especially given the explosive growth of the biosciences literature. The total number of articles indexed by Medline, for example, has been growing exponentially, adding an average of 1800 new articles per day in 2005 [5].

This situation motivates the need for tools for assist in the extraction protein-protein interaction information from the scientific literature. An early approach by the DIP team used discriminating words to identify Medline abstracts that were likely to discuss protein interactions [7]. Human curators could then focus their attentions on these high-scoring articles, yielding a more efficient use of valuable human time. More ambitious

systems aim to directly extract interacting pairs from text, often using rule or pattern-based approaches [4]. Our system is designed to automatically extract interacting pairs using a 'bag of words' approach with sentence-level topics.

2 The model

The central feature of the model is the sentence-level topic model. Each sentence is considered to be generated by either an 'interaction' topic or a 'background' topic. Each topic is associated with a different 'bag of words' multinomial. Furthermore, each interaction sentence is associated with exactly one pair of proteins. Words in an interaction sentence can either be drawn from the interaction word bag or a 'protein pair bag' which contains all possible identifiers for each of the two proteins. This is necessary because there may be multiple identifiers which refer to the same protein.

The p value represents the probability of a sentence in the abstract having the interaction topic. For each document, a new p value is generated from a Dirichlet distribution. This allows the proportion of interacting sentences to vary between different documents. This flexibility should be valuable for modeling the abstracts of different types of articles, such as articles that are primarily concerned with protein-protein interactions or those that mention them only in passing (if at all). The use of a multinomial over latent topics whose parameters are themselves generated by a Dirichlet distribution is characteristic of the Latent Dirichlet Allocation (LDA) model [2]. The general outline of our generative model is as follows

for each doc d in our corpus C
  p_d =dir(α)
    for each sentence ℓ
      r_ℓ =mult(p_d)
      t_ℓ =mult(θ)
      for each word k
        $s_{\ell k}$ = bern(μ)
        if $r_{\ell k} = 0$
          $w_{\ell k}$ =mult(β_0)
        else if $r_{\ell k} = 1$
          if $s = 0$
            $w_{\ell k}$ =mult(β_1)
          else if $s = 1$
            $w_{\ell k}$ =pair(t_ℓ)

The model parameters and variables are

- θ = The protein pair selection multinomial
  (θ_t = probability of selecting protein pair t)
- α = Dirichlet hyperparameter for topic selection variable

- β = Word bags for 'interaction' and 'background' topics
  (β_{jw} = probability of word w under topic j)
- μ = Probability for pair switch variable in 'interaction' sentences
  ($\mu_0 = P(s = 0)$)
- r = Topic switch variable
  ($r = 0$ means background topic, $r = 1$ means interaction topic)
- s = Pair switch variable for 'interaction' sentences
  ($s = 1$ means select protein, else $s = 0$ means select from word bag)
- t = Protein pair switch variable
  (specifies a pair of proteins)
- w = The observed words
  ($w_{d\ell k}$ is the word in document d, sentence ℓ, word k)

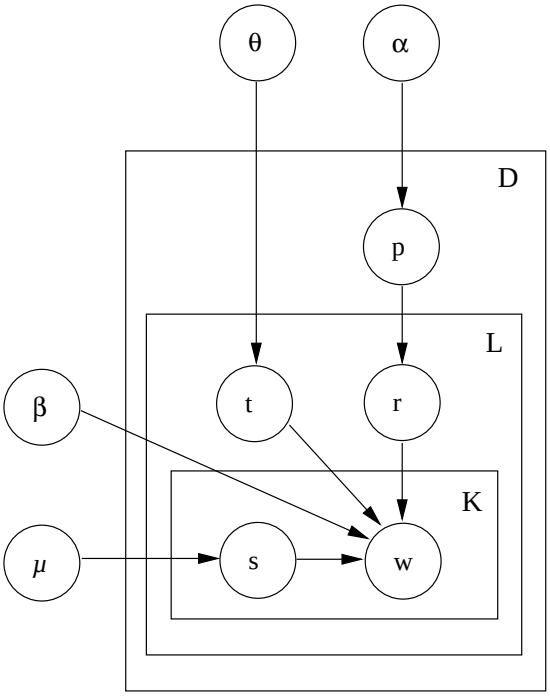

Figure 1: Graphical representation of the model.

3 Parameter estimation

3.1 Document likelihood

The parameters of our model are α, θ, β, and μ. The hidden variables are p, t, r, and s, and the only observed values are the actual words w. The log likelihood of a single document can be obtained by marginalizing over the hidden variables

$$\begin{aligned}\log(P(d|\theta,\alpha,\beta,\mu)) = \int_p \sum_\ell \sum_{(r_\ell,t_\ell)} \sum_k \sum_s & \log(P(w_{\ell k}|r_\ell,t_\ell,s,\beta)) \\ & + \log(P(r_\ell|p)) \\ & + \log(P(p|\alpha)) \\ & + \log(P(t_\ell|\theta)) \\ & + \log(P(s|\mu))dp\end{aligned}$$

Where ℓ and k are indices into sentences and words within sentences, respectively. $P(w_{\ell k}|r_\ell,t_\ell,s_{\ell k},\beta,\theta)$ uses indicator functions of t_ℓ, r_ℓ, and $s_{\ell k}$ to model the probability of a single word

$$P(w|\beta,t,s,r) = \mathbf{1}_{r==0}\beta_{0w} + \mathbf{1}_{r==1}(\mathbf{1}_{s==0}\beta_{1w} + \mathbf{1}_{s==1}y_t(w))$$

$y_t(w)$ represents the protein pair specified by the switch variable t.

$$y_t(w) = \begin{cases} 1 & w \in \sigma_t \\ 0 & \text{else} \end{cases}$$

where σ_t is the set of identifiers for the two proteins in the protein-protein pair specified by the index t.

3.2 Corpus likelihood

In order to determine the log likelihood of a corpus C, we assume the documents to be independent of one another. This allows us to simply extend the above equation by summing over all training documents.

$$\log(P(C|\theta,\alpha,\beta,\mu)) = \sum_{d\in C} \log(P(d|\theta,\alpha,\beta,\mu))$$

3.3 Variational EM

To set the parameters for our model, we want to find the parameter values that maximize the log likelihood of the training corpus (maximum likelihood estimation or MLE). We cannot directly optimize by taking the derivatives with respect to the parameters and setting them to zero, due to the presence of the hidden variables. Furthermore, we cannot use a standard expectation maximization (EM) approach because the posterior

distribution of the hidden variables in our model is intractable due to coupling issues [3].

We overcome this problem by restricting the form of the auxiliary distribution to a family of fully factorized distributions parameterized by a new set of parameters. This approach is known as *variational inference* and the new parameters are referred to as the *variational parameters* [6, 2]. The restricted family of fully factorized distributions we will use is

$$q(p, r_{1:L}|\gamma, \phi_{1:L}) = q(p|\gamma)\prod_{\ell=1}^{L} q(r_\ell|\phi_\ell)$$

$\phi_{\ell j}$ is the probability that $r_\ell = j$ where $j \in 0, 1$. γ_0 and γ_1 are the parameters of the beta distribution $q(p|\gamma)$.

This new distribution is the centerpiece of our variational EM scheme. We start by fixing the model parameters and multiplying $P(C|\alpha, \theta, \beta, \mu)$ by $\frac{q(p,r_{1:L}|\gamma,\phi_{1:L})}{q(p,r_{1:L}|\gamma,\phi_{1:L})}$. Just as in standard EM, we then use Jensen's inequality to obtain a lower bound on the log likelihood by pulling a $q(p, r_{1:L}|\gamma, \phi_{1:L})$ term out of the log.

$$\log(P(C|\alpha, \theta, \beta, \mu)) \geq F(q, \Theta)$$

$$F(q, \Theta) = \int_p \sum_{r_{1:L}} q(p, r_{1:L}|\gamma, \phi_{1:L}) \log(P(p|\alpha)P(r_{1:L}|p)\frac{P(C|r_{1:L}, \theta, \beta, \mu)}{q(p, r_{1:L}|\gamma, \phi_{1:L})})dp$$

(Θ is shorthand notation for the model parameters θ, β, α, and μ.)

Since $q(p, r_{1:L}|\gamma, \phi)$ is a valid probability distribution and we are marginalizing over p and $r_{1:L}$, this expression is equal to the expectation over q of the log expression.

$$F(q, \Theta) = \mathrm{E}_q(\log(P(p|\alpha)P(r_{1:L}|p)\frac{P(C|r_{1:L}, \theta, \beta, \mu)}{q(p, r_{1:L}|\gamma, \phi_{1:L})}))$$

3.3.1 E-step

We want to maximize this lower bound. First, we do a variational E-step and optimize with respect to the variational parameters γ and $\phi_{1:L}$. To do this we first must expand out the lower bound equation, first by rewriting the log of a product as a sum of logs, and then by rewriting the expectation of a sum as a sum of expectations.

$$F(q,\Theta) = \mathrm{E}_q(\log(P(p|\alpha)) + \log(P(r_{1:L}|p)) + \log(P(C|r_{1:L},\theta,\beta,\mu)) - \log(q(p,r_{1:L}|\gamma,\phi_{1:L})))$$

$$\mathrm{E}_q(\log(P(p|\alpha))) = \log(\Gamma(\sum_i \alpha_i)) - \sum_i \log(\Gamma(\alpha_i)) + \sum_i \left((\alpha_i - 1)(\psi(\gamma_i) - \psi(\sum_j \gamma_j)) \right)$$

$$\mathrm{E}_q(\log(P(r_{1:L}|p))) = \sum_{(\ell,i)} \phi_{\ell i}(\psi(\gamma_i) - \psi(\sum_j \gamma_j))$$

$$\mathrm{E}_q(\log(P(C|r_{1:L},\theta,\beta,\mu))) = \sum_{d \in C} \sum_{(\ell,k)} \sum_{t_\ell s_{\ell k}} \left(\log(P(t_\ell|\theta)) + \log(P(s_{\ell k}|\mu)) + \sum_j (\phi_{\ell j} P(w_{d\ell k}|r_{\ell j}, t_\ell, \beta, s)) \right)$$

$$\mathrm{E}_q(\log(q(p,r_{1:L}|\gamma,\phi_{1:L}))) = \sum_{(\ell,i)} \phi_{\ell i} \log(\phi_{\ell i})$$

ψ is the digamma function, which is equal to the first derivative of the log of the gamma function.

$$\psi(\alpha_i) = \frac{\partial}{\partial \alpha_i}(\log(\Gamma(\alpha_i)))$$

Also, the above derivations calculate the expectation of the log of a Dirichlet random variable p using this formula [2]

$$\mathrm{E}(\log(p|\alpha)) = \psi(\alpha_i) - \psi(\sum_j \alpha_j)$$

We can now directly optimize the lower bound with respect to the variational parameters by simply taking their derivatives and setting them to zero, and using Lagrange multipliers where necessary.

$$\frac{\partial F}{\partial \phi_{\ell i}} = \psi(\gamma_i) - \psi(\sum_j \gamma_j) + \sum_{d \in C} \sum_{t_\ell} \sum_k \sum_{s_{\ell k}} \log(P(w_{d\ell k}|r_{\ell i}, t_\ell, \beta, s_{\ell k})) - \log(\phi_{\ell i}) + 1 + \lambda_\ell$$

$$\frac{\partial F}{\partial \gamma_i} = \psi(\gamma_i)(\alpha_i + \sum_\ell \phi_{\ell i} - \gamma_i) - \psi(\sum_j \gamma_j)(\alpha_i + \sum_\ell \phi_{\ell i} - \gamma_i)$$

These equations can be set to zero and used to solve for the variational parameter values.

$$\phi_{\ell i} \propto \exp(\psi(\gamma_i) - \psi(\sum_j \gamma_j) + \sum_{d \in C} \sum_{t_\ell} \sum_k \sum_{s_{\ell k}} \log(P(w_{d\ell k}|r_{\ell i}, t_\ell, \beta, s_{\ell k})))$$

$$\gamma_i = \alpha_i + \sum_\ell \phi_{\ell i}$$

We have now completed the variational E-step by finding the q^* within our restricted family that maximizes the lower bound, assuming the model parameters to be fixed.

3.3.2 M-step

The M-step will now consist of finding the model parameter values that maximize the lower bound, assuming the q^* distribution to be fixed. The parameters that must be updated in this step are α, θ, β, and μ.

First, we will consider the case of α. Taking the derivative of the lower bound with respect to α gives us

$$\frac{\partial F}{\partial \alpha_i} = \psi(\sum_j \alpha_j) - \psi(\alpha_i) + \psi(\gamma_i) - \psi(\sum_j \gamma_j)$$

Note that the summation over all α_j inside the first digamma function prevents us from setting the equation to zero and solving directly for each α_j. One possible workaround is to use the Newton-Raphson optimization procedure. The update equation for this procedure is

$$\alpha_{new} = \alpha_{old} - H^{-1}(\alpha_{old}) g(\alpha_{old})$$

$g(\alpha)$ is the gradient, and H^{-1} is the inverse of the Hessian. Generally the matrix inversion results in $O(N^3)$ complexity, but for our special case the form of the matrix allows us to achieve a linear-time Newton-Raphson method [2]. The individual entries in the Hessian are then given by

$$\frac{\partial F}{\partial \alpha_i \alpha_j} = \delta(i,j) M \psi'(\alpha_i) - \psi'(\sum_j \alpha_j)$$

where M is the number training documents.

Now we must update the other model parameters θ, β, and μ. Because we are again dealing with hidden variables in the form of t and s, we must perform another 'inner' standard EM procedure. In order to avoid confusion, we will refer to the inner EM auxiliary distribution as $u(t,s)$ and the associated lower bound as $G(u,\Theta)$. We formulate the EM problem in the same way as before (pulling a $u(t,s)$ term out of the log to get an expectation and using Jensen's inequality). The new lower bound of the log likelihood of a document is then

$$G(u,\Theta) = (\sum_{d \in C} \sum_{(\ell,k)} \mathrm{E}_u(\log(P(t_\ell|\theta)) + \log(P(s_{\ell k}|\mu)) - \log(u(t,s)) + \sum_j \phi_{\ell j} \log(P(w_{d\ell k}|r_{\ell j}, t_\ell, \beta_j, s)))$$

For the inner E-step, the optimal u^* is the posterior probability of the hidden variables, which can be computed using Bayes' Rule

$$u^*(t,s) = P(t,s|w,\theta,\beta,\mu,\phi) = \frac{P(w|t,s,\beta,\phi)P(t,s|\theta,\mu)}{\sum_{(t,s)} P(w|t,s,\theta,\beta,\mu,\phi)}$$

Once the u^* distribution has been computed, the updated parameters can be computed by taking derivatives and setting them to 0.

$$\frac{\partial G}{\partial \theta_t} = \sum_{d \in C} \sum_{(\ell,k)} \frac{u^*(t)}{\theta_t} - \lambda$$
$$\frac{\partial G}{\partial \mu_s} = \sum_{d \in C} \sum_{(\ell,k)} \frac{u^*(s)}{\mu_s} - \lambda$$
$$\frac{\partial G}{\partial \beta_{jw}} = \frac{\partial}{\partial \beta_{jw}} (\sum_{d \in C} \sum_{(\ell,k)} \mathrm{E}_u(\phi_{\ell j} \log(P(w_{d\ell k} | r_{\ell j}, t_\ell, \beta_j, s))) + \lambda_j (1 - \sum_{w'} \beta_{jw'}))$$

Setting these equations to zero and solving for the parameters gives us

$$\theta_t = u^*(t)$$
$$\mu_s = u^*(s)$$
$$\beta_{0w} \propto \sum_{d \in C} \sum_{(\ell,k)} \phi_{\ell 0} \mathbf{1}_{w==w_{d\ell k}}$$
$$\beta_{1w} \propto \sum_{d \in C} \sum_{(\ell,k)} u^*(s=0) \phi_{\ell 1} \mathbf{1}_{w==w_{d\ell k}}$$

This inner EM procedure occurs during the M-step of our variational EM scheme. Since we are performing variational EM until convergence and the inner EM scheme is embedded within the variational M-step, there is no need to directly iterate the inner EM step.

4 Conclusion and future work

Our LDA model tries to capture the differences in the vocabulary used to discuss protein-protein interactions versus background discussion. Furthermore, it models this difference using topics selected at the sentence level. This should allow the model to localize the interaction topics, and thereby enable the identification of the interacting proteins.

Given a training corpus with labeled interacting proteins, training could be accomplished by 'freezing' sentence and protein labels during parameter estimation. Once the model parameters are determined, information extraction could then be performed on new abstracts by 'freezing' all model parameters except θ and then retraining on the new abstracts. The θ values for protein-protein pairs could then be loosely interpreted as the 'interaction strength', allowing the identification of interacting pairs.

The next step in this work is to implement these learning algorithms and apply them to an actual corpus. The Biomolecular Interaction Database (BIND) contains PubMed reference links for its interactions, allowing the automatic creation of a training corpus [1]. Also, we are in possession of a corpus of Escheria Coli abstracts with an index of proteins appearing in them. This corpus could be combined with DIP data to develop another training set.

References

[1] The Biomolecular Interaction Network Database and related tools 2005 update, Nucleic Acids Res. 2005 Jan 1;33(Database issue):D418-24.

[2] D. Blei, A. Ng and M. Jordan,
Latent Dirichlet Allocation,
Journal of Machine Learning Research, vol. 3, pp.993–1022, 2003.

[3] J. Dickey,
Multiple hypergeometric functions: Probabilistic interpretations and statistical uses,
Journal of the American Statistical Association, 78:628–637, 1983.

[4] M. Huang, X. Zhu, Y. Hao, D. Payan, K. Qu, and M. Li,
Discovering patterns to extract protein-protein interactions from full texts,
Bioinformatics, 2004.

[5] L. Hunter and K.B. Cohen
Biomedical Language Processing: What's Beyond PubMed?
Mol Cell 21:589-594, 2006.

[6] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul,
An introduction to variational methods for graphical models,
Learning in Graphical Models, Cambridge: MIT Press, 1999.

[7] E. Marcotte, I. Xenarios, and D. Eisenberg,
Mining literature for protein-protein interactions,
Bioinformatics, 2000.

[8] D. L. Nelson, and M. M. Cox
Lehninger Principles of Biochemistry, Fourth Edition, 2004.

[9] A. Ramani, R. Bunescu, R. Mooney, and E. Marcotte,
Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome,
Genome Biology, 2005.

[10] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg,
The Database of Interacting Proteins: 2004 update,
NAR 32 Database issue:D449-51, 2004.

[11] J. Temkin and M. Gilder,
Extraction of protein interaction information from unstructured text using a context-free grammar,
Bioinformatics, 2003.