A Variational Inference Strategy for “Back-Predicting” Pre-2015 Statcast Data in Major League Baseball

Isaac Kleisle-Murphy[*,1]
[*]Stanford University

ABSTRACT This paper explores how variational Bayesian inference can be leveraged to “back-predict” Statcast estimators in Major League for pre-2015 seasons. In particular, it finds that combination of MAP estimation and ELBO maximisation show the most promise, particularly for constructing posterior point estimates.

KEYWORDS

1. Introduction

In 2015, Major League Baseball (MLB) introduce Statcast, a complex motion-capture system designed to track and measure all in-game player and ball movements. For each play, the Statcast system records a plethora of measurements (“fields”) useful to the analysis of baseball, including: pitch velocity, pitch movement, pitch spin, batted-ball exit velocity, batted-ball launch angle, and player sprint speed.

Since its introduction, Statcast has reshaped the baseball, and in particular, the front office landscape. Among other things, teams can now: precisely quantify the athletic abilities of their players and opponents; control for fortune (or lack thereof) and degree of difficulty in their evaluations of plays and players; and compute live event probabilities, to better understand and optimize in-game decisions and tactics. In many ways, Statcast has spawned a second “Moneyball” across Major League Baseball.

However, the Statcast dataset – while *prima facie* voluminous – is in many ways limited. With only 2016-present[2] available, analysts often find themselves with too little to explicate long term Statcast-related trends, and in particular, aging trends. In light of this, this paper proposes method of “back-predicting” Statcast fields from pre-Statcast seasons, for use as a “pre-Statcast Statcast” approximation.

More precisely, this paper derives and tests two flavors of variational Bayesian inference that show some promise in back-predicting these seasons. As a benchmark for these variational inference models, it also proffers a very simple Bayesian neural network. It ultimately finds that while the variational approximation shows promise as a point estimator, additional research and engineering is necessary for a fully Bayesian synthesis.

2. Variational Inference

A. Multinomial Notation

For unique players $p = 1, \dots, P$; seasons $j = 1, \dots, J_p$ of player p; and plate appearances (PA) $i = 1, \dots, n_{p,j}^{(Y)}$ by player p in season j, let $Y_{p,j} \in \mathbb{R}^K$ be a vector of K-outcome multinomial count data, i.e.

$$Y_{p,j} = \left\langle \sum_{i=1}^{n_{p,j}^{(Y)}} \delta(Y_{i,p,j} = 1), \dots, \sum_{i=1}^{n_{p,j}^{(Y)}} \delta(Y_{i,p,j} = K) \right\rangle,$$

where $Y_{i,p,j}$ is the K-outcome categorical realization of a single plate appearance and

$$\vec{1}^T Y_{p,j} = n_{p,j}^{(Y)}.$$

Intuitively, $Y_{p,j}$ might be a vector containing the total number of singles, home-runs, strikeouts, etc. achieved by player p in season j. That is, $Y_{p,j}$ is a multinomial realization of observed baseball outcomes, where each plate appearance is a single categorical trial.

B. Statcast Notation

Similarly, let $\ell = 1, \dots, L$ index a particular set of Statcast fields (e.g. exit velocity, launch angle, etc.) and let

$$X_{i,j,p} = \left\langle X_{i,j,p}^{(\ell=1)}, \dots X_{i,j,p}^{(\ell=L)} \right\rangle$$

where: Q is the (unknown) underlying joint distribution of the fields (parameterized by Θ); $p = 1, \dots, P$ again index the unique players; $j = 1, \dots, J_p$ again index the unique seasons of player p; and where $i = 1, \dots, n_{p,j}^{(X)}$ index the batted-ball events of player p in season j.[3].

doi: 10.1534/genetics.XXX.XXXXXX
Manuscript compiled: Saturday 5th June, 2021

[2] 2015 Statcast data quality is widely questioned across baseball.

[3] Note that a plate appearance (PA) is distinct from a batted-ball event (BBE): a BBE requires the ball be hit into play, whereas the PA may conclude with a static event such as a strikeout or walk

C. Generative Model 1.0

The full generative model, as explicated in the paragraphs that follow, is given as follows:

$$\begin{aligned}
X_{i,j,p} &\sim Q(\Theta) && i = 1, \dots, n^{(X)}_{p,j} \\
\pi_{p,j} &= g(\bar{X}_{p,j}, \theta, \beta) && \theta \in \mathbb{R}^{L \times K}, \beta \in \mathbb{R}^K \\
\vec{\theta} &\sim \text{MVN}(\vec{0}, \tau I) \\
\beta &\propto 1 \\
Y_{p,j} &\sim \text{Multinomial}(\pi_{p,j}, n^{(Y)}_{p,j}).
\end{aligned}$$

Here, $g(\cdot)$ is the softmax function $g : L \times \dim(\theta) \to \Delta^K$, given by

$$g(\bar{X}_{p,j}; \theta, \beta) := \frac{1}{\sum_{w=1}^{K} \exp(X_{p,j}\theta_w + \beta_w)} \cdot \Big\langle \exp(X_{p,j}\theta_1 + \beta_1), \dots, \exp(X_{p,j}\theta_K + \beta_K) \Big\rangle.$$

Meanwhile, prior Q is the underlying joint distribution of the Statcast fields $X_{i,j,p}$, at the batted-ball-event level – i.e. each pitch hit into play is a realization of Q. We may write that Q is parameterized by some Θ, though this is largely trivial, as the variational approximation does not deal directly with it.

Note that for this generative model, the p, j indexing/notation is temporary. While the p, j assumption is useful for initially setting up the problem, a more generalized notation (introduced in section H.) is used for describing model classes and inference techniques.

D. Central Limit Theorem

Unfortunately $Q_{p,j}$ may be unruly and/or nonparametric, which precludes any direct posterior inference on $\Theta_{p,j} | Y_{p,j}$. However, the Central Limit Theorem[4] gives

$$\sqrt{n^{(X)}_{p,j}} \Big(\bar{X}_{p,j} - E[Q] \Big) \xrightarrow{d} MVN\Big(\vec{0}, E[(Q - E[Q])(Q - E[Q])^T] \Big).$$

This encourages the CLT assumption for the prior

$$\bar{X}_{p,j} \sim MVN\Big(E[Q_{j,p}], \frac{1}{n} E[(Q - E[Q])(Q - E[Q])^T] \Big)$$

Now, if we instead aim to conduct posterior inference on $\bar{X}_{p,j} | Y_{p,j}, \theta, \beta$, the variational approximation

$$\bar{X}_{p,j} | Y_{p,j}, \theta, \beta, \Theta \sim N(\mu_{p,j}, \Sigma_{p,j})$$

suddenly becomes much more palatable. Note that for any such variational approximation, we will use $q(x_{p,j}; N(\mu_{p,j}, \Sigma_{p,j}))$ to identify the variational density, as opposed to $p(x_{p,j} | Y, \Theta)$ to identify the true posterior density.

E. $\bar{X}_{p,j}$ as a Latent Variable

For any season j in the Statcast era, parametric inference attempts about $\bar{X}_{p,j} | Y_{p,j}$ are uninteresting: measured values of $X_{i,j,p}$ (for batted balls $i = 1, \dots, n^{(X)}_{p,j}$) are readily available for these seasons, and thus it makes more sense to work directly with these individual batted-ball-event data points.

However, for non-Statcast seasons, $\bar{X}_{p,j}$ effectively serves as a latent variable for the multinomial model. Since measurements of $X_{i,j,p}$ do not exist, the variational approximation is much more useful. In particular, if we can retrieve latent variational parameters $\mu_{p,j}, \Sigma_{p,j})$ that maximize (or maximize a lower bound of) the multinomial logit model's likelihood, we can use these posterior variational parameters to approximate the distribution of sufficient statistics of Statcast fields from pre-Statcast seasons.

F. The Prior Distribution of $\bar{X}_{p,j}$

As set forth in section D, it is reasonable to assume a

$$\bar{X}_{p,j} \sim MVN(\vec{\mu}_0, \Sigma_0)$$

prior distribution. Assuming data from at least one Statcast season is available, the population empirical Bayes choice emerges

$$\vec{\mu}_0 = \bar{X}_{\mathcal{S}} := \frac{\sum_{(p,j) \in \mathcal{S}} n^{(X)}_{p,j} \bar{X}_{p,j}}{\sum_{(p,j) \in \mathcal{S}} n^{(X)}_{p,j}}$$

and

$$\Sigma_0 = \bar{\Sigma}_{\mathcal{S}} := \frac{\sum_{(p,j) \in \mathcal{S}} n^{(X)}_{p,j} (\bar{X}_{p,j} - \bar{X}_{\mathcal{S}})(\bar{X}_{p,j} - \bar{X}_{\mathcal{S}})^T}{\sum_{(p,j) \in \mathcal{S}} n^{(X)}_{p,j}},$$

where $\mathcal{S}$ is the set of all p, j pairs from Statcast seasons. In the interests of simplicity and computational efficiency, we may even simplify to

$$\Sigma_0 = \text{diag}(\bar{\Sigma}_{\mathcal{S}}).$$

Importantly, if the $X_{p,j}, \quad (p, j) \in \mathcal{S}$ have been (weighted) mean and center scaled, then the diagonalized prior becomes

$$\bar{X}_{p,j} \sim MVN(\vec{0}, I).$$

G. Discussion of Generative Model

In the baseball community, observed outcomes $Y_{i,p,j}$ are often understood and/or modeled as functions of various $X_{i,p,j}$. Intuitively, if a player hits the ball hard (e.g. high exit velocity) and the air (e.g. launch angle $\sim\in [10, 25]$ degrees), they can expect better outcomes $Y_{i,p,j}$ (e.g. home run, double, single) to occur. Conversely, if a player hits the ball softly (e.g. low exit velocity) or at an unfavorable angle (e.g. high or low launch angle), they can generally expect worse outcomes to occur (e.g. groundout, popout). Of course, fortune is involved: an opposing fielder may spectacularly catch a hard hit line drive, or a poorly-positioned defense may allow a softly hit ground ball to squeak through. However, on balance, it has been repeatedly demonstrated that contact quality – as reflected in many Statcast fields – underlies observed outcomes.

Given this dependence, it is natural to model

$$Y_{p,j} \approx \text{Multinomial}(n^{(Y)}_{p,j}, g(\bar{X}_{p,j}; \theta)),$$

where: g is some function $g : L \times \dim(\theta) \to \Delta^K$; θ are auxiliary parameters in support of g; and the Statcast fields $\bar{X}_{p,j}$ are averaged at the season level. An obvious choice for g is the multinomial-logit construction, wherein

$$g(\bar{X}_{p,j}; \theta, \beta) := \frac{1}{\sum_{w=1}^{K} \exp(X_{p,j}\theta_w + \beta_w)} \cdot \Big\langle \exp(X_{p,j}\theta_1 + \beta_1), \dots, \exp(X_{p,j}\theta_K + \beta_K) \Big\rangle.$$

In terms of the X and $\bar{X}$ density specification, the Gaussian assumption for the $\bar{X}_{p,j}$ follow from the CLT argument above; the variational Gaussian assumption for the posterior subsequently follows from this. Finally, the prior on $\bar{X}_{p,j}$ is a straightforward

[4] Assuming Q has finite covariance - a reasonable assumption.

empirical Bayes choice, and the diagonalization of Σ_0 and $\Sigma_{p,j}$ is for computational complexity for this initial research. During future development of this model, a complete covariance structure deserves further consideration.

H. Generative Model 2.0

While the above i, j, p notation was useful for specifying and motivating the generative model, we now refresh the notation for X, Y to a more generalized form that better serves the remainder of the paper. Specifically, let:

- $\mathcal{S}$ be the set of all player-season pairs for which Statcast data exists.
- $\tilde{\mathcal{S}}$ be the set of all player-season pairs for which Statcast data *does not exist.*
- $i = 1, \ldots, N$ index all of the player/season pairs (i.e. "rows" of data) for all seasons in $\mathcal{S}$. In terms of the old notation, each n here corresponds to a Statcast j, p tuple above.
- $\tilde{i} = 1, \ldots, \tilde{N}$ index all of the player/season pairs (i.e. "rows" of data) for all seasons in $\tilde{\mathcal{S}}$. In terms of the old notation, each n here corresponds to a non-Statcast j, p tuple above.
- $X \in \mathbb{R}^{N,L}$ be a design matrix of Statcast fields, as averaged at the player/season level. In terms of the old notation, each row of X consists of one $\bar{X}_{p,j}$. It will be our aim to infer $\tilde{X} \in \mathbb{R}^{\tilde{N},L}$.
- $Y \in \mathbb{R}^{N,K}$ be the multinomial-count response matrix of observed results from Statcast seasons. Each row in this matrix corresponds to a player-season pair $1, \ldots, N \in \mathcal{S}$, and the entries of each row are multinomial counts of various baseball outcomes (e.g. strikeout, groundout, home run, etc.).
- $\tilde{Y} \in \mathbb{R}^{\tilde{N},K}$ be the multinomial-count response matrix of observed results from non-Statcast seasons. Each row in this matrix corresponds to a player-season pair $1, \ldots, \tilde{N} \in \tilde{\mathcal{S}}$, and the entries of each row are multinomial counts of baseball outcomes.
- $\theta \in \mathbb{R}^{L,K}$ be a matrix of multinomial-logit weights, with $\beta \in \mathbb{K}$ a corresponding vector of multinomial-logit intercepts.
- $n_i^{(X)}, n_i^{(\tilde{X})}$ be the number of batted-ball events (theoretically capable of being tracked by Statcast) for a player/season pairing.
- $n_i^{(Y)}, n_i^{(\tilde{Y})}$ be the number of categorical trials for a player/season pairing – i.e., the number of plate appearances.

The formal model is then refreshed to:

$$\vec{\mu}_0 = \frac{\sum_{i \in \mathcal{S}} n_i^{(X)} X_i}{\sum_{i \in \mathcal{S}} n_i^{(X)}}$$

$$\Sigma_0 = \text{diag}\left(\frac{\sum_{i \in \mathcal{S}} n_i^{(X)} (X_i - \bar{X}_{\mathcal{S}})(X_i - \bar{X}_{\mathcal{S}})^T}{\sum_{i \in \mathcal{S}} n_i^{(X)}.} \right)$$

$$X_i \sim MVN(\vec{\mu}_0, \Sigma_0); \quad \tilde{X}_i \sim MVN(\vec{\mu}_0, \Sigma_0)$$

$$\vec{\theta} \sim MVN(\vec{0}, \tau I); \quad \beta \propto 1$$

$$\pi_i = g(X_i, \theta, \beta); \quad \tilde{\pi}_i = g(\tilde{X}_i \theta, \beta)$$

$$Y_i \sim \text{Multinomial}(n_i^{(Y)}), \pi_i); \quad \tilde{Y}_i \sim \text{Multinomial}(n_i^{(\tilde{Y})}), \tilde{\pi}_i)$$

over all $i \in \mathcal{S}$ and $i' \in \tilde{\mathcal{S}}$. Lastly, for the unsupervised data, the variational approximation to the posterior takes the form

$$\tilde{X}_i | \tilde{Y}_i, \theta, \beta \sim N(\vec{\mu}_{i'}, \Sigma_{i'}) \approx N(\vec{\mu}_{i'}, diag(\vec{\sigma}_{i'}^2)),$$

$\forall i' \in \tilde{S}$, where the diagonalization of the variational covariance is for (computational) simplicity. Observe that under this construction, a natural partition between supervised and unsupervised data arises: the (X, Y) data, corresponding to Statcast seasons, is the "supervised" data, as exact Statcast measurements exist here. Meanwhile, the $\tilde{X}, \tilde{Y}$ approximation and data is "unsupervised," as there do not exist exact measurements of $\tilde{X}$, and hence, we must approximate it through the variational inference step.

I. Method I: Supervised IRLS + Gradient Ascent on ELBO

The first method for obtaining these variational approximations involves single round of supervised IRLS, followed by a single round of gradient ascent on the ELBO. Specifically, this procedure involves:

1. Identify MAP
$$\hat{\theta}, \hat{\beta} = \arg\max_{\theta, \beta} \ell(X; Y, \theta, \beta) + \log(\pi(\theta)).$$
Note here that this maximization occurs only on the supervised data – i.e., we are only using data where both the observed outcomes and Statcast fields (as opposed to the observed outcomes but not the Statcast fields) are known to fit $\hat{\theta}$ and $\hat{\beta}$.

2. Initialise variational parameters $\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2 \; \forall i' \in \tilde{S}$. Again, this approximation is
$$p(\tilde{x}_{i'} | y_{i'}, \theta, \beta, \Theta) \approx q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2))$$
This can be done via random initialisation or mean/variance initialisation. If X has been mean-center scaled, then $\vec{\mu}_{i'} = \vec{0}, \vec{\sigma}_{i'}^2 = \vec{1}$ is the mean/variance initialisation.

3. For a specified number of iterations and/or until convergence, maximize $ELBO(\{\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2 : i' \in \tilde{S}\}, \theta, \beta)$ via gradient ascent, i.e. over all $i' \in \tilde{S}$
$$\vec{\mu}_{i'} := \vec{\mu}_{i'} + \eta \nabla_{\vec{\mu}_{i'}} ELBO(\{\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2 : i' \in \tilde{S}\}, \theta, \beta).$$
and
$$\vec{\mu}_{i'} := \vec{\sigma}_{i'}^2 + \eta \nabla_{\vec{\sigma}_{i'}^2} ELBO(\{\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2 : i' \in \tilde{S}\}, \theta, \beta).$$

The first step is straightforward. We have

$$\begin{aligned}
\hat{\theta}, \hat{\beta} &= \arg\max_{\theta, \beta} \Big[\ell(Y; X, \theta, \beta) + \log(\pi(\theta)) \Big] \\
&= \arg\max_{\theta, \beta} \Bigg[\sum_{i \in \mathcal{S}} \sum_{k=1}^{K} Y_{i,k} \Bigg(\log \exp(X_i \theta_k) - \log \sum_{v=1}^{K} \exp(X_i \theta_v) \Bigg) \\
&\quad + \sum_{k=1}^{K} + \frac{1}{2\tau} \theta_k^T \theta_k \Bigg].
\end{aligned}$$

As is standard for this kind of multinomial logit model, $\hat{\theta}, \hat{\beta}$ can be quickly achieved by a run-of-the-mill Newton or Nesterov optimizer.

However, the third step is slightly more involved. Expanding out the ELBO, we have

$$\begin{aligned}
ELBO(\{\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2\}, \theta, \beta) &= -D_{KL}\Big(q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2)) \Big|\Big| p(\tilde{x}_{i'} | \tilde{y}_{i'}, \theta, \beta) \Big) \\
&= E_{q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2))} \Big[\log q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2)) \Big] - \\
&\qquad E_{q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2))} \Big[\log p(\tilde{x}_{i'} | \tilde{y}_{i'}, \theta, \beta) \Big] \\
&= D_{KL}\Big(N(x_{i'}; \vec{\mu}_{i'}, \vec{\sigma}_{i'}^2) \Big|\Big| N(_{i'}; \vec{\mu}_0, \vec{\Sigma}_0) \Big) - \\
&\qquad E_{q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2))} \Big[\log p(y_{i'} | g(x_{i'}, \theta, \beta) \Big] + c \\
&= \frac{1}{2} \Bigg[\log \frac{|\Sigma_0|}{|diag(\vec{\sigma}_{i'}^2)|} - L + tr\{\Sigma_0^{-1} diag(\vec{\sigma}_{i'}^2)\} + \\
&\qquad (\vec{\mu}_{i'} - \vec{\mu}_0)^T diag(\vec{\sigma}_{i'}^2)^{-1} (\vec{\mu}_{i'} - \vec{\mu}_0) \Bigg] - \\
&\qquad E_{q(\tilde{x}_{i'}; N(\vec{\mu}_{i'}, \vec{\sigma}_{i'}^2))} \Big[\log p(y_{i'} | g(x_{i'}, \theta, \beta)) \Big] + c.
\end{aligned}$$

In order to obtain the gradient w.r.t. either $\vec{\mu}_{i'}$ or $\vec{\sigma}^2_{i'}$, it is easiest to a.) substitute $\vec{\phi}_{i'} = \log\vec{\vec{\sigma}}^2_{i'}$ and b.) differentiate the KL term and the expectation term above separately. Specifically, we have:

$$\begin{aligned}
\nabla_{\vec{\mu}_{i'}} D_{KL}&\left(N(x_{i'};\vec{\mu}_{i'},\vec{\sigma}^2_{i'}) \middle|\middle| N(_{i'};\vec{\mu}_0,\vec{\Sigma}_0) \right) \\
&= \nabla_{\vec{\mu}_{i'}} \frac{1}{2}\Bigg[\log\frac{|\Sigma_0|}{|diag(\vec{\sigma}^2_{i'})|} - L + tr\{\Sigma_0^{-1} diag(\vec{\sigma}^2_{i'})\} + \\
&\qquad (\vec{\mu}_{i'} - \vec{\mu}_0)^T diag(\vec{\sigma}^2_{i'})^{-1}(\vec{\mu}_{i'} - \vec{\mu}_0) \Bigg] \\
&= (\vec{\mu}_{i'} - \vec{\mu}_0)^T diag(\vec{\sigma}^2_{i'})^{-1}.
\end{aligned}$$

and, for $\ell = 1, \ldots, L$

$$\begin{aligned}
\nabla_{\vec{\phi}_{i',\ell}} D_{KL}&\left(N(x_{i'};\vec{\mu}_{i'},\vec{\sigma}^2_{i'}) \middle|\middle| N(_{i'};\vec{\mu}_0,\vec{\Sigma}_0) \right) \\
&= \nabla_{\vec{\phi}_{i',\ell}} \frac{1}{2}\Bigg[\log\frac{|\Sigma_0|}{|diag(\vec{\sigma}^2_{i'})|} - L + tr\{\Sigma_0^{-1} diag(\vec{\sigma}^2_{i'})\} + \\
&\qquad (\vec{\mu}_{i'} - \vec{\mu}_0)^T diag(\vec{\sigma}^2_{i'})^{-1}(\vec{\mu}_{i'} - \vec{\mu}_0) \Bigg] \\
&= \nabla_{\vec{\phi}_{i',\ell}} \frac{1}{2}\Bigg[\log\frac{|\Sigma_0|}{\prod_{\ell=1}^{L}\exp(\phi_{i',\ell})} + \sum_{\ell=1}^{L}\frac{\exp(\phi_{i',\ell})}{\Sigma_{0,\ell,\ell}} \Bigg] \\
&= \frac{1}{2}\Bigg[-1 + \frac{\exp(\phi_{i',\ell})}{\Sigma_{0,\ell,\ell}} \Bigg].
\end{aligned}$$

Lastly, the $\nabla E_{q(\tilde{x}_{i'};N(\vec{\mu}_{i'},\vec{\sigma}^2_{i'}))}\Big[\log p(y_{i'}|g(x_{i'},\theta,\beta) \Big]$ is found via the reparameterization trick, in conjunction with Monte Carlo approximation. Specifically, for a standard normal draw $\epsilon \sim N(0,I)$, we have, recalling $x_{i'} \sim N(\vec{\mu}_{i'}, \mathrm{e\vec{x}p}(\phi_{i'}))$

$$E_{q(\tilde{x}_{i'};N(\vec{\mu}_{i'},\mathrm{e\vec{x}p}(\phi_{i'})))}\Big[\log p(y_{i'}|g(x_{i'},\theta,\beta)) \Big]$$
$$=$$
$$E_{\epsilon\sim N(0,I)}\Big[\log p(y_{i'}|g(\vec{\mu}_{i'} + \epsilon\cdot\mathrm{e\vec{x}p}(\phi_{i'})^{1/2},\theta,\beta)) \Big].$$

In turn, we have for our single ϵ draw

$$\begin{aligned}
\nabla E_{q(\tilde{x}_{i'};}N(\vec{\mu}_{i'},\mathrm{e\vec{x}p}(\phi_{i'})))&\Big[\log p(y_{i'}|g(x_{i'},\theta,\beta)) \Big] \\
&= E_{\epsilon\sim N(0,I)}\Big[\nabla\log p(y_{i'}|g(\vec{\mu}_{i'} + \epsilon\cdot\mathrm{e\vec{x}p}(\phi_{i'})^{1/2},\theta,\beta)) \Big].
\end{aligned}$$

Importantly, we can apply the chain rule to the softmax function to obtain a closed-form

$$\nabla\log p(y_{i'}|g(\vec{\mu}_{i'} + \epsilon\cdot\mathrm{e\vec{x}p}(\phi_{i'})^{1/2},\theta,\beta));$$

however, a numeric approximation is often faster and more stable, so the closed-form expression is left as an exercise. Then, to obtain an approximation of the gradient of the above expectation, we can (1) sample $\epsilon_s \sim_{i.i.d.} N(0,I), s = 1,\ldots,S$; (2) compute a reparameterized gradient via numeric methods above; and (3) average these values for our Monte Carlo gradient.

J. Method II: Semi-Supervised EM

The procedure outlined above can be modified and repeated into a full-fledged semi-supervised EM algorithm. Specifically, this involves

1. Identify MAP

$$\hat{\theta},\hat{\beta} = \arg\max_{\theta,\beta} \ell(Y;X,\theta,\beta) + \log(\pi(\theta)).$$

Again, this initial maximization occurs only on the supervised data – i.e., we are only using data where both the observed outcomes and Statcast fields (as opposed to the observed outcomes but not the Statcast fields) are known to fit $\hat{\theta}$ and $\hat{\beta}$.

2. Initialise variational parameters $\vec{\mu}_{i'},\vec{\sigma}^2_{i'}\ \forall i' \in \tilde{S}$. As before can be done via random initialisation or mean/variance initialisation. If X has been mean-center scaled, then $\vec{\mu}_{i'} = \vec{0}, \vec{\sigma}^2_{i'} = \vec{1}$ is the mean/variance initialisation.

3. For a specified number of iterations and/or until convergence of the ELBO or log-likelihood:

   (a) Maximize the $ELBO(\{\vec{\mu}_{i'},\vec{\sigma}^2_{i'} : i' \in \tilde{S}\},\theta,\beta)$ via gradient ascent, i.e. over all $i' \in \tilde{S}$

$$\vec{\mu}_{i'} := \vec{\mu}_{i'} + \eta\nabla_{\vec{\mu}_{i'}} ELBO(\{\vec{\mu}_{i'},\vec{\sigma}^2_{i'} : i' \in \tilde{S}\},\theta,\beta).$$

   and

$$\vec{\mu}_{i'} := \vec{\sigma}^2_{i'} + \eta\nabla_{\vec{\sigma}^2_{i'}} ELBO(\{\vec{\mu}_{i'},\vec{\sigma}^2_{i'} : i' \in \tilde{S}\},\theta,\beta).$$

   (b) Identify MAP

$$\hat{\theta},\hat{\beta} = \arg\max_{\theta,\beta} \Big[\ell(Y;X,\theta,\beta) + \ell(\tilde{Y};\tilde{X},\theta,\beta) + \log(\pi(\theta)) \Big].$$

   Note here the inclusion of the log-likelihood of the unsupervised data, thus making the overall operation semi-supervised.

K. Method III: Non-Variational Feedforward Network

Lastly, as a reference point for the above variational models, we also fit a simple and strictly-supervised single-layer feedforward network, i.e. the model for $H > 0$:

$$\begin{aligned}
&X_i \sim N((\sigma(Y_i\theta_0 + \beta_1)\theta_1 + \beta_1),\Omega), \\
&\vec{\theta_0} \sim N(\vec{0}, I/M_{0,0}); \quad \vec{\theta_1} \sim N(\vec{0}, I/M_{0,1}) \\
&\beta_1 \sim N(\vec{0}, I/M_{1,0}); \quad \beta_1 \sim N(\vec{0}, I/M_{1,1}) \\
&\Omega \sim InvWishart(K, I/\kappa).
\end{aligned}$$

Note here that $H, M_{i,j}$, and κ are hyperparameters, and that $\theta_0 \in \mathbb{R}^{K,H}, \theta_1 \in \mathbb{R}^{H,L}, \beta_0 \in \mathbb{R}^H, \beta_1 \in \mathbb{R}^L$. Additionally, σ is a standard activation function – here, $\sigma = tanh$. For expediency and/or tuning purposes, one might drop the Ω term and simply find

$$\begin{aligned}
\theta_0,\beta_0,\theta_1,\beta_1 = \arg\min_{\theta_0,\beta_0,\theta_1,\beta_1} &\sum_{i\in\mathcal{S}}\sum_{L}\mathcal{L}\Big(X_i, \sigma(Y_i\theta_0 + \beta_1)\theta_1 + \beta_1 \Big) + \\
&\frac{1}{2M_{0,0}}||\theta_0||_2^2 + \frac{1}{2M_{0,1}}||\theta_1||_2^2 + \\
&\frac{1}{2M_{1,0}}||\beta_0||_2^2 + + \frac{1}{2M_{1,1}}||\beta_1||_2^2.
\end{aligned}$$

This would be much less expensive for tuning.

Again, this final model does not include a variational approximation of any kind – it is simply a baseline trained on X, Y that can be applied to the unsupervised data $\tilde{X}, \tilde{Y}$ to obtain a very basic point of reference.

3. Model Fitting

A. Overview

To fit the model, choose Y_i (respectively, $\tilde{Y}_{i'}$

$$Y_i = \begin{bmatrix} Strikeouts_i \\ Walks_i + HBP_i \\ IFH_i \\ 1BGB_i \\ 1BAB_i \\ 2BGB_i \\ 2BFB_i \\ 3B_i \\ HR_i \\ GBOut_i \\ LDOut_i \\ FBOut_i \\ PUOut_i \end{bmatrix} := \begin{bmatrix} \textit{\# Strikeouts} \\ \textit{\# Walks + Hit Batsmen} \\ \textit{\# Infield Singles} \\ \textit{Non-Infield Groundball Singles} \\ \textit{\# Non-Infield Airball Singles} \\ \textit{\# Groundball Doubles} \\ \textit{\# Airball Doubles} \\ \textit{\# Triples} \\ \textit{\# Home Runs} \\ \textit{\# Ground Ball Outs} \\ \textit{\# Line Drive Outs} \\ \textit{\# Fly Ball Outs} \\ \textit{\# Pop Up Outs} \end{bmatrix}.$$

That is, these were the player/season multinomial outcomes. For the Statcast fields $X, \tilde{X}$, I chose a limited number of fields for this initial analysis, to wit

$$X_i = \begin{bmatrix} \bar{EV}_i \\ \bar{LA}_i \\ xw\bar{OBA}_i \\ x\bar{SLG}_i \\ x\bar{BA}_i \end{bmatrix} := \begin{bmatrix} \textit{Avg. Exit Velocity} \\ \textit{Avg. Launch Angle} \\ \textit{Avg. Expected Weighted-on-Base} \\ \textit{Avg. Expected Slugging Pct.} \\ \textit{Avg. Expected Batting Avg.} \end{bmatrix}$$

Again, these feature choices deserve scrutiny in future iterations of this model. The categories of the multinomial component were chosen simply for their loose correspondence to batted ball type and direction; the X fields were selected a.) for their ubiquity/popularity in contemporary baseball analyses, and b.) in limited fashion, for a simpler initial model.

B. Data

All data was obtained from Baseball Savant, a website maintained by Major League Baseball Advanced Media. In order for a player's season to be included in the modeling process, Statcast must have tracked at least 75 batted balls by the player, during a regular season games in the season of interest.

C. Tuning

All three versions of the model were tuned on the 2016 and 2017 seasons, with the 2018 season held out as the validation set. For each of Methods I-III, following table summarizes: a.) the grid of hyperparameters searched during tuning, b.) the ultimate values chosen, and c.) validation-set scoring for those best parameters:

Figure 1 Tuning Grid (Final Parameters in Bold)

Method	Grid
I: MAP + GA on ELBO	$\{\lambda_2 : [.1, .01, \mathbf{.005}, .001], \alpha : [100, \mathbf{1}, .01]\}$
II: EM	$\{\lambda_2 : [.1, \mathbf{.01}, .005, .001], \alpha : [100, \mathbf{1}, .01]\}$
III: FFNN	{H: [5, **10**, 15]

(a) Note that α reflects the relative weighting between the KL divergence component of the ELBO and the log-probability component of the ELBO. In most ELBO optimization settings, this hyperparameter is tuned exhaustively; the grid here is much coarser due to computational considerations.

The convergence of methods I-II can be seen through the following ELBO plots:

Figure 2 ELBO Convergence, Method I

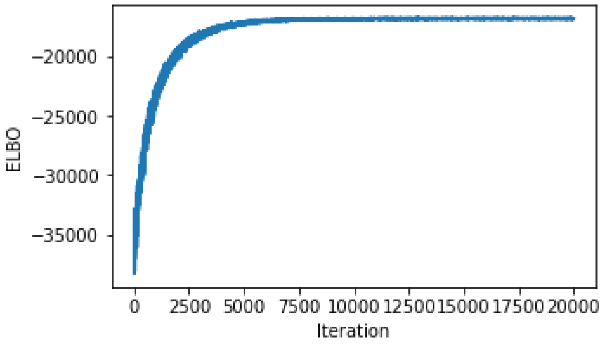
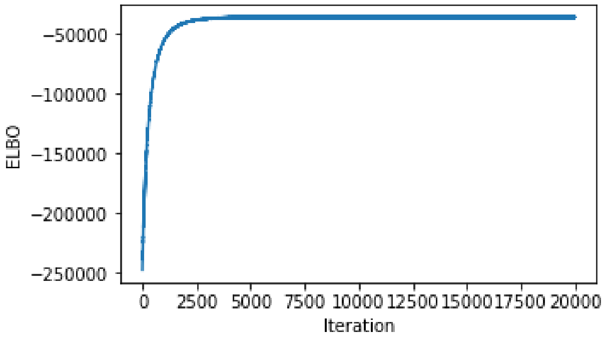


Figure 3 ELBO Convergence, Method II



(a) Note that in the semi-supervised setting, the inclusion of the supervised log loss (as opposed to just the unsupervised log loss) changes the ELBO values achieved.

While Methods I-II have "from-scratch" implementations provided in the attached code, the final tuning/fitting process leveraged the TFP library.

4. Performance

With the hyperparameters set forth above, Methods I-III were then fit on the entirety of the 2016-2018 data set, with 2019 used as the test set. For interpretability purposes, the principal evaluation criterion here was residual mean squared error (RMSE), as evaluated in the fields' original units. The results for the three methods are presented seriatim.

A. Method I (MAP + Gradient Ascent on ELBO)

A.1. Pointwise Errors We next examine the weighted (by PA) residual mean-squared error, the weighted mean absolute error, and the weighted residual for the posterior means of our variational approximations.

Figure 4 Method I: Posterior Mean Losses

Loss	Avg EV	Avg LA	xwOBA	xSLG	xBA
wRMSE	2.811	1.801	0.021	0.045	0.018
wMAE	2.288	1.427	0.017	0.035	0.014
wResidual	1.607	-0.710	0.005	0.007	0.005

Figure 5 Method I: 90% Posterior Predictive Capture Rate

90% P.P.	Avg EV	Avg LA	xwOBA	xSLG	xBA
Capture Rate	.902	.995	.861	.875	.838

(a) Entries in this table reflect the frequency at which test data fell in between a 90% posterior predictive interval. This rate is determined by $S = 10,000$ posterior predictive samples.

A.2. Posterior Predictive Checks

A.3. Selected Cases For the Method I (MAP + Gradient Ascent on ELBO), we plot the posterior predictive average launch angle and average exit velocity from testing for four of the brightest young stars in baseball: Andrew Knapp, Rhys Hoskins, Bryce Harper, and Cesar Hernandez. Note here that the color bar reflects the approximate proportion of posterior predictive draws in the range of the relevant Statcast field, while the red point reflects the player's true 2019 (test set) performance.

Figure 7 Posterior Predictive Avg. Exit Velo, Select Players

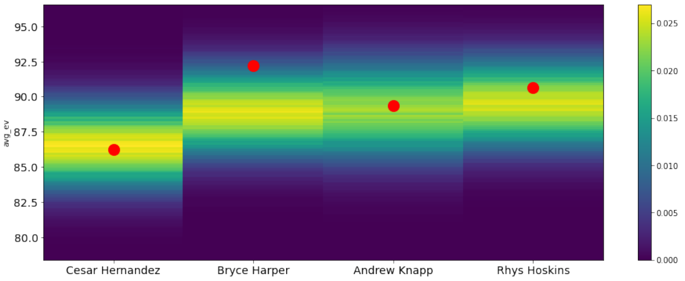
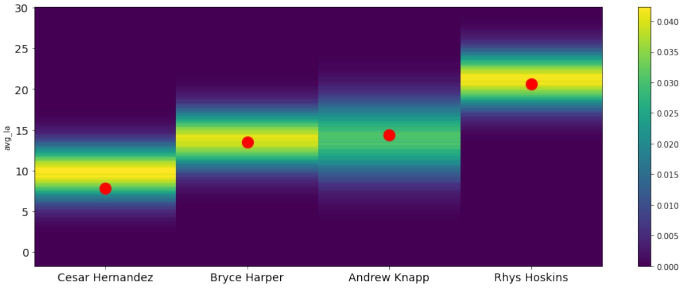


Figure 8 Posterior Predictive Avg. Launch Angle, Select Players



As we see, this method performs respectably for point estimates, but the posterior predictive checks leave much to be desired. In particular, the posterior launch angles are distributed far too wide, while the "x-stats" are distributed far too narrowly. In other words, the Gaussians that are the variational posterior approximations appear to be too fat-tailed about the average launch angles, and too narrow about the "x-stats."

B. Method II (EM)

B.1. Pointwise Errors We perform a similar analysis for Method II.

Figure 9 Method II: Posterior Mean Losses

Loss	Avg EV	Avg LA	xwOBA	xSLG	xBA
wRMSE	2.724	2.019	0.023	0.057	0.019
wMAE	2.185	1.599	0.019	0.046	0.015
wResidual	122	-0.578	0.007	0.009	0.006

Figure 10 Method II: 90% Posterior Predictive Capture Rate

90% P.P.	Avg EV	Avg LA	xwOBA	xSLG	xBA
Capture Rate	.881	.995	.914	.901	.854

(a) Entries in this table reflect the frequency at which test data fell in between a 90% posterior predictive interval. This rate is determined by $S = 10,000$ posterior predictive samples.

B.2. Posterior Predictive Checks Pointwise, while the EM approach performs better for average exit velocity, it generally does worse in all other areas. The posterior predictive capture rate is slightly better for xwOBA and xSLG, though the same issues persist for average launch angle and xBA as before.

B.3. Selected Cases We use the same players for the selected cases. As we see, the Method II (EM) struggles in comparison, particularly for average exit velocity. Again, the color bar reflects the approximate proportion of posterior predictive draws in the range of the relevant Statcast field, while the red point reflects the player's true 2019 (test set) performance.

Figure 12 Posterior Predictive Avg. Exit Velo, Select Players

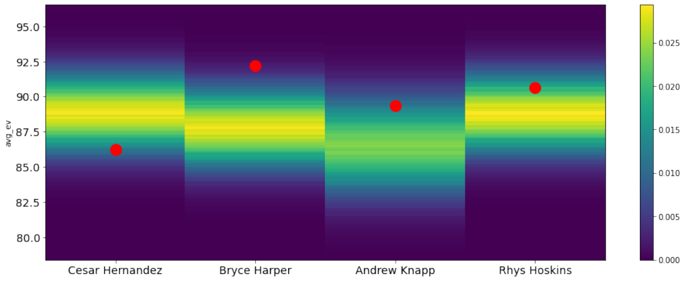
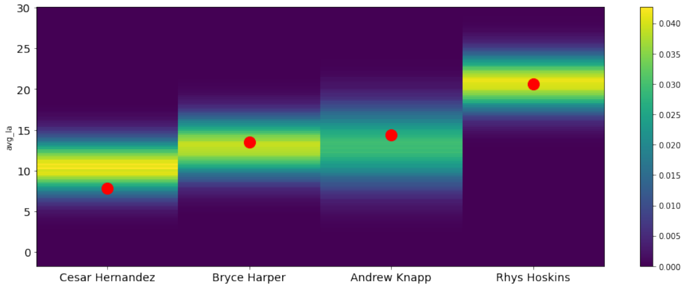


Figure 13 Posterior Predictive Avg. Launch Angle, Select Players



C. Method III (FFNN)

C.1. Pointwise Errors Finally, we repeat a similar analysis for the simple neural network.

Figure 14 Method III: Posterior Mean Losses

Loss	Avg EV	Avg LA	xwOBA	xSLG	xBA
wRMSE	2.960	2.261	0.024	0.047	0.013
wMAE	2.537	1.849	0.020	0.037	0.010
wResidual	2.274	-198	-0.001	0.009	-0.002

Figure 15 Method III: 90% Posterior Predictive Capture Rate

90% Post. Pred.	Avg EV	Avg LA	xwOBA	xSLG	xBA
Capture Rate	0.519	0.928	0.847	0.868	0.940

(a) Entries in this table reflect the frequency at which test data fell in between a 90% posterior predictive interval. This rate is determined by $S = 10,000$ posterior predictive samples.

C.2. *Posterior Predictive Checks*

C.3. Selected Cases Once more, use the same players for the selected cases. As before, the color bar reflects the approximate proportion of posterior predictive draws in the range of the relevant Statcast field, while the red point reflects the player's true 2019 (test set) performance.

Figure 17 Posterior Predictive Avg. Exit Velo, Select Players

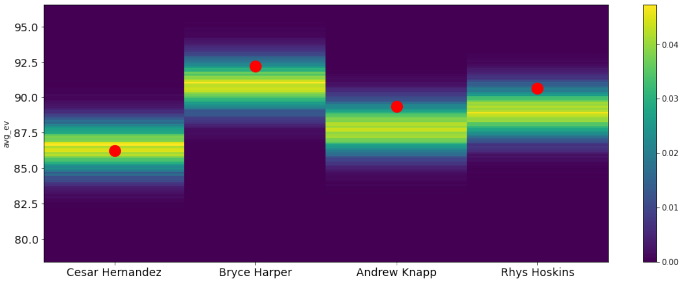
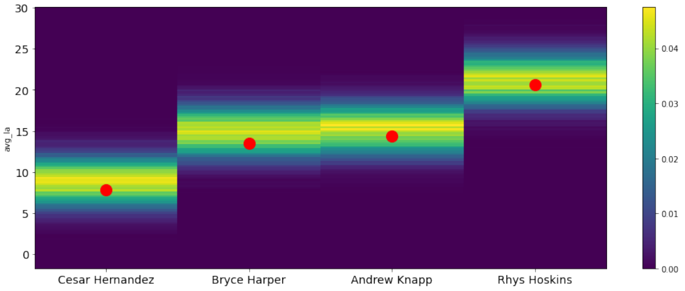


Figure 18 Posterior Predictive Avg. Launch Angle, Select Players



As we see, this method is a close second to Method I – however, its posterior predictive intervals for average exit velocity are egregiously narrow. Furthemore, it trails slightly on the majority of the wMAE and wRMSE losses. In this way, it has adequately served its role as a stable benchmark.

Of course, given this network's simplicity, one must wonder whether a deeper network might not only outperform this simpler one, but also the variational models. This is a principal question for future iterations of this research.

D. *Discussion*

On balance, the Method I (MAP MN-Logit + Gradient Ascent on ELBO) was the winner. And for a first pass, this model did a respectable job with respect to point predictions for the 2019 test-set season. However, there are numerous ways to improve the model, which likely include:

- The inclusion of additional static, non-variational features in the multinomial logit model. For instance, features such as park factors [5], baseball properties[6], or defensive positionings[7] may help to explain discrepancies between and/or additional variance in contact quality and observed results. In this way, the multinomial logit component would be more robust – ideally leading to variational approximations that are better "sheltered" from the above sources of noise.
- Encouraging "wider" variational approximations. As shown by the posterior predictive checks, the 90% capture rate performance of the variational approximations was poor, with the variational densities distributed too narrowly about their respective means. There are at least three obvious corrections to try here, including:
  1. Using a fatter-tailed variational assumption – most likely a Student's T-distribution – instead of a Gaussian.
  2. Dropping the diagonalized covariance assumption in the variational step, in favor of a full covariance matrix. For this paper, we used the assumption that the five Statcast fields are independent strictly for computational ease – in actuality, these fields are surely correlated. Perhaps correcting this elision via fully-specified covariances could improve the intervals returned by the model.
  3. Introducing a hierarchical component/partial pooling component to the variational covariances. By (partially) sharing covariances across players-season pairs, the variational densities might not situate so narrowly about their means, thereby inducing the wider and more realistic intervals we desire.
- Exploring different weightings between KL Divergence and log probability in the ELBO, as well as different weightings between ELBO and supervised likelihood in the EM case. For time/computation reasons, the KL Divergence and log probabilities were left unweighted here; however, such hyperparameters are routinely tuned in other semi-supervised settings, and deserve such treatment in future iterations of this analysis.

In all, the methods here outline a starting point for more refined variational approximations. In pursuing the methods/"next-steps" set forth above, one could hopefully achieve more robust approximations.

5. Conclusion

This paper uses simple variational inference techniques to approximate averages of Major League Baseball Statcast data, for seasons

[5] Stadiums often impact the results of batted balls. For example, Coors Field in Colorado (on account of the elevation) is more favorable to hitters, while Oracle Park in San Francisco (on account of the wind, temperature, and relatively deep fences), is more favorable to pitchers. So two identically hit baseballs in Denver and San Francisco could have vastly different outcome probabilities, despite being fundamentally the same. This sort of engineering would likely bolster the multinomial logit model

[6] In recent years, Major League Baseball is understood to have altered the physical composition of the baseballs in recent seasons, with baseballs from late 2018 and the entire 2019 having aerodynamic properties more favorable to hitters. Again, this could create discrepancies between tracking measurements and outcomes, and controlling for such factors could improve the accuracy of the multinomial logit model.

[7] Defenses who position themselves well (perhaps by "shifting") against opposing hitters are more likely to convert harder-hit ground balls and line drives into outs; again, this would improve the mapping of batted ball measurement to observed outcome.

for which such data does not exist. The methods presented appear promising in terms of posterior point estimation, though further research is necessary fully Bayesian posterior intervals.

6. References

Works Cited

"A new way to dissect baseball's park factors". *MLB.com* () 5 June 2021 <https://www.mlb.com/news/park-factors-measured-by-statcast>.

Arthur, Rob and Tim Dix. "We X-Rayed Some MLB Baseballs. Here's What We Found." *FiveThirtyEight* (Mar. 2018) 5 June 2021 <https://fivethirtyeight.com/features/juiced-baseballs/>.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". *Journal of the American Statistical Association* 112.518 (Apr. 2017). arXiv: 1601.00670: 859–877 5 June 2021 <http://arxiv.org/abs/1601.00670>.

Linderman, Scott. "STATS371 Lecture Notes: Factor Analysis, VAEs, and Variational EM" (). <https://github.com/slinderman/stats271sp2021/blob/main/slides/lap6_vaes.pdf>.

"PyMC3 Inference" (2018). publisher: The PyMC3 Development Team. <https://docs.pymc.io/api/inference.html>.

Scott, Linderman. "STATS371 Lectures Slides: Mixed Membership Model, Topic Models, and Variational Inference" (). <https://github.com/slinderman/stats271sp2021/blob/main/slides/lap5_mixed_membership.pdf>.

"TFP Probabilistic Layers: Variational Auto Encoder". *TensorFlow* () 5 June 2021 <https://www.tensorflow.org/probability/examples/Probabilistic_Layers_VAE>.

Webster, Kevin. "Maximising the ELBO - Variational Autoencoders". *Coursera* (). Used this video to learn alternative constructions of VAEs within TFP. 5 June 2021 <https://www.coursera.org/lecture/probabilistic-deep-learning-with-tensorflow2/maximising-the-elbo-yDXSt>.

Wiecki, Thomas. "Variational Inference: Bayesian Neural Networks — PyMC3 3.10.0 documentation". *PyMC3* (). publisher: The PyMC3 Development Team 5 June 2021 <https://docs.pymc.io/notebooks/bayesian_neural_network_advi.html>.

Willman, Daren. "Baseball Savant" (June 2021). publisher: Major League Baseball Advanced Media. <2021/06/04>.