Bayesian Skill Ranking

Leland Chen, Joseph Huang, Ryan Thompson

March 16, 2011

1 Introduction

Template

2 Related Work

For head-to-head competitions (Player A vs. Player B) with binary outcomes (Player A Wins, Player A Loses), there exist many well-understood techniques for compressing player performances into a single skill parameter. ELO [1] is the most widespread such ranking system. ELO was designed around the logistic probabilistic player comparison model employed by ELO is the Bradley-Terry's Here, player performance is modelled as a logistic function, as suggested originally by [2].

ELO was later changed to a Gaussian model

Glicko is modelled using both logistic and Gaussian

Talk about the history of ELO [1] and Glicko [3], and the ongoing debate between logistic and gaussian (logit vs. probit).

Let's talk about [4] and [5] and then just follow the citations that those guys make to introduce the general challenges and state-of-the art.

What if you want multiple players? TrueSkill addresses having multiple players. But it's only Gaussian.

The Elo rating system is used for calculating relative skill levels of players in two-player games, but has been adapted to team sports like soccer, football, basketball, and baseball. The performance of players/teams is inferred from wins, losses, and draws against other players/teams, while depending on the ratings of the opponent and the results scored against them. (Elo also has some mathetmatical issues of that have been addressed through different means). Elo only considers the final score, which is not always indicative of how competitive the game was, and when used for teams, does not consider the players individually. Therefore, our intent is to build a more sophisticated model that determines 6 different skills (3 on offensive and 3 on defense) for each player using Bayesian Networks. By doing so, we can make better inferences on how well these sets of players contribute as a team on each possession and how likely they are to score 0, 1, 2, or 3 points per possession against their opponent.

Glicko - their algorithm ignores certain features of model that a likelihood-based analysis would use for computational ease; do not account for posterior correlations of players' strengths - excluded correlations so that # of model parameters greatly reduced. Approximating algorithm performs reasonably well.

Basketball also brings another key challenge to the field of Bayesian Skill estimation.

- Results are not binary Win/Lose outcomes.
- Results are influenced by the skill differential of more than two players.

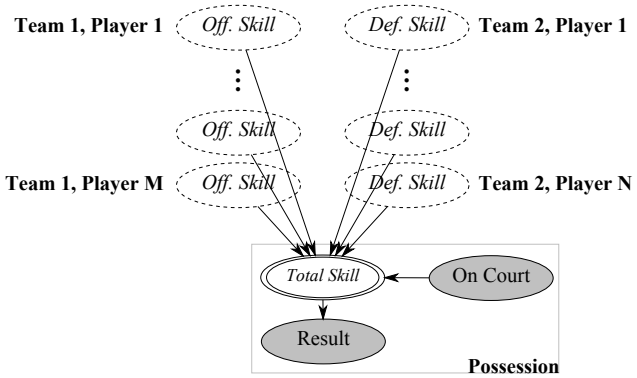
3 Method

3.1 Traditional Networks

We model the result of each possession as an independent and identically distributed random variable, *Result*. An NBA roster has 12 players, but only five of them are on the court for a given team during any single posession.

(Are we still doing this?) To reduce noise, we ignore "garbage time" posessions. To simplify the model we will only consider possessions in the first half of games.

We begin with a very basic model that follows traditional "skill ranking" in the sense that each player has an unobserved (or parameterized) skill that has some distribution, and using inference (or parameter estimation) we can determine the value of each player's skill. In this network, we decided each player would have a hidden offsensive and defensive skill value, shared across possessions:



These skills would contribute deterministically in some way to an "effective" total skill differential between the two teams, and then the *Result* variable would be one of four outcomes:

- $R = 0$ Offensive Team Scores nothing, change of possession (e.g. turnover, defensive rebound, etc.)
- $R = 1$ Offensive Team Scores 1 point
- $R = 2$ Offensive Team Scores 2 points
- $R = 3$ Offensive Team Scores 3 points

Note: We are ignoring the rare 4 point plays.

The *On Court* random variable "multiplexes" between those players that are on the court and those that are not.

3.2 Issues

These traditional models [4] have difficulty capturing the proper causalities when Results can have multinomial-valued outcomes. For example, regardless of the parameterization of the *Results* CPD, both $\Pr\{R = 2\}$ and $\Pr\{R = 3\}$ would depend on the same skills of the same players. The relative distribution of outcomes $\Pr\{R = 2\}$ vs. $\Pr\{R = 3\}$ would be

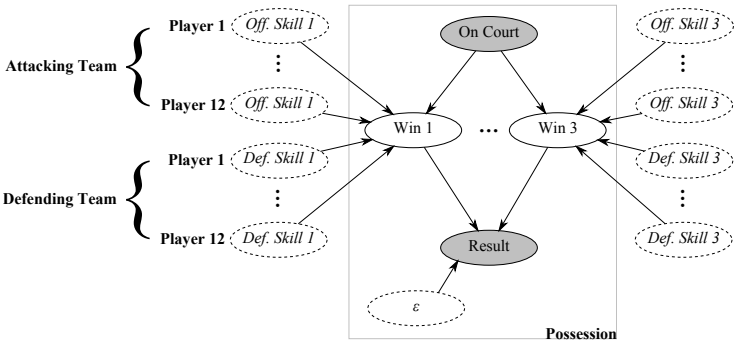
shared across all "units" (i.e. all combinations of *On Court* assignments).

In reality, a team that scores $R = 3$ half the time and $R = 1$ half the time is just as good as a team that scores $R = 2$ all the time. However, any traditional Win/Lose model will unfairly penalize the likelihood of one of these teams over the other and leads to under-fitting. This might suggest that we have a random variable that represents $\mathrm{E}\left[\text{points scored}\right]$, but this doesn't pass the clarity test.

Secondly, there is a lot of value in being able to compare with the state-of-the-art in the Win/Lose based Bayesian Skill Ranking literature. Specifically, there is common debate over logistic vs. Gaussian skill/performance distributions and we wish to be sensitive to that conversation in this project. Having a multinomial outcome makes it difficult to directly compare logistic vs. Gaussian in the standard skill/performance framework because there is no consensus in the literature about how to extend these models just to support ties [6], let alone general multinomial outcomes.

3.3 Proposed Network

To address some of the shortcomings discussed in **Section 3.2**, we propose the following:



In this network, each player is represented by three skill parameters:

1. Skill 1: Ability to score (or defend against) one-point opportunities
2. Skill 2: Ability to score (or defend against) two-point opportunities
3. Skill 3: Ability to score (or defend against) three-point opportunities

For notational convenience, let

$$A\left(\text{Skills}\right) \triangleq \exp\left(\sum_i \text{Off. Team, Player i Off. Skill } k\right)$$

$$B\left(\text{Skills}\right) \triangleq \exp\left(\sum_i \text{Def. Team, Player i Def. Skill } k\right)$$

where

$$\vec{\theta}_k = \left\{\text{Off. Team, Player i Off. Skill } k\right\}_k \cup \left\{\text{Def. Team, Player i Def. Skill } k\right\}_k$$

On each possession, there are three binary hidden random variables defined by a probability function with parameters that we will learn. In particular, each of the "Win" random variables depends on the corresponding skills of players on court as input (which are learned):

1. *Win 1*: True if there was a **guaranteed opportunity** to score one point at some point during the possession.
2. *Win 2*: True if there was a **guaranteed opportunity** to score two points at some point during the possession.
3. *Win 3*: True if there was a **guaranteed opportunity** to score three points at some point during the possession.

Note: The term "opportunity" is not intended in the typical basketball sense, where it often means "the opportunity to take a shot".

Each of the *Win k* events has a CPD parameterized by $\vec{\theta}_k$ (whose values we will learn during training), which depends on the skills of the players. In the basic model of player skills, the skill of a five-man team is the sum of the skills of its players.

For example, the Bradley-Terry model corresponds to:

$$\Pr\left\{\text{Win}_k = \textbf{True} \mid \text{Skills}\right\} = \frac{A\left(\text{Skills}\right)}{A\left(\text{Skills}\right) + B\left(\text{Skills}\right)}$$

Alternatively, the Thurstone Case V model corresponds to:

$$\Pr\left\{\text{Win}_k = \textbf{True} \mid \text{Skills}\right\} = \Phi\left(A - B\right)$$

The logit (Bradley-Terry) and probit(Thurstone Case V) are the main Binary Response Models used throughout the history of the skill ranking literature [1, 3, 4, 5].

However, the power of this Bayesian Network construction is that any Binary Response Model can be plugged in modularly. One can easily compare the performance of Cauchy [7], Log-Log and Complementary Log-Log [8], Scobit [9], etc. without changing the network or the core inference algorithm.

For notational convenience, we will write $W_k = w_k^1$ for $\text{Win}_k = \textbf{True}$, and $W_k = w_k^0$ when **False**.

To understand the relationship between $\{Win\}$ and $Result$, let's go through an example. Imagine that for a particular five-man unit, the θ are known and the resulting probabilities are:

$$\begin{aligned} \Pr\left\{W_1 = w_1^1\right\} = 5\% \quad &, \quad \Pr\left\{W_1 = w_1^0\right\} = 95\% \\ \Pr\left\{W_2 = w_2^1\right\} = 35\% \quad &, \quad \Pr\left\{W_2 = w_2^0\right\} = 65\% \\ \Pr\left\{W_3 = w_3^1\right\} = 10\% \quad &, \quad \Pr\left\{W_3 = w_3^0\right\} = 90\% \end{aligned}$$

This means, on each possession

- the offensive team has a 10% probability of scoring 3 points (i.e. $R = r^3$ with 90% probability)
- the offensive team has a $(100\% - 10\%) \times 35\%$ probability of scoring 2 points (i.e. $\Pr\left\{R = r^2\right\} = 31.50\%$)
- the offensive team has a $90\% \times 65\% \times 5\%$ probability of scoring 1 point (i.e. $\Pr\left\{R = r^1\right\} = 2.925\%$)

- the offensive team has a 55.575% chance of scoring 0 points

Essentially, *Result* simply selects the highest-scoring decision available during the possession. If *Win 3* is true, *Result* is 3; If *Win 3* is false but *Win 2* is true, *Result* is 2; etc.

Result can be specified with either a tree-CPD or table-CPD, but it has the following probabilities.

$R \mid W_1, W_2, W_3$	$R = r^0$	$R = r^1$	$R = r^2$	$R = r^3$
$w_3^1 w_2^1 w_1^1$	ε	ε	ε	$(1-3\varepsilon)$
$w_3^1 w_2^1 w_1^0$	ε	ε	ε	$(1-3\varepsilon)$
$w_3^1 w_2^0 w_1^1$	ε	ε	ε	$(1-3\varepsilon)$
$w_3^1 w_2^0 w_1^0$	ε	ε	ε	$(1-3\varepsilon)$
$w_3^0 w_2^1 w_1^1$	ε	ε	$(1-3\varepsilon)$	ε
$w_3^0 w_2^1 w_1^0$	ε	ε	$(1-3\varepsilon)$	ε
$w_3^0 w_2^0 w_1^1$	ε	$(1-3\varepsilon)$	ε	ε
$w_3^0 w_2^0 w_1^0$	$(1-3\varepsilon)$	ε	ε	ε

The single parameter ε is inspired by the noisy-max and essentially indicates unmodelled errors (e.g. offensive or defensive mistakes). The better our model fits the actual flow of the game, the smaller ε should be.

4 Implemention

We will perform Parameter Estimation with Missing Data. Each player's skill is treated as a fixed parameter of the *Win* CPDs. ε is also a parameter. Each possession from our dataset is treated as an i.i.d. observation from the joint distribution of the entire network. **Result** and **OnCourt** are always observed, *Win 1*, *Win 2*, and *Win 3* are always missing/hidden.

The general algorithm, then, consists of iteratively maximizing:

$$
\begin{aligned}
L &= \prod_{\mathcal{D}} \Pr\left\{\mathcal{D} \mid \vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \varepsilon\right\} \\
&= \prod_{\mathcal{D}} \sum_{\substack{w_1 \in \mathrm{Val}(W_1) \\ w_2 \in \mathrm{Val}(W_2) \\ w_3 \in \mathrm{Val}(W_3)}} \\
&\qquad \Pr\left\{R = \mathcal{D}_r, W_1 = w_1, W_2 = w_2, W_3 = w_3, \mathcal{D}_C \mid \vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3, \varepsilon\right\} \\
&= \prod_{\mathcal{D}} \sum_{\substack{w_1 \in \mathrm{Val}(W_1) \\ w_2 \in \mathrm{Val}(W_2) \\ w_3 \in \mathrm{Val}(W_3)}} \\
&\qquad \Pr\left\{R = \mathcal{D}_r \mid W_1 = w_1, W_2 = w_2, W_3 = w_3, \cancel{\mathcal{D}_C, \vec{\theta}_1, \vec{\theta}_2, \vec{\theta}_3}, \varepsilon\right\} \\
&\qquad \Pr\left\{W_1 = w_1 \mid C = \mathcal{D}_c, \vec{\theta}_1, \cancel{\vec{\theta}_2, \vec{\theta}_3, \varepsilon}\right\} \\
&\qquad \Pr\left\{W_2 = w_2 \mid C = \mathcal{D}_c, \vec{\theta}_2, \cancel{\vec{\theta}_1, \vec{\theta}_3, \varepsilon}\right\} \\
&\qquad \Pr\left\{W_3 = w_3 \mid C = \mathcal{D}_c, \vec{\theta}_3, \cancel{\vec{\theta}_1, \vec{\theta}_2, \varepsilon}\right\}
\end{aligned}
$$

In the E-step, we perform inference on the eight possible combinations of *Win 1*, *Win 2*, and *Win 3* and make soft-assignments to each of the eight versions each datapoint.

In the M-step, we perform maximum likelihood estimation of parameters $\vec{\theta}_1$, $\vec{\theta}_2$, $\vec{\theta}_3$, and ε using the same algorithms as if $\mathcal{D}$ had completely observed $W_1 = \mathcal{D}_1$, $W_2 = \mathcal{D}_2$, and $W_3 = \mathcal{D}_3$ (weighted by the soft assignments of the E-step).

During the M-step, we can take advantage of global decomposition allowing us to log-maximize each section of the network separately:

$$
\begin{aligned}
\ell = &\left(\sum_{\mathcal{D}} \ell\left\{R = \mathcal{D}_r \mid W_1 = \mathcal{D}_1, W_2 = \mathcal{D}_2, W_3 = \mathcal{D}_3, \varepsilon\right\}\right) \\
&\left(\sum_{\mathcal{D}} \ell\left\{W_1 = \mathcal{D}_1 \mid C = \mathcal{D}_c, \vec{\theta}_1\right\}\right) \\
&\left(\sum_{\mathcal{D}} \ell\left\{W_2 = \mathcal{D}_2 \mid C = \mathcal{D}_c, \vec{\theta}_2\right\}\right) \\
&\left(\sum_{\mathcal{D}} \ell\left\{W_3 = \mathcal{D}_3 \mid C = \mathcal{D}_c, \vec{\theta}_3\right\}\right)
\end{aligned}
$$

$C = \mathcal{D}_c$ is the *OnCourt* variable. It is always observed and assumed to have a uniform prior so it cancels out during any $\arg\max$ operation.

4.1 M-Step: Maximum Likelihood ε

Using the table CPD for from **Section 3.2** for $\Pr\{R = \mathcal{D}_r \mid W_1 = \mathcal{D}_1, W_2 = \mathcal{D}_2, W_3 = \mathcal{D}_3, \varepsilon\}$ and collecting like terms, we find:

$$
\begin{aligned}
&\operatorname*{argmax}_{\varepsilon} \left\{\prod_{\mathcal{D}} \Pr\left\{R = \mathcal{D}_r \mid W_1 = \mathcal{D}_1, W_2 = \mathcal{D}_2, W_3 = \mathcal{D}_3, \varepsilon\right\}\right\} \\
&\qquad = \operatorname*{argmax}_{\varepsilon} \left\{(1-3\varepsilon)^{M_{modelled}} (\varepsilon)^{M_{noise}}\right\}
\end{aligned}
$$

with solution

$$
\varepsilon = \frac{1}{3} \frac{M_{noise}}{M_{noise} + M_{modelled}}
$$

where

$$
\begin{aligned}
M_{modelled} \triangleq \quad & \mathrm{M}\left[r^3, w_3^1\right] \\
& + \mathrm{M}\left[r^2, w_3^0, w_2^1\right] \\
& + \mathrm{M}\left[r^1, w_3^0, w_2^0, w_1^1\right] \\
& + \mathrm{M}\left[r^0, w_3^0, w_2^0, w_1^0\right]
\end{aligned}
$$

and M_{noise} is a count of the remaining observations such that $M_{modelled} + M_{noise} = M$, the total number of observations.

4.2 M-Step: Maximum Likelihood θ_i

In this project, we implement the two major pairwise comparison models: Bradley-Terry, and Thurstone Case V.

In Bradley-Terry, each *Win i* random variable follows a logistic distribution. Let

$$
\Pr\left\{W_i = w_i^0 \mid \vec{C}, \vec{\theta}_i\right\} \triangleq \frac{1}{1 + \exp\left(\Delta_i\right)}
$$

so that

$$\Pr\left\{W_i = w_i^1 \mid \vec{C}, \vec{\theta}_i\right\} \triangleq \frac{1}{1+\exp\left(-\Delta_i\right)}$$

where

$$\Delta_i \triangleq \left[\theta_{i,\text{Off.P1}}, \ldots, \theta_{i,\text{Def.P12}}\right] \vec{C}$$

and the *OnCourt* variable $\vec{C}$ is a vector of indicator functions. For any particular possession, exactly ten elements of $\vec{C}$ are 1 (there are ten basketball players on the court at once). All the other elements are zero.

Without loss of generality, we can assume that the defensive θs are negative numbers, that reduce the overall $\Pr\{Win\}$ when added. (This is simpler than subtracting positive θ_{Def} and then having to keep track of positive and negative signs the whole time.)

Now, we wish to compute:

$$\underset{\vec{\theta}_i}{\text{argmax}} \left\{ \prod_{\mathcal{D}} \Pr\left\{W_i = \mathcal{D}_i \mid \vec{\theta}_i\right\} \right\}$$

when W_i is fully observed with the solution expressed with respect to sufficient statistics of $\vec{\theta}_i$.

This reduces to weighted logistic regression. We use a weighted variant of the basic Newton-Raphson logistic regression technique [10].

In Thurstone Case V, each *Win i* random variable follows a Gaussian distribution.

$$\Pr\left\{W_i = w_i^0 \mid \vec{\theta}_i\right\} \triangleq \Phi\left(\Delta_i\right)$$

replaces the logit function with the probit function. Again, we use a weighted variant of [11], the basic Newton-Raphson procedure for probit regression.

4.3 Expectation-Maximization: E-step

During the E-step all parameters are fixed, so we simply evaluate probabilities directly. For each datapoint $\langle \mathcal{D}_r, \mathcal{D}_c \rangle$ we create soft-datapoints:

- $\left\langle \mathcal{D}_r, \mathcal{D}_c, w_1^1, w_2^1, w_3^1 \right\rangle$ with weight proportional to $\Pr\left\{\mathcal{D}_r, \mathcal{D}_c, w_1^1, w_2^1, w_3^1 \mid \vec{\theta}, \varepsilon\right\}$
- $\left\langle \mathcal{D}_r, \mathcal{D}_c, w_1^1, w_2^1, w_3^0 \right\rangle$ with weight proportional to $\Pr\left\{\mathcal{D}_r, \mathcal{D}_c, w_1^1, w_2^1, w_3^0 \mid \vec{\theta}, \varepsilon\right\}$
- $\vdots$
- $\left\langle \mathcal{D}_r, \mathcal{D}_c, w_1^0, w_2^0, w_3^0 \right\rangle$ with weight proportional to $\Pr\left\{\mathcal{D}_r, \mathcal{D}_c, w_1^0, w_2^0, w_3^0 \mid \vec{\theta}, \varepsilon\right\}$

4.4 Expectation-Maximization: Initialization

Since soft assignments to W_3 represent roughly the probability of scoring three points during a possession, we can initialize $\Pr\left\{W_3 = w_3^1\right\}$ to the fraction

$$w_3^{\text{init}} := \frac{\text{M}\left[r^3\right]}{M}$$

Similarly, soft assignments to W_3 and W_2 suggest that

$$\frac{\text{M}\left[r^2\right]}{M} \rightarrow \Pr\left\{r^2\right\} \approx \left(1 - \Pr\left\{W_3 = w_3^1\right\}\right) \times \Pr\left\{W_2 = w_2^1\right\}$$

is the probability of scoring two points during a possession. So let's initialize

$$w_2^{\text{init}} := \left(\text{M}\left[r^2\right] \div M\right) \div \left(1 - w_3^{\text{init}}\right)$$

And for the same reason,

$$w_1^{\text{init}} := \left(\text{M}\left[r^1\right] \div M\right) \div \left(1 - w_2^{\text{init}}\right)$$

With these soft assignments we can begin EM on the M-step.

4.5 Implementation Considerations

E-Step MLE of θ are not closed-form but require iteration. We choose closed-form Hessian-based weighted regression. Overshooting and step size, stopping criterion, underflow and pruning points, are all real considerations.

5 Results

We started by running a single game between DET and CLE on 2007 Feb 2.

Then all games between CLE and DET

Then full SW division

Evaluation:

We tried separating 25% of the possessions for testing and 75% of the possessions for training

When doing this, we notice that the empirical expectation

$$\text{E}\left[R\right] \approx \frac{1}{m} \sum_{R} r \,\text{M}\left[r\right] = 1.0486$$

is 1.0486 points per possession (over 2364 possessions) in the training data, yet it predicts

$$\text{E}\left[\text{E}_{\text{training}}\left[R|C\right]\right] = 1.0183 \text{ or } 1.0217$$

points per possession (over 788 possession) for the possessions in the test set.

The actual number of points scored in the test set was 1.0178, which indicates that the model is accounting for player skills.

6 Analysis

Metrics: epsilon test/training vs. dataset size $E[Prdatapoint]$ M_count

(This is just one example though...do we need/want more?)

"[NBA West Southwest Intradivision games] In the 2008-2009 NBA season, Shane Battier was on the Second team NBA All-Defensive Team. However, he had a non-adjusted +/- of -18 in the games he played against the Southwest division, which is reflected in the results of only -1.594, -0.313, and 1.675 in his defensive skill ranking against 1 pt, 2 pt, and 3 pt respectively.

Surpringly, Ryan Bowen had 1 pt defensive skills of -13.475, 2 pt defensive skills of -2.207, and 3 pt defensive skills of -18.819 . Bowen only participated in one of seven of his team's intradivision games, playing 16 minutes and ending with a non-adjusted +/- of -3. However, when he was on the floor, the opponent had 32 possessions but only R(3) = 2 (6.25

(Jekyll and Hyde like numbers). On offense, Matt Carroll posted a 1 pt offensive skill of -14.792, 2 pt offensive skill of 0.713, and 3 pt offensive skill of -14.604. +0.713 put him second out of all players in intradivision games in the Southwest division. His defensive skill for 1 pt was -13.074, 2 pt 0.748, and 3 pt -14.252. So why the disparity in the numbers? It turns out that Carroll only played in one game for about 5 minutes. While in the game, his opponent never scored 1 point or 3 points on any of their possessions, and hence the "great" defensive skill. On the offensive end, Carroll's team also did not score 1 point or 3 points on any of their possessions. They did convert on 4/9 (44.4

"

7 Conclusions

Bayesian prior/smoothing Coach skill - can add coach as another variable, and see if they are using the most optimal lineups against their opponent. More binary response models

8 References

References

[1] A. Elo, *The rating of chessplayers, past and present.* Batsford, 1978.

[2] R. Bradley and M. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[3] M. Glickman, "Parameter estimation in large dynamic paired comparison experiments," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 48, no. 3, pp. 377–394, 1999.

[4] R. Herbrich, T. Minka, and T. Graepel, "TrueSkillTM: A Bayesian skill rating system," *Advances in Neural Information Processing Systems*, vol. 20, pp. 569–576, 2007.

[5] R. Coulom, "Whole-history rating: A bayesian rating system for players of time-varying strength," *Computers and Games*, pp. 113–124, 2008.

[6] D. Hunter, "MM algorithms for generalized Bradley-Terry models," *The Annals of Statistics*, vol. 32, no. 1, pp. 384–406, 2004.

[7] "Maximum likelihood estimation," Charles H. Franklin, June 2005, http://users.polisci.wisc.edu/franklin/Content/MLE/Lecs/MLELec07p4up.pdf.

[8] J. Long, *Regression models for categorical and limited dependent variables.* Sage Publications, Inc, 1997.

[9] J. Nagler, "Scobit: an alternative estimator to logit and probit," *American Journal of Political Science*, vol. 38, no. 1, pp. 230–255, 1994.

[10] "Logistic regression," Jia Li, September 2008, http://www.stat.psu.edu/ jiali/course/stat597e/notes2/logit.pdf.

[11] E. Demidenko, "Computational aspects of probit model," *Mathematical Communications*, vol. 6, no. 2, 2001.