Predicting Productivity of Hockey Players via Mixture Models (Empirical Bayes Methodology)

by

Connor Campbell

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mathematical and Statistical Sciences

University of Alberta

Abstract

The objective of this thesis is to show how advanced methods based on mixture models can be used to predict the productivity of hockey players, measured by the rate at which they produce goals and assists. The performance of the methods is evaluated on existing data from one full National Hockey League (NHL) season. Over a large time frame, such predictions come fairly easily, regardless of the method we choose. However, our focus is on predictions obtained from relatively – sometimes even significantly – short sampling periods. If we look solely at the first 3-5 weeks of the season, the naïve estimator, based on maximum likelihood, is essentially useless at predicting how someone will perform for the remainder of the year. Simple methods such as "one-fits-all" estimators and naïve shrinkage estimators represent an improvement, but it turns out we can do better. We look at both parametric and nonparametric empirical Bayes approaches to fitting mixture models, with the objective of showing that these methods provide good predictions in a small time frame. In particular, we will cover two competitive approaches, the Poisson-Gamma parametric model and the Kiefer-Wolfowitz nonparametric method. Both of them construct certain mixtures of Poisson distributions, but contrary to the setting prevailing in the literature, we have to deal with the fact that our Poisson outcomes are for different players observed over different time periods, depending on the number of games played, or the total amount of time spent on ice.

Acknowledgements

I would first like to thank my supervisor, Dr. Ivan Mizera, for his continuous support and guidance over the course of my Masters program. Without his knowledge and expertise much of this work would not have been possible.

I express my gratitude to Dr. Adam Kashlak and Dr. Rohana J. Karunamuni for being a part of my defence committee.

I would also like to thank my parents, Stephen and Carolyn Campbell, and my brother, Brendan Campbell, for their help and encouragement over the last two years.

Lastly, I would like to give a special thanks to my girlfriend, Katelyn McEwen, for her unwavering love and support. These last two years were made infinitely easier with her by my side, and I am extremely grateful for everything she has done for me.

Contents

List of Tables

List of Figures

Chapter 1

Predicting Hockey Productivity

The aim of this chapter is to familiarize ourselves with a variety of statistical methods for predicting scoring rates in hockey. Sections 1.1 and 1.2 start us along this path by outlining the problem at hand and introducing Poisson processes. Then, in Section 1.3, we look at simple methods for making scoring rate predictions. Section 1.4 covers more complex methods based on mixture models.

1.1 The Outline of the Problem

With the rise of analytics in hockey, many new metrics have revolutionized the way players are evaluated. Despite these recent developments, simple counting statistics like goals, assists, and points (goals plus assists) are still some of the first things that hockey fans gravitate towards when assessing the talent level of individual players.

A goal is scored when the puck completely crosses the goal line and goes into the net. The last player to touch the puck before it goes in is awarded the goal, and assists are attributed to a maximum of two teammates of the goal scorer who were the last ones to touch the puck in between the opposing teams most recent possession and the goal. A primary assist is given to the player that passed the puck directly to the goal-scorer, while a secondary assist is given to the player who passed the puck to the primary assistant.

The objective of this thesis is to show how advanced methods based on mixture models can be used to predict the productivity of hockey players,

measured by the rate at which they produce goals and assists. The performance of these methods is evaluated on existing data from one full National Hockey League (NHL) season; in particular, we will look at the data from 2018-19, which was the most recent non-shortened NHL season at the time of this writing.

Over a large time frame, desired predictions come fairly easily, regardless of the method we choose. However, our specific objective is to obtain good predictions from relatively – sometimes even significantly – short sampling periods. There are various ways to make these predictions, some of them based on expert knowledge, but in this thesis we concentrate on prediction strategies that depend solely on the observed data, assuming minimal knowledge of certain other covariates (injury history, rookie status, team quality, production in previous years, etc.).

A first obvious thing to do is to try the *naïve estimator*, based on maximum likelihood; this is as simple as taking a player's scoring totals from the first batch of games, and dividing by their number of games played or their total time on ice. It turns out this is generally the worst option, especially when the number of weeks used to make predictions is small.

It may be a surprise to many that one can actually make better predictions by simply applying the average leaguewide scoring rate to each player. This "one-fits-all" estimator may improve results for the majority of players, but not everyone is average. For example, the best players in the league can be expected to surpass league average by a wide margin, while a bottom of the roster player is unlikely to score many points at all. This inability to capture the outer edges of our dataset make predicting from the overall average suboptimal.

We can combat this issue by using a shrinkage estimator. As the name suggests, these estimators "shrink" predictions towards certain "common values", and often provide more accurate predictions than the aforementioned methods. The celebrated example of a shrinkage estimator is the James-Stein estimator.

When moving to more complex methods we find that we are able to pro-

duce more desirable results. This thesis looks at parametric and nonparametric empirical Bayes approaches to fitting mixture models, and evaluates their performance on the real data from 2018-19 NHL season. In particular, we will cover two competitive approaches, the Poisson-Gamma parametric model and the Kiefer-Wolfowitz nonparametric method. Both of them construct certain mixtures of Poisson distributions, but contrary to the setting prevailing in the literature, we have to deal with the fact that our Poisson outcomes are for different players observed over different time periods, depending on the number of games played, or the total amount of time spent on ice. To account for differences in games played, we introduce a new variable, g, which we refer to as a time epoch; this can be either the number of games played, or the total time spent on ice. The setting incorporating g differs from that in previous literature: for instance, Robbins (1956) and Maritz and Lwin (1970), consider always the Poisson model exclusively with $g = 1$. In our case, each player, j, is given an individual time epoch, g_j, that corresponds to the number of games they played, or time they spent on ice, in the chosen time span.

An important distinction to note is the difference in usage between forwards and defencemen. In hockey, forwards are typically expected to be the ones scoring most of the goals, while defencemen are primarily tasked with keeping the puck out of their own net. While defencemen are less likely to score, they still often pick up assists due to their role of moving the puck up ice. This leads to a disparity between goal and assist totals for defencemen that does not exist for forwards. Furthermore, while there are obviously some outliers, the vast majority of the leading point-getters will be forwards. Therefore, combining both positions when making predictions will likely lead to noisy results. Due to this difference in behaviour, we postulate that predictions can be further improved if we make separate datasets for forwards and defencemen. Figure 1.1 shows the relationship between goal and assist totals when forwards and defencemen are separated. The two panels indeed illustrate different behaviour for each of the groups.

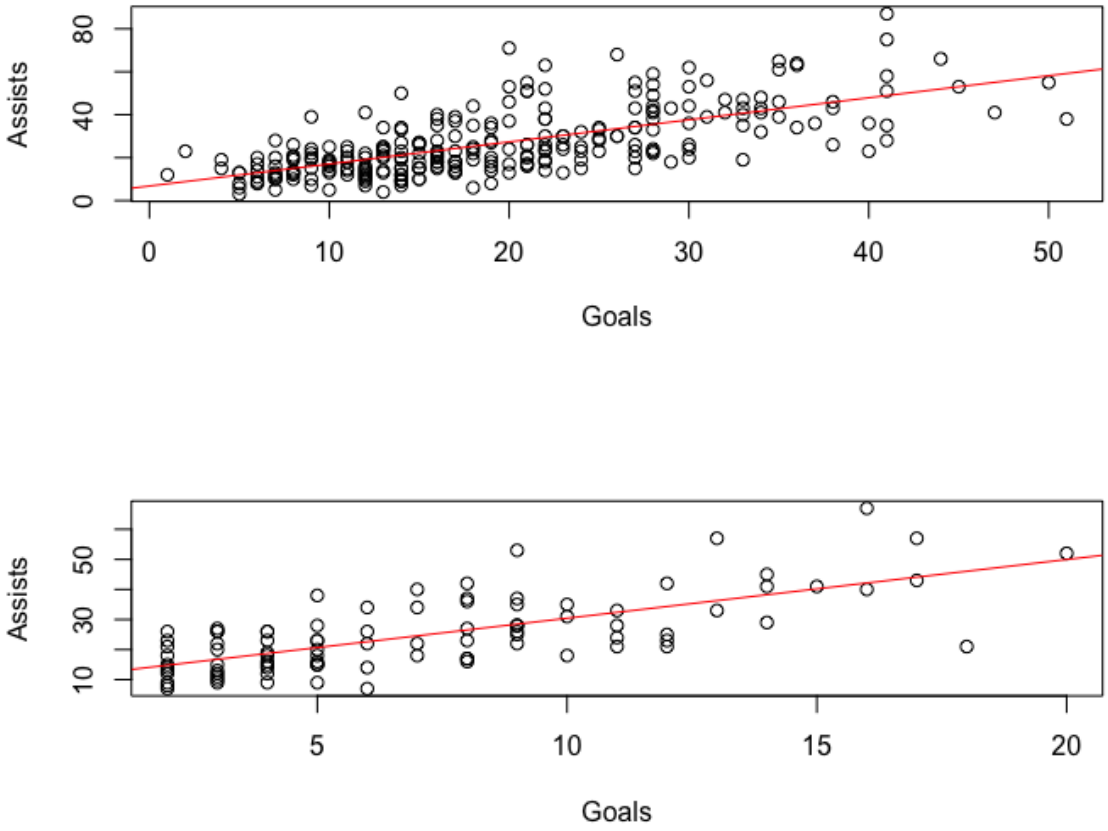

Figure 1.1: *Top.* Scatterplot of assist vs. goal totals for forwards. *Bottom.* Scatterplot of assist vs. goal totals for defencemen (2018-19 NHL regular season, min. 60 games played).

While the separate panels of Figure 1.1 indicate a positive correlation between goals and assists, the combined plot (forwards and defencemen together) in Figure 1.2 exhibits rather low correlation – in fact, a simple linear regression yielded an adjusted R-squared value of just 0.3521. Therefore, while it may be of interest to consider prediction of goals and assists simultaneously (perhaps in future work), in this thesis we opted for a simpler approach, which was to make predictions for players' goal and assist rates independently.

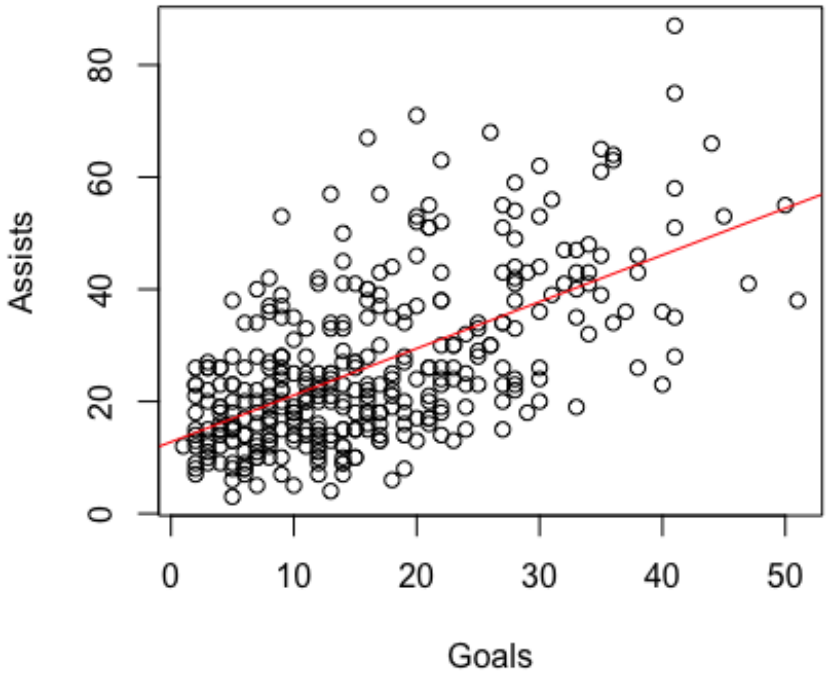

Figure 1.2: Scatterplot of assist vs. goal totals for all NHL players during the 2018-19 regular season (min. 60 games played).

Predictions will be evaluated using the mean squared error loss function

$$MSE(\lambda) = \frac{1}{n}\sum_{j}^{n}(\hat{\lambda}_j - \lambda_j)^2, \tag{1.1}$$

where n represents the number of players in the sample, $\hat{\lambda}_j$ represents the estimate derived from player j's first z weeks of the season, and λ_j represents the validation set, which, for now, will be player j's full season rate minus the rate from their first eight weeks. Obviously, the smaller this quantity is, the better the prediction. It should be stressed, however, that the predictions are formed exclusively from the data available in the first z weeks; knowledge of the remainder of the season is used exclusively for their evaluation.

A variety of other loss functions could have been used here. For instance, the following loss function, proposed by Clevenson and Zidek (1975), "penal-

izes heavily for bad estimates when the λ_j's are small":

$$\ell(\hat{\lambda}, \lambda) = \frac{1}{n} \sum_{j}^{n} \lambda_j^{-1} (\hat{\lambda}_j - \lambda_j)^2$$

Alternatively, one could formulate a loss function that penalizes heavily for bad estimates when the λ_j's are large:

$$\ell(\hat{\lambda}, \lambda) = \frac{1}{n} \sum_{j}^{n} \lambda_j (\hat{\lambda}_j - \lambda_j)^2. \tag{1.2}$$

In professional hockey, where teams put significant care into saving money by not overpaying players, the loss function given in (1.2) might be of more use because it prevents bad overestimates of scoring rates from being made. Also, this loss function stresses the importance of accurately predicting the performance of the best players more than the performance of the worst players.

In the end, we chose MSE because it is standard in literature, and gives us the posterior mean (a concept introduced in Section 1.4.1) when minimized, making it analytically convenient. When separating forwards and defencemen we calculate the overall MSE by

$$MSE(\lambda) = \frac{1}{n_F + n_D} \left[\sum_{j \in F} (\hat{\lambda}_j - \lambda_j)^2 + \sum_{j \in D} (\hat{\lambda}_j - \lambda_j)^2 \right],$$

where n_F represents the number of forwards in the sample, and n_D represents the number of defencemen in the sample.

1.2 Poisson Processes

A typical example of a Poisson process is the arrival of cars to a toll booth. We think of the arrival of each car as a discrete event occurring at a random time, but at a known average rate. It turns out that scoring events in hockey can be modelled the same way, where the occurrence of goals and assists is akin to the arrival of a car. We elect to model two separate Poisson processes due to the differing parameters associated with goals and assists. We also recognize

that each individual player in our dataset scores at differing rates, so we are in essence observing a large number of Poisson processes simultaneously. The somewhat informal axioms of the Poisson process from Lawler (2006) stipulate

(1) The number of events during one time interval does not affect the number of events in a different (non-overlapping) time interval.

(2) The "average" rate at which events occur remains constant.

(3) Events happen one at a time.

It is clear to see that axiom (1) is satisfied. In more exact terms, it is expressed as

(1′) The number of events occurring in intervals $[t, s]$ and $[u, v]$, $t < s \leqslant u < v$, are independent random variables.

Due to the random nature of scoring events, it is hard to imagine that axiom (2) will be wholly satisfied, but for our purposes we choose to accept that the productivity of an individual player remains at least approximately constant. Axiom (2) can be written more formally as follows:

(2′) The distribution (probability law) of the random variable recording the number of events happening in an interval $[t, s]$, $t < s$, depends only on the length, $s - t$, of this interval.

Axiom (3) is also acceptable, and is more formally phrased as

(3′) The probability of two events happening in an interval $[t, s]$, $t < s$, is $o(s - t)$ – that is, $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$, where $\Delta t = s - t$.

Axioms (1')-(3') imply that the random variable, X, recording the number of events (goals or assists) happening in an interval $[t, s]$, $t < s$, follows a Poisson distribution:

$$P[X = x] = \frac{(\lambda(s - t))^x e^{-\lambda(s-t)}}{x!}, \quad x = 0, 1, 2, ...,$$

where λ represents the rate at which events occur. Seeing as hockey players are not on the ice at all times, the interval $[t, s]$ will be measured using the discrete time process of games played (g), rather than the continuous time process described above. Therefore, we can rewrite the Poisson distribution as

$$P[X = x] = \frac{(\lambda g)^x e^{-\lambda g}}{x!}, \quad x = 0, 1, 2, ...$$

The expected value of X is:

$$E[X] = \lambda g.$$

Of interest to us is the expected number of events, X, per game. In the Poisson model this is expressed by

$$E\left[\frac{X}{g}\right] = \lambda.$$

However, the non-integer values of X/g necessitates us to work with X instead. Other important properties of the distribution of X are

$$Var[X] = \lambda g = E[X].$$

and

$$E[X^2] = Var[X] + E[X]^2 = \lambda g + (\lambda g)^2$$

Looking through a Bayesian lens, we now think of the distribution of the productivity of an individual player in g games as a conditional distribution of X given $\Lambda = \lambda$:

$$P[X = x|\Lambda = \lambda] = \frac{(\lambda g)^x e^{-\lambda g}}{x!}, \quad x = 0, 1, 2, ..., \tag{1.3}$$

If Λ has the distribution Q, the joint distribution of X and Λ can be written, with some abuse of notation, as

$$P[X = x, \Lambda = \lambda] = P[X = x|\Lambda = \lambda]\, dQ(\lambda) = \frac{(\lambda g)^x e^{-\lambda g}}{x!}\, dQ(\lambda)$$

and the mixing distribution is

$$P[X = x] = \int \frac{(\lambda g)^x e^{-\lambda g}}{x!}\, dQ(\lambda).$$

1.3 Simple Methods of Prediction

In this section we introduce several simple methods which constitute the first take on the prediction task in our situation. Specifically, these methods are the naïve estimator (Subsection 1.3.1), "one-fits-all" estimators based on maximum likelihood estimation (MLE) and the method of moments (MM) (Subsection 1.3.2), and the "poor man's shrinkage" estimator (Subsection 1.3.3).

1.3.1 The Naïve Estimator

Let $X_1, ..., X_n$ be independent identically distributed (iid) random variables with probability density functions $f(x_i; \lambda)$ for all $i = 1, ..., n$. The likelihood function refers to the joint probability of these $X_i's$ and can be written as:

$$L(\lambda_1, ..., \lambda_n; X_1, ..., X_n) = \prod_{i=1}^{n} f(x_i; \lambda_i). \tag{1.4}$$

As is obvious from the name, the objective of maximum likelihood estimation is to find λ that maximizes (1.4). Hence, this method is a good starting point in our effort to find the best possible estimate for NHL players' scoring rates. We can use maximum likelihood estimation to predict each individuals scoring rate (the naïve estimator), as well as the leaguewide scoring rate in the following way:

Let $X_j \sim \text{Poisson}(\lambda_j g_j)$, where X_j represents the scoring metric of interest, λ_j represents an individual's scoring rate, and g_j represents their number of

games played. Then, using (1.3) given in Section 1.2, we obtain the likelihood function

$$L(\lambda_i, ..., \lambda_n; X_1, ..., X_n) = \prod_j \frac{(\lambda_j g_j)^{x_j} e^{-\lambda_j g_j}}{x_j!}. \tag{1.5}$$

To find the naïve estimate, we maximize the likelihood of each player individually

$$L(\lambda_j; X_j) = \frac{(\lambda_j g_j)^{x_j} e^{-\lambda_j g_j}}{x_j!}. \tag{1.6}$$

For simplicity, we take the negative of the natural logarithm of (1.6) to obtain the log-likelihood function

$$\ell(\lambda_j; X_j) = \lambda_j g_j - x_j \log(\lambda_j) - x_j \log(g_j) + \log(x_j!).$$

Next, we take the partial derivative with respect to λ_j and set the result equal to zero

$$\frac{\partial}{\partial \lambda_j} \ell(\lambda_j; X_j) = g_j - \frac{x_j}{\lambda_j} = 0.$$

A quick manipulation gives us the MLE

$$\hat{\lambda}_j = \frac{x_j}{g_j}.$$

As we can see, the naïve estimate for a player's end of season scoring rate is simply their scoring rate from the first z weeks of the season divided by the number of games (or time on ice) they played in those z weeks.

1.3.2 One-fits-all Estimators

We can find an estimate for the rest of season leaguewide scoring rate by setting $\lambda_j = \lambda$. This is considered a "one-fits-all" estimate, and is designed to

obtain relatively conservative results. There are two ways to get a "one-fits-all" estimate, the first being the method of moments, which simply sets λ as the mean of the naïve estimator:

$$\hat{\hat{\lambda}} = \frac{1}{n}\sum_{j}^{n}\frac{x_j}{g_j}.$$

The other option is to once again use maximum likelihood estimation. Recalling (1.5), and again taking the log-likelihood, we get

$$\ell(\lambda; X_j) = \sum_{j}\Big[\lambda g_j - x_j\log(\lambda) - x_j\log(g_j) + \log(x_j!)\Big].$$

Taking the same steps as before, we soon find the MLE

$$\hat{\hat{\hat{\lambda}}} = \frac{\sum_j x_j}{\sum_j g_j}.$$

No more complicated than the preceding MLE, the estimate for the overall mean is just the overall amount of goals or assists accumulated leaguewide divided by the total number of games that have been played collectively.

1.3.3 A "Poor Man's Shrinkage" Estimator

For fitting mixtures of normal distributions, the celebrated James-Stein estimator shrinks naïve predictions towards the common mean, utilizing sophisticated theory to estimate the amount of shrinkage. While, we do not pursue such an elaborate development in the Poisson case, we include as the last simple prediction, the "poor man's shrinkage" estimator, which takes one of the "one-fits-all" estimators (in this case, the maximum likelihood estimator) and shrinks the naïve ones towards them. We select the shrinkage amount in an ad hoc manner to be 0.5, and obtain the following estimator:

$$\hat{\hat{\hat{\lambda}}}_j = (1 - 0.5)\hat{\lambda}_j + 0.5\hat{\hat{\hat{\lambda}}},$$

where $\hat{\lambda}_j$ and $\hat{\hat{\hat{\lambda}}}$ refer to estimates found in Subsections 1.3.1 and 1.3.2.

1.4 Mixture Models

Better methods than those mentioned above can be developed through the use of mixture models. These models can be best described within the framework of the empirical Bayes methodology. In other words we borrow the Bayesian paradigm to describe the setting of mixture models, while still remaining on the ground of frequentist models. To this end we recall some notions from Bayesian statistics.

1.4.1 Bayesian Paradigm

A study is created with the aim to find the proportion of green-eyed people in each Albertan city and town. We postulate that a random sample of 1000 people from each location will be large enough to draw conclusions from. Suppose that after compiling the data, we find the sample proportion of green-eyed people in Edmonton to be 0.40, while everywhere else had sample proportions ranging from 0.10 to 0.15. This disparity calls into question the reliability of our Edmonton sample, and suggests that a different strategy might be more appropriate here.

One option would be to indirectly estimate Edmonton's proportion by borrowing information from our samples of the surrounding areas. This has the benefit of achieving results more congruent with what we expected, but we may not want to entirely discount the information gathered in our original sample. Therefore, taking a weighted average of the direct and indirect estimates is preferable here. Given that Edmonton's current proportion is not in line with the rest of the province, it is fair to say that the indirect estimate should be weighted more heavily. As the size of our Edmonton sample increases, our direct estimate becomes more reliable, so its weight should increase in turn. This emphasis on the importance of indirect prior information in estimation provides the basis for Bayesian statistics.

Bayesian analysis applies the logical assumption that an experimenter can use past experience to influence future decisions. By specifying a prior (or, also, a mixing distribution), the experimenter is able to let "data from a single

trial report add to available evidence rather than form the basis for decision-making in themselves" [28].

In the Bayesian paradigm we posit that rather than being thought of as a fixed constant, the unknown parameter, λ, can be more accurately described as a random variable with its own probability distribution, $Q(\lambda)$. Otherwise known as the prior distribution, $Q(\lambda)$ "summarizes any information we have about λ not related to that provided by the data" [5].

We condition on λ to obtain the likelihood function of the data, $f(x|\lambda)$. The joint distribution of λ and the data is obtained by multiplying the likelihood function by the density of the prior, $\pi(\lambda)$. Finally, we obtain the posterior distribution of λ, given the data, the distribution representative of our updated belief about λ, via Bayes' theorem:

$$\pi(\lambda|x) = \frac{f(x|\lambda)\pi(\lambda)}{f(x)}, \tag{1.7}$$

where $f(x)$ is the density of the marginal distribution of the data,

$$f(x) = \int f(x|\lambda)\, dQ(\lambda),$$

and

$$\int dQ(\lambda) \equiv \int \pi(\lambda)\, d\lambda.$$

Dividing by $f(x)$, the marginal probability density function of the data, ensures that $\pi(\lambda|x)$ integrates to 1. The resulting posterior density is used to find an estimator for λ, depending on a given loss function [21]. The mean squared error loss function leads to the mean of the posterior distribution. Our next steps are aptly described in a paragraph from Maritz and Lwin (1970, p. 2):

> Generally, and more formally, we shall be concerned with problems arising in the following manner: an observation x is made on a random variable X whose distribution depends on the parameter

λ. Our task is to make a decision $\delta(x)$ about the value of λ. Typically the decision may be the calculation of a point estimate of λ, or it may be a choice between two hypothetical values of λ. The dependence of the decision on x is indicated by using the symbol $\delta_Q(x)$, which is said to represent the decision function.

The optimal decision function is thus the one that minimizes our expected loss, or the average risk, with respect to the prior, $Q(\lambda)$ [20]. Also known as Bayes risk, it has the form

$$W(\delta) = \iint \ell(\delta(x), \lambda) f(x|\lambda)\, dx\, dQ(\lambda). \tag{1.8}$$

Minimizing Bayes risk will give us the desired $\delta(x)$, which is denoted as $\delta_Q(x)$ because of its dependence on the prior distribution [20]. For the squared-error loss, we obtain

$$\delta_Q(x) = E(\lambda|x) = \frac{\int \lambda f(x|\lambda)\, dQ(\lambda)}{\int f(x|\lambda)\, dQ(\lambda)}.$$

The optimal estimate for λ is thus shown to be the mean of the posterior distribution [21].

The following theorem, from Lehmann and Casella (1998, p. 228), states that given a selected loss function, Bayes rule will exist.

Theorem 1.4.1 Suppose the following assumptions hold for the problem of estimating $g(\Lambda)$ with non-negative loss function $\ell(\delta(x), \lambda)$.

(a) There exists an estimator δ_0 with finite risk.

(b) For almost all x, there exists a value $\delta_Q(x)$ minimizing

$$E\{\ell[\Lambda, \delta(x)]|X = x\}$$

.

Then $\delta_Q(X)$ is a Bayes estimator.

Proof. See Lehmann and Casella (1998, p. 228).

1.4.2 Empirical Bayes Methodology

The success of Bayesian methods depends heavily on the availability of a prior that conforms to the data. These methods would be very useful if we could combine our objective observations on scoring rates with hockey experts opinions quantified in the form of priors. In practice, however, a valid prior is not that easy to find. Therefore, some criticism of Bayesian procedures include "their inability to deal with all but the most basic examples, [their] overreliance on computationally convenient priors, and [fragility] in their dependence on a specific prior" [5]. For these reasons, we shift our attention to empirical Bayesian methods.

Classical Bayesian approaches require eliciting a prior distribution before viewing the data [5]. In cases where this is not feasible, statisticians such as Herbert Robbins chose to use the data to estimate the prior (via method of moments or maximum likelihood) instead [5]. The various statistical strategies resulting from this have come to be referred to as empirical Bayes methods.

As stated by Efron (2014), "the essential empirical Bayes task [is] learning an appropriate prior distribution from ongoing statistical experience, rather than knowing it by assumption". Interestingly, by utilizing the data in prior estimation, we are returning to a methodology that is inherently non-Bayesian [4]. However, since the data is used again to compute the posterior, the Bayesian philosophy is still critical to our predictions [4].

So far, we have only considered one parameter, λ, with prior distribution, $Q(\lambda)$, and density function, $\pi(\lambda)$. In actuality, the prior may also be dependent on some (hyper)parameter, η, so our density function for λ can be rewritten as $\pi(\lambda|\eta)$ [5]. When η is known, Bayes' Theorem (1.7) receives the minor revision:

$$\pi(\lambda|x,\eta) = \frac{f(x|\lambda)\pi(\lambda|\eta)}{\int f(x|\lambda)\,\pi(\lambda|\eta)d\lambda}. \tag{1.9}$$

A hierarchical model can be formed based on the intuition that each successive hyperprior also depends on some parameters. This hierarchy theoretically extends indefinitely until we reach a stage where the final prior's parameters are assumed known. To avoid having to "make this assumption, the empirical Bayes (EB) approach uses the observed data to estimate these final stage parameters (or to directly estimate the Bayes decision rule) and then proceeds as though the prior were known" [5]. To simplify computation, we may wish to limit our model to two stages. To do this, we can replace η in (1.9) with it's estimate, $\hat{\eta}$, effectively making the hyperprior our final stage. This estimate is obtained by finding the value which maximizes the marginal distribution, $f(x)$. Replacing η with $\hat{\eta}$ is beneficial because it allows us to pull the estimated hyperprior $h(\hat{\eta})$ outside of the integral, thus increasing ease in computation. We can now write (1.9) as follows

$$\pi(\lambda|x,\eta) = \frac{f(x|\lambda)\pi(\lambda|\hat{\eta})}{\int f(x|\lambda)\,\pi(\lambda|\hat{\eta})d\lambda}.$$

Further distinction comes when comparing parametric and nonparametric empirical Bayes methods. Parametric methods refer to those used when we are able to select a prior, $\pi(\lambda|\eta)$, that fits a well known probability distribution. Assuming a parametric distribution, the only requirement for estimating the posterior is to replace η with $\hat{\eta}$. Now, given a set of coordinates, $\lambda_1, ..., \lambda_n$, inferences about λ_i can be made by pulling information from the remaining $\lambda_1, ..., \lambda_i - 1, \lambda_i + 1, ..., \lambda_n$. Estimates obtained in this way typically "outperform the MLE [and] even [do so] when [Q] is misspecified, or even when the λs are not sampled from a prior" [5]. Unfortunately, experiments usually do not follow a common parametric distribution, so nonparametric methods must be explored.

Nonparametric empirical Bayesian methods were introduced as a way to approximate the unknown prior $Q(\lambda)$ when its form did not follow any standard probability distribution [29]. In these cases, we would like to estimate Q using the observed data, $x_1, ..., x_n$. There are several ways to do it, but

the following will be particularly useful when we reach Section 1.4.4 on the Kiefer-Wolfowitz method.

Let X be a random variable having probability density function or probability mass function depending on a parameter θ,

$$L(\theta) = f(x|\theta)$$

and the parameter θ be a random variable with a prior distribution function Q. The marginal distribution of X is then the mixture

$$L(Q) = f(x|Q) = \int_{\Omega} L(\theta)\, dQ(\theta), \tag{1.10}$$

where Ω is the parameter space.

To get an idea of the differences between empirical and classical Bayes, as well as how we deal with known versus unknown priors, we now look at some examples from Maritz and Lwin (1970) showing calculations of decision functions. The first two examples follow a classical Bayes approach, while the third example employs a nonparametric method called Robbins marginal maximum likelihood estimate.

Example 1.4.1 Let $p(x|\theta)$ be the Poisson probability distribution

$$p(x|\theta) = \frac{e^{-\theta}\theta^x}{x!}, \quad x = 0, 1, 2, ...$$

and $Q(\theta)$ a Gamma cumulative distribution function with probability density function

$$\pi(\theta) = \frac{1}{\Gamma(\alpha)}\beta^{\alpha}\theta^{\alpha-1}e^{-\beta\theta}, \quad \alpha, \beta > 0.$$

Then

$$\delta_Q(x) = \frac{\displaystyle\int \theta \frac{e^{-\theta}\theta^x}{x!}\frac{1}{\Gamma(\alpha)}\beta^\alpha\theta^{\alpha-1}e^{-\beta\theta}\,d\theta}{\displaystyle\int \frac{e^{-\theta}\theta^x}{x!}\frac{1}{\Gamma(\alpha)}\beta^\alpha\theta^{\alpha-1}e^{-\beta\theta}\,d\theta}.$$

Now, cancelling out everything constant with respect to θ and combining like terms we obtain

$$\delta_Q(x) = \frac{\displaystyle\int \theta^{\alpha+x}e^{-\theta(\beta+1)}\,d\theta}{\displaystyle\int \theta^{\alpha+x-1}e^{-\theta(\beta+1)}\,d\theta}.$$

We can see that the numerator and denominator both closely resemble Gamma distributions. Denoting the denominator as $p_Q(x)$ and the numerator as $p_Q(x+1)$, and recalling that we have prior density $\pi(\theta) \sim \Gamma(\alpha, \beta)$ gives

$$p_Q(x) \sim \Gamma(\alpha + x, \beta + 1)$$

and

$$p_Q(x+1) \sim \Gamma(\alpha + x + 1, \beta + 1),$$

where $p_Q(x)$ represents the marginal density of X, and $p_Q(x+1)$ represents the marginal density of $X + 1$.

The convenience of this is that with the inclusion of the previously removed constant terms the numerator and denominator will both integrate to 1, as shown below

$$\delta_Q(x) = \frac{\dfrac{\Gamma(\alpha+x+1)}{(\beta+1)^{\alpha+x+1}}\displaystyle\int \frac{1}{\Gamma(\alpha+x+1)}(\beta+1)^{\alpha+x+1}\theta^{\alpha+x}e^{-\theta(\beta+1)}\,d\theta}{\dfrac{\Gamma(\alpha+x)}{(\beta+1)^{\alpha+x}}\displaystyle\int \frac{1}{\Gamma(\alpha+x)}(\beta+1)^{\alpha+x}\theta^{\alpha+x-1}e^{-\theta(\beta+1)}\,d\theta}.$$

Recalling that $\Gamma(x) = (x-1)!$, it can be seen that

$$\delta_Q(x) = \frac{(\alpha+x)!(\beta+1)^{\alpha+x}}{(\alpha+x-1)!(\beta+1)^{\alpha+x+1}}.$$

Simplifying further, we get the final result

$$\delta_Q(x) = \frac{\alpha+x}{\beta+1}.$$

In example 1.4.1 we were given a prior distribution with known cumulative distribution and density functions. As was noted earlier, it is uncommon in practice to have a known prior at our disposal, and "even when $[Q]$ may be assumed to exist it is generally unknown to the experimenter" [27]. This leads us to our next example, where Q is not given:

Example 1.4.2 Let $p(x|\theta)$ be the Poisson probability distribution

$$p(x|\theta) = \frac{e^{-\theta}\theta^x}{x!}, \quad x = 0, 1, 2, ...$$

$$\delta_Q(x) = \frac{\frac{1}{x!}\int \theta^{x+1}e^{-\theta}\, dQ(\theta)}{\frac{1}{x!}\int \theta^{x}e^{-\theta}\, dQ(\theta)}$$

$$= \frac{(x+1)\int \frac{\theta^{x+1}e^{-\theta}}{(x+1)!}\, dQ(\theta)}{\int \frac{\theta^{x}e^{-\theta}}{x!}\, dQ(\theta)}$$

$$= (x+1)\frac{p_Q(x+1)}{p_Q(x)},$$

where $p_Q(x)$ is a mixed Poisson probability distribution.

Without an explicit prior, estimation will prove quite difficult. If we instead use a nonparametric empirical Bayes method, we gain the ability to estimate $p_Q(x)$.

An initial attempt at a nonparametric empirical Bayes method was made by Robbins (1955). He postulated that given a random variable y, one could compute a "completely nonparametric estimate ... by estimating the marginal probabilities [of y] by their empirical frequencies, [y_i], namely" [5]

$$\delta_n(y_1, ..., y_n; y) = (y_i + 1)\frac{\#(ys \text{ equal to } y_i + 1)}{\#(ys \text{ equal to } y_i).}$$

This method is utilized in the following example:

Example 1.4.3 As in the last two examples, let $p(x|\theta)$ be the Poisson probability distribution

$$p(x|\theta) = \frac{e^{-\theta}\theta^x}{x!}, \quad x = 0, 1, 2, ...$$

Suppose that among the past observations there are $f_n(x)$ having the value x, $x = 1, 2, ...$ Since $x_1, x_2, ..., x_n$ are independent realizations of X_Q, with probability distribution $p_Q(x)$, we can estimate $p_Q(x)$ by $\frac{f_n(x)}{n}$. Including the current x we have $[1 + f_n(x)]$ observations with the value x, out of a total of $n + 1$ observations, and $f_n(x + 1)$ with value $x + 1$. Therefore we have an estimate of the Bayes estimate given by

$$\delta_n(x_1, ..., x_n; x) = (x + 1)\frac{f_n(x + 1)}{[1 + f_n(x)]}$$

with $f_0(x), f_1(x), ...$ representing our observations.

While Robbins marginal maximum likelihood method represents an improvement in estimation techniques, it might not perform that well in practice. Particularly, in the case of predicting scoring rates in hockey, Robbins method does not seem to have an adequate way of dealing with the time epoch variable, g (which may be different for different players). We could not see how this method could be adapted to different time epochs, so it will not be used in our analysis. Nonetheless, Robbins method represents an important first step in the realm of nonparametric empirical Bayes estimation.

The next two subsections focus on two empirical Bayes methods that work particularly well when dealing with hockey statistics. Subsection 1.4.3 covers a parametric approach, the Poisson-Gamma model, while Subsection 1.4.4 looks at the nonparametric Kiefer-Wolfowitz method.

1.4.3 Poisson-Gamma Parametric Model

The Poisson-Gamma model arises when the Gamma distribution is assigned as the prior for λ. This is referred to as a conjugate prior because the posterior will be a Gamma distribution as well [5]. Conjugate priors are computationally convenient because they "reduce Bayesian updating to modifying the parameters of the prior distribution ... rather than computing integrals" [23]. The following is the derivation of the Poisson-Gamma model, and the ensuing predictions are made for Poisson outcomes that are observed for different players over different time epochs.

Suppose $X|\lambda \sim \text{Poisson}(\lambda g)$, $\lambda \sim \Gamma(\alpha, \beta)$, where the unknown parameters $\alpha, \beta > 0$ are the shape and rate parameters, respectively [5].

Let $\Gamma(x)$ represent a Gamma function where

$$\Gamma(x) = \int_0^\infty z^{x-1} e^{-z}\, dz, \quad x > 0, \tag{1.11}$$

and

$$P[\Lambda = \lambda] = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}. \tag{1.12}$$

To find the marginal distribution of X we first integrate the product of (1.12) and (1.3) to get

$$P[X = x] = P[X|\Lambda = \lambda]P[\Lambda = \lambda] = \int_0^\infty \frac{(\lambda g)^x e^{-\lambda g}}{x!} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}\, d\lambda.$$

Combining like terms gives

$$P[X=x] = \frac{g^x \beta^\alpha}{x!\Gamma(\alpha)} \int_0^\infty \lambda^{x+\alpha+1} e^{-(g+\beta)\lambda}\, d\lambda$$

This integral resembles (1.11) with $z = (g+\beta)\lambda$, so we can rewrite $P[X=x]$ as follows

$$P[X=x] = \frac{g^x \beta^\alpha}{\Gamma(\alpha)\Gamma(x+1)} \int_0^\infty \left(\frac{z}{g+\beta}\right)^{x+\alpha-1} e^{-z} \frac{dz}{g+\beta}.$$

By using the same trick as in example 1.4.1, this becomes

$$P[X=x] = \left(\frac{\beta}{g+\beta}\right)^\alpha \left(\frac{g}{g+\beta}\right)^x \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)\Gamma(x+1)} \int_0^\infty \frac{1}{\Gamma(x+\alpha)} z^{x+\alpha-1} e^{-z}\, dz.$$

$$= \left(\frac{\beta}{g+\beta}\right)^\alpha \left(\frac{g}{g+\beta}\right)^x \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)\Gamma(x+1)},$$

which is a negative binomial distribution with parameters α and $\beta/(g+\beta)$. Now, we minimize the marginal distribution of X in an effort to obtain the best possible estimate of λ. Possible methods to do this are:

(1) Maximum Likelihood Estimation

Just as with previous MLE problems, we look at minimizing the likelihood function, and manipulating to obtain estimates for $\hat{\alpha}$ and $\hat{\beta}$.

$$L(\lambda_j; X_j) = \prod_j \left(\frac{\beta}{g_j+\beta}\right)^\alpha \left(\frac{g_j}{g_j+\beta}\right)^{x_j} \frac{\Gamma(x_j+\alpha)}{\Gamma(\alpha)\Gamma(x_j+1)}.$$

$$\ell(\lambda_j; X_j) = -\sum_j \big[\alpha \log(\beta) - \alpha \log(g_j+\beta) + x_j \log(g_j) - x_j \log(g_j+\beta) + \log(\Gamma(x_j+\alpha)) - \log(\Gamma(\alpha)) - \log(\Gamma(x_j+1))\big]$$

From here, solving by hand becomes fairly difficult. Therefore we utilize the non-linear minimization (`nlm()`) function in R, which uses a Newton-type algorithm to minimize the desired function. The `nlm()`

function has arguments f and p, where f represents "the function to be minimized, returning a single numeric value ... [and p represents the] starting parameter values for the minimization" [25]. The result is the Poisson-Gamma maximum likelihood estimator, $\tilde{\lambda}_j$.

(2) Method of Moments

Recall some basic properties for method of moments:

$$E(X|\lambda) = g\lambda$$

$$V(X|\lambda) = g\lambda$$

$$E(\lambda) = \frac{\alpha}{\beta}$$

$$V(\lambda) = \frac{\alpha}{\beta^2}$$

1^{st} moment:

$$E(X) = E(E(X|\lambda)) = E(g\lambda) = gE(\lambda) = \frac{g\alpha}{\beta}.$$

2^{nd} moment:

$$E\left[\frac{X(X-1)}{g^2}\right] = \frac{1}{g^2}[E(X^2) - E(X)].$$

To solve this, we first need to solve for $E(X^2)$:

$$E(X^2) = E(E(X^2|\lambda)) = E[g\lambda + (g\lambda)^2] = gE(\lambda) + g^2E(\lambda^2)$$

$$= g\frac{\alpha}{\beta} + g^2\frac{\alpha}{\beta^2} + g^2\frac{\alpha^2}{\beta^2}.$$

Therefore,

$$E\left[\frac{X(X-1)}{g^2}\right] = \frac{1}{g^2}\left[g\frac{\alpha}{\beta} + g^2\frac{\alpha}{\beta^2} + g^2\frac{\alpha^2}{\beta^2} - g\frac{\alpha}{\beta}\right]$$

$$= \frac{1}{g^2}\left[g^2\Big(\frac{\alpha}{\beta^2} + \frac{\alpha^2}{\beta^2}\Big)\right]$$

$$= \frac{\alpha}{\beta^2}(\alpha+1).$$

Replacing $E\Big[\frac{X(X-1)}{g^2}\Big]$ with its empirical moment we get

$$\frac{\alpha}{\beta^2}(\alpha+1) = \frac{1}{n}\sum_j^n \frac{x_j(x_j-1)}{g_j}$$

Then to solve for β we manipulate the following:

$$\frac{E\left[\dfrac{X_j(X_j-1)}{g_j^2}\right]}{E\left[\dfrac{X_j}{g_j}\right]} - E\left[\frac{X_j}{g_j}\right] = \frac{\dfrac{\alpha(\alpha+1)}{\beta^2}}{\dfrac{\alpha}{\beta}} - \frac{\alpha}{\beta} = \frac{\alpha+1}{\beta} - \frac{\alpha}{\beta} = \frac{1}{\beta}$$

Using the empirical moment, we find the estimate for β is

$$\hat{\beta} = \begin{cases} \left[\dfrac{\dfrac{1}{n}\displaystyle\sum_j^n \frac{x_j(x_j-1)}{g_j^2}}{\dfrac{1}{n}\displaystyle\sum_j^n \frac{x_j}{g_j}} - \frac{1}{n}\sum_j^n \frac{x_j}{g_j}\right]^{-1}, & \frac{1}{\beta} > 0 \\ \infty, & \text{otherwise} \end{cases}$$

For the estimate of α, we set

$$\hat{\alpha} = \hat{\beta} E\left[\frac{X_j}{g_j}\right]. \tag{1.13}$$

If $\beta = \infty$, we replace it in 1.13 by some large value ($\beta = 5000$, say).

Implementing the same strategy as example 1.4.1, we calculate the posterior mean:

$$E[\Lambda|X = x] = \frac{x + \hat{\alpha}}{\hat{\beta} + g},$$

The Poisson-Gamma method of moments estimate for the j^{th} player is then

$$\tilde{\tilde{\lambda}}_j = \frac{x_j + \hat{\alpha}}{\hat{\beta} + g_j}.$$

1.4.4 The Kiefer-Wolfowitz Nonparametric Method

The method proposed by Kiefer and Wolfowitz (1956) attempts to use maximum likelihood estimation to estimate the prior distribution, Q, in a nonparametric way. Recalling (1.10), we write this as

$$\max_{Q \in \mathcal{P}} L(Q) = \max_{Q \in \mathcal{P}} \int_{\Omega} L(\lambda)\, dQ(\lambda),$$

where $\mathcal{P}$ is the class of all probability measures on Ω. Once again we would like to compute the posterior mean. Using the estimate for Q, it can be found by

$$\frac{\int_{\Omega} \lambda L(\lambda)\, d\hat{Q}(\lambda)}{\int_{\Omega} L(\lambda)\, d\hat{Q}(\lambda)}.$$

The following example is taken directly from Tao (2014):

Example 1.4.4 Let $X_i \sim \text{Poisson}(\theta_i)$, where $i = 1, ..., n$ and θ_i's are taken from a distribution function Q. The Kiefer-Wolfowitz MLE solves

$$\max_{Q\in\mathcal{P}}\left[\sum_{i=1}^{n}\ln(L_i(Q))\right]=\max_{Q\in\mathcal{P}}\left[\sum_{i=1}^{n}\ln\left(\int_\Omega\frac{e^{-\theta}\theta^{x_i}}{x_i!}\,dQ(\theta)\right)\right].$$

Suppose there are K distinct data points. The Kiefer-Wolfowitz estimator can be rewritten as

$$\max_{Q\in\mathcal{P}}\left\{\ln\left[\prod_{i=1}^{K}(L_i(Q))^{n_i}\right]\right\}=\max_{Q\in\mathcal{P}}\left[\sum_{i=1}^{K}n_i\ln(L_i(Q))\right]$$

which is equivalent to

$$\min_{L(Q)\in\mathcal{M}}\left[-\sum_{i=1}^{K}n_i\ln(L_i(Q))\right],$$

where $\mathcal{M}$ is a convex hull (see def. 2.1.5) representing the set of mixture density vectors $\mathcal{M}=\{L(Q)|Q\in\mathcal{P}\}$ and $L(Q)=\{L_1(Q),...,L_K(Q)\}$.

The Kiefer-Wolfowitz method has two formulations; primal and dual. In our analysis we exclusively work with the primal problem, but the dual problem has many advantages in practice, so both formulations are presented here.

Given an unspecified prior, the form of the infinite-dimensional Kiefer-Wolfowitz primal problem is

$$\min_{Q\in\mathcal{P}}-\sum_{i=1}^{n}\log\int_\Omega f(y_i|\theta)\,dQ(\theta). \tag{1.14}$$

In high-dimensional problems computing (1.14) can become exceedingly difficult. Therefore, we may wish to solve the dual problem instead, as it is conveniently finite-dimensional [30]. The following theorem, from Koenker and Mizera (2014), gives the dual formulation:

Theorem 1.4.2 The solution, $\hat{Q}$, of (1.14) exists, and is an atomic probability measure with no more than n atoms. The locations, $\hat{\theta}_j$, and the masses, $\hat{f}_j$, at these locations can be found via the following dual characterization: the solution, $\hat{v}$ of

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \log v_i \\ \text{subject to} \quad & \sum_{i=1}^{n} v_i f(y_i|\theta) \leqslant n, \quad \forall \theta \end{aligned} \tag{1.15}$$

satisfies the extremal equations (n equations in less than n variables)

$$\sum_j f(y_i|\hat{\theta}_j)\hat{f}_j = \frac{1}{\hat{v}_i},$$

and $\hat{\theta}_j$ are exactly those θ where the dual constraint is active – that is, the constraint function in (1.15) is equal to n.

The Kiefer-Wolfowitz estimate for scoring rates in hockey is given by

$$\tilde{\tilde{\lambda}}_j = \frac{\sum_j \lambda_j g_j f(x_j|\lambda_j g_j)\hat{\pi}_i}{\sum_j f(x_j|\lambda_j g_j)\hat{\pi}_i}.$$

Chapter 2

Implementation and Results

The main goal of this chapter is to implement empirical Bayes methods in the prediction of hockey players' scoring rates. This basically amounts to implementing the Kiefer-Wolfowitz method, as other methods discussed above are rather straightforward computationally.

In the past, statisticians such as Jiang and Zhang (2009) and Laird (1978) have used the EM-algorithm to obtain the Kiefer-Wolfowitz MLE. Koenker and Mizera (2014) found that better predictive performance could be achieved if a convex optimization approach was taken instead. Thus, Section 2.1 will introduce some important concepts relating to convex optimization. Afterwards, we apply what we have learned to hockey data in Section 2.2.

2.1 Convex Optimization

The following paragraph, from Koenker and Mizera (2014), list some of the benefits of using convex optimization in estimation:

> In contrast to prior methods for these problems, our new approaches are cast as convex optimization problems that can be efficiently solved by modern interior point methods. In particular, we show that the reformulation of the Kiefer–Wolfowitz estimator as a convex optimization problem reduces the computational effort by several orders of magnitude for typical problems, by comparison to prior EM-algorithm based methods, and thus greatly expands the practical applicability of the resulting methods.

This section begins with the basic ideas of optima and convexity in Sections 2.1.1 and 2.1.2, before introducing linear optimization in Section 2.1.3.1. This serves as a precursor to Section 2.1.3.2, which covers the type of convex optimization of particular interest to us: conic optimization.

2.1.1 Local and Global Optimization

A general optimization problem involves functions that contain both local and global minima. A local minimum refers to any point on a function or set that is less than its neighbouring points. The global minimum is the absolute minimum value of said function or set. Lili Mou's *Introduction to Machine Learning* course [22] formally defined local and global optima as follows:

Definition 2.1.1 $x^* \in X$ is a global optimum if

$$\forall y \in \mathrm{dom} f,\ f(y) \geqslant f(x^*)$$

Definition 2.1.2 $x^* \in X$ is a local optimum if

$$\exists\ \epsilon > 0\ \ \forall y \in \mathrm{dom} f$$

such that if

$$|y - x^*| < \epsilon$$

then

$$f(y) \geqslant f(x^*)$$

for

$$y \in \mathcal{N}_\epsilon(x^*)$$

where $\mathcal{N}_\in(x^*)$ are the set of points neighbouring x^*.

Shown in figure 2.1 is a visualization of the global and local optima of a function:

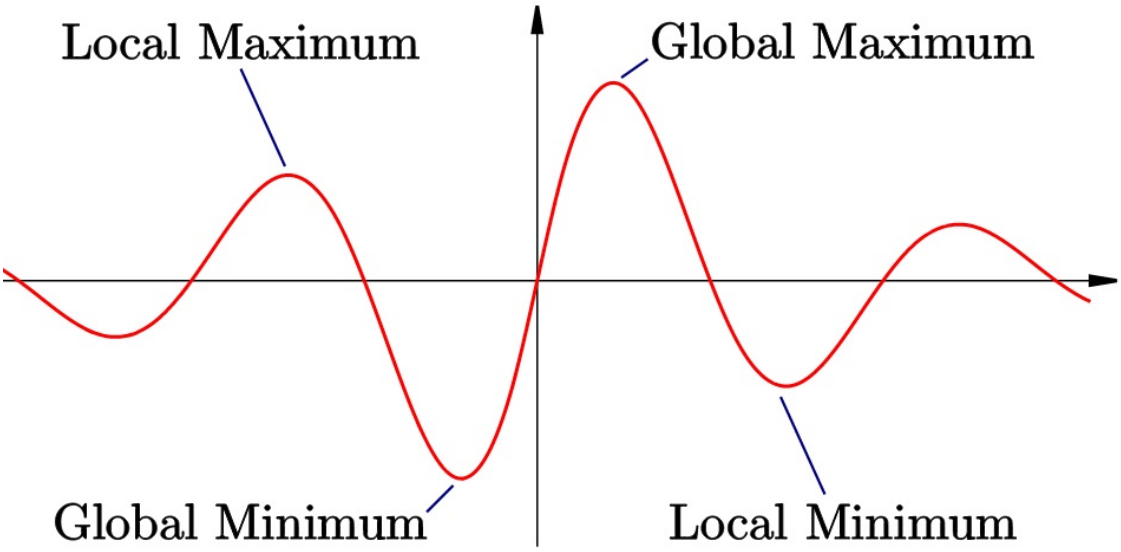

Figure 2.1: Local and global minima/maxima of a given function [10].

In theory, when optimizing, we would always like to find the global minimum of a function. However, obtaining the global minimum is not always possible, and even when it is, it can take an unreasonable amount of time to solve for [3]. Therefore, it is common to focus on local optimization instead, especially when faced with non-linear constraints. While we are no longer guaranteed to reach the true optimal point, these methods are desirable because they "can be fast, can handle large-scale problems, and are widely applicable, since they only require differentiability of the objective and constraint functions" [3]. Unfortunately, local optimization methods typically are not refined enough to meet our standards, so we aspire for something better. This leads us to Subsection 2.1.2.

2.1.2 Convexity

By nature, a convex function can have no more than one minimum, meaning any local minimum is guaranteed to be the global minimum as well [3]. Because of the resulting ease in computation this creates, convex functions are

highly desirable in optimization problems. Thus, it will be helpful to familiarize ourselves with some basic terminology relating to convexity. Boyd and Vandenberghe (2004) gave the following definitions of convex functions and sets:

Definition 2.1.3 (Convex Set). A set $\mathcal{S} \subseteq \mathbb{R}^d$ is convex if and only if

$$\forall x, y \in \mathcal{S}, \quad \forall \theta \in (0,1)$$

$$\theta x + (1-\theta)y \in \mathcal{S}$$

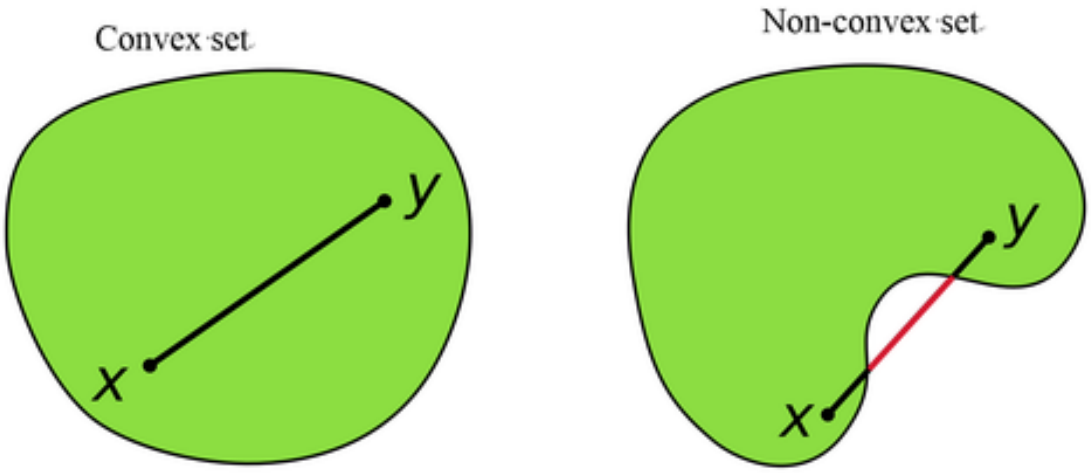

Figure 2.2: Convex vs. non-convex sets [11].

Definition 2.1.4 (Convex Function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if

(i) $\mathrm{dom} f$ is a convex set

(ii) (Jensen's inequality).

$$\forall x, y \in \mathrm{dom} f, \quad \forall \theta \in (0,1)$$

$$f(\theta x + (1-\theta)y) \leqslant \theta f(x) + (1-\theta)f(y).$$

Assuming that f is a twice differentiable function, the following two conditions also arise

(iii) (First-order condition).

$$\forall x, y \in \mathrm{dom} f$$

$$f(y) \geqslant f(x) + [\nabla f(x)]^T (y - x).$$

(iv) (Second-order condition).

$$\forall x, y \in \mathrm{dom} f$$

$$\nabla^2 f(x) \succeq 0.$$

In other words, the Hessian of f must be positive semidefinite.

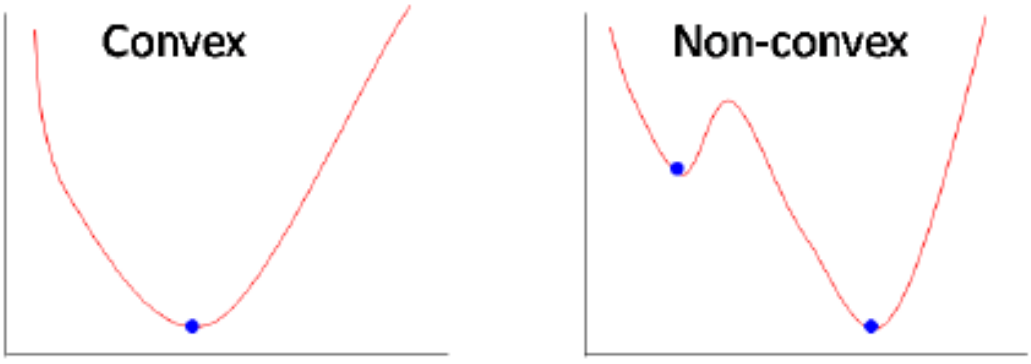

Figure 2.3: Convex vs. non-convex functions [12].

The definition of a convex hull (a topic briefly mentioned in Section 1.4.4) is provided by Tao (2018):

Definition 2.1.5 (Convex hull). The convex hull of a set C, denoted $\text{conv}(C)$, is the set of all convex combinations of points $x_1, ..., x_n$ in C:

$$\text{conv}(C) = \left\{ \sum_{i=1}^{n} \alpha_i x_i : \ x_i \in C, \ \alpha_i \geqslant 0, \ \sum_{i=1}^{n} \alpha_i = 1, \quad i = 1, ..., n \right\}.$$

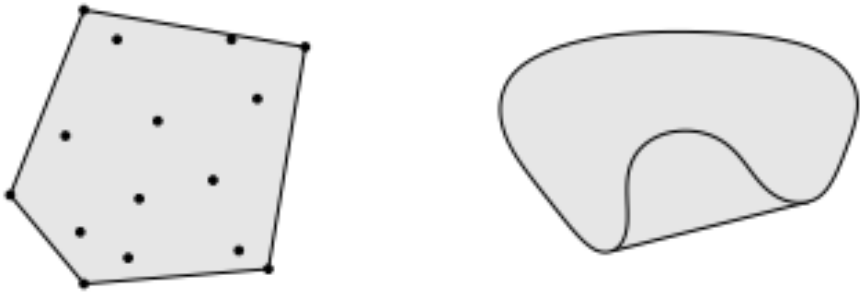

Figure 2.4: *Left.* A set of points enclosed by a pentagonal convex hull. *Right.* A nonconvex kidney shaped set enclosed by a convex hull (both sets are in $\mathbb{R}^2$) [3].

The examples below, from the homework assignments of Lili Mou's *Introduction to Machine Learning* course explain how to show whether a function or set is convex.

Example 2.1.1 Show that $\mathcal{S} = \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| + |x_2| \leqslant 1\}$ is a convex set.

Solution: Let $x_1, x_2, y_1, y_2 \in \mathcal{S}$. Then

$$|\theta x_1 + (1-\theta)y_1| + |\theta x_2 + (1-\theta)y_2| \leqslant |\theta x_1| + |(1-\theta)y_1| + |\theta x_2| + |(1-\theta)y_2|$$

$$= \theta|x_1| + (1-\theta)|y_1| + \theta|x_2| + (1-\theta)|y_2|$$

$$= \theta(|x_1| + |x_2|) + (1-\theta)(|y_1| + |y_2|) \leqslant \theta + (1-\theta) \leqslant 1$$

Therefore, $\mathcal{S}$ is a convex set.

Example 2.1.2 Prove $f : \mathbb{R}^2 \to \mathbb{R},\ f(x) = |x_1| + x_2|$ is a convex function

Solution: $f(\theta x + (1-\theta)y) = |\theta x_1 + (1-\theta)y_1| + |\theta x_2 + (1-\theta)y_2|$

$$\leqslant \theta(|x_1| + |x_2|) + (1-\theta)(|y_1| + |y_2|)$$

$$= \theta f(x) + (1-\theta)f(y)$$

Therefore, $f(x)$ is a convex function.

2.1.3 Optimization Techniques

Convex optimization problems take the general form

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0, \quad i = 1, ..., m, \\ & h_i(x) = 0, \quad i = 1, ..., p \end{aligned} \tag{2.1}$$

where $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $f_i : \mathbb{R}^n \to \mathbb{R}$ are the inequality constraint functions, and $h_i : \mathbb{R}^n \to \mathbb{R}$ are the equality constraint functions [30]. For this to be a proper convex optimization problem functions $f_0, f_1, ..., f_m$ must be convex, and functions $h_1, ..., h_p$ must be affine. In Boyd and Vandenberghe (2004, p. 36) an affine function is defined as

Definition 2.1.6 (Affine function) $h : \mathbb{R}^n \to \mathbb{R}$ is an affine function if it is the sum of a linear function and a constant.

In the next two subsections we will focus on more specific types of convex optimization, with the form of (2.1) supplying the basis for them.

2.1.3.1 Linear Optimization

In all forms of optimization our goal is to minimize some objective function $f_0(x)$. Linear optimization is the simplest of all types because of the linearity of both objective function and constraints, thus making it a good starting point for our discussion. These problems (also known as the primal problem) take the form [1]

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b, \\ & x \geqslant 0. \end{aligned} \tag{2.2}$$

In certain cases the linear optimization problem (2.2) may be infeasible. To determine whether or not a problem is feasible we can look at the feasible set, which is defined as

$$\mathcal{F}_p = \{x \in \mathbb{R}^n | Ax = b, x \geqslant 0\}.$$

As long as $\mathcal{F}_p$ is not empty (at least one point x lies within $\mathcal{F}_p$), a problem is considered feasible. What follows is a simple example of a linear optimization problem given in the Mosek Modeling Cookbook (MMC) (2022):

Example 2.1.3

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 - x_3 \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 1, \\ & x_1, x_2, x_3 \geqslant 0. \end{aligned}$$

We can clearly see that the optimal solution occurs when

$$(x_1^*, x_2^*, x_3^*) = (0, 0, 1),$$

with optimal value

$$p^* = x_1^* + 2x_2^* - x_3^* = -1.$$

In general, the optimal objective value, p^*, is found by calculating

$$p^* = \inf_x \{c^T x | Ax = b, x \geqslant 0\}.$$

If p^* is finite (as is the case in example 2.1.3) we have an optimal solution, but this does not hold true when $p^* = \pm\infty$. In the event that $p^* = -\infty$, the solution is feasible, but its unbounded nature means no true minimum is ever obtained. On the other hand, when $p^* = \infty$, there are no feasible solutions.

The linear optimization problem (2.2) has an associated Lagrangian function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n_+ \to \mathbb{R}$ that is a weighted combination of both the equality and inequality constraints added to the objective function. It is written as

$$L(x, y, s) = c^T x + y^T(b - Ax) - s^T x$$

where the weights $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n_+$ are referred to as Lagrange multiplier variables [1]. Taking the minimum of $L(x, y, s)$ over x produces the dual function [1]

$$g(y, s) = \min_x L(x, y, s) = \min_x x^T(c - A^T y - s) + b^T y = \begin{cases} b^T y, & c - A^T y - s = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

The resulting value of $g(y, s)$ is the lower bound of p^* for all possible pairs (y, s) [1]. Because this is an optimization problem, we would like to find the best possible lower bound. This results in the dual problem [1]

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & c - A^T y = s, \\ & s \geqslant 0. \end{aligned} \tag{2.3}$$

Given these new calculations, we now refer to the optimal objective value as d^* [1]. The possible values of d^* are the same as was seen for p^* (finite or $\pm\infty$), but because we are now maximizing instead of minimizing their significance

has reversed. That is, $d^* = -\infty$ is now an infeasible solution, while $d^* = \infty$ is feasible but unbounded [1].

Interestingly, if the primal problem is feasible and bounded we achieve what is called strong duality [1]. Strong duality occurs when $d^* = p^*$, thus allowing us to verify that both the primal and dual solutions are optimal [1]. The explanation of why this works, given in a paragraph from the MMC (2022), stems from the notion of weak duality:

> Suppose x^* and (y^*, s^*) are feasible points for the primal and dual problems (2.2) and (2.3), respectively. Then we have
>
> $$b^T y^* = (Ax^*)^T y^* = (x^*)^T (A^T y^*) = (x^*)^T (c - s^*) = c^T x^* - (s^*)^T x^* \leqslant c^T x^*$$
>
> so the dual objective value is a lower bound on the objective value of the primal. In particular, any dual feasible point (y^*, s^*) gives a lower bound:
>
> $$b^T y^* \leqslant p^*.$$

From this, we obtain the lemma for weak duality:

Lemma 2.1.1 (Weak Duality). $d^* \leqslant p^*$.

With weak duality, we can only guarantee both the primal and dual objective functions are optimal if $b^T y^* = c^T x^*$, or $d^* = p^*$, hence confirming what was stated above [1].

2.1.3.2 Conic Optimization

Linear optimization methods are useful when our problems have simple linear constraints. However, once nonlinear constraints are introduced these methods become ineffective, forcing us to consider alternative strategies. The approach we will employ is a reformulation to conic form.

Cones and convex cones are defined in Boyd and Vandenberghe (2004, p. 25) as follows:

Definition 2.1.7 (Cones). A set K is called a cone if for every $x \in K$ and $\theta \geqslant 0$ we have

$$\theta x \in K.$$

Definition 2.1.8 (Convex Cones). A set K is called a convex cone if for any $x_1, x_2 \in K$ and $\theta_1, \theta_2 \geqslant 0$, we have

$$\theta_1 x_1 + \theta_2 x_2 \in K.$$

Conic optimization problems take the general form

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b, \\ & x \in K, \end{aligned}$$

where $K \subseteq \mathbb{R}^n$ is a convex cone [1]. Similarly to linear optimization ,we may encounter cases where a conic problem is infeasible. This occurs when the feasible set, $\mathcal{F}_p$, is empty. In conic optimization, the feasible set is defined as

$$\mathcal{F}_p = \{x \in \mathbb{R}^n | Ax = b\} \cap K,$$

where $\mathcal{F}_p$ is a section of K. Brief descriptions of quadratic cones and duality in conic optimization are given below.

Quadratic Cones Conic quadratic optimization is a good starting point in this discussion because it is quite similar to linear optimization. That is, we are still using linear functions and constraints to optimize, but choose to represent the variables in a quadratic conic form [1]. We define n-dimensional quadratic cone as

$$\mathcal{Q}^n = \left\{ x \in \mathbb{R}^n \middle| x_1 \geqslant \sqrt{x_2^2 + x_3^2 + ... + x_n^2} \right\}.$$

Some basic convex sets that can be reformulated as quadratic cones are absolute values and Euclidean norms. For example, in a linear setting, the convex set $|x| \leqslant t$ would simply be modeled as the double inequality

$$-t \leqslant x \leqslant t,$$

but its quadratic conic form is

$$(t, x) \in \mathcal{Q}^2$$

where $\mathcal{Q}^2$ represents a two-dimensional quadratic cone [1]. We can also see that the Euclidean norm of $x \in \mathbb{R}^n$

$$||x||_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$$

with inequality

$$||x||_2 \leqslant t$$

has the quadratic conic form

$$(t, x) \in \mathcal{Q}^{n+1}$$

where $\mathcal{Q}^{n+1}$ represents an $(n+1)$-dimensional quadratic cone [1].

Slightly more complex is the notion of a rotated quadratic cone. Such a cone, in n-dimensions, is defined as

$$\mathcal{Q}_r^n = \{x \in \mathbb{R}^n | 2x_1x_2 > x_3^2 + ... + x_n^2,\ x_1, x_2 \geqslant 0\}$$

Figure 2.5 gives a clear visualization of what this cone, as well as the regular quadratic cone, looks like in 3-dimensional form:

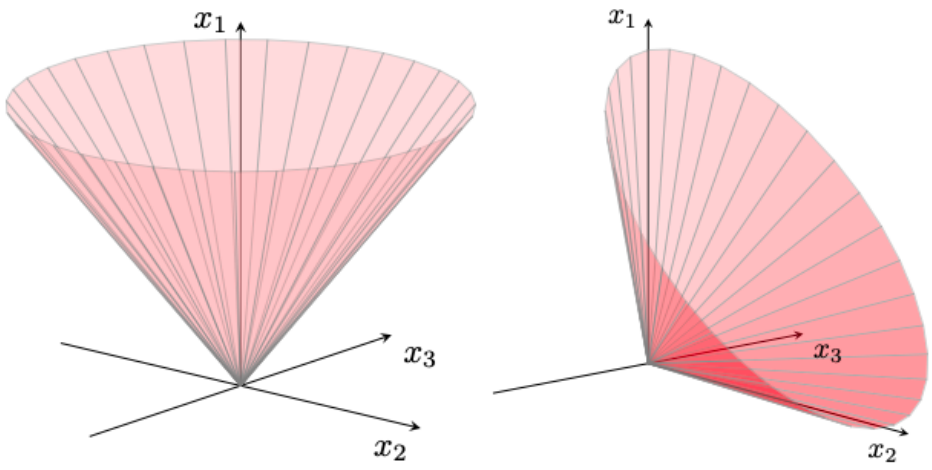

Figure 2.5: *Left.* Boundary of the quadratic cone, $x_1 \geqslant \sqrt{x_2^2 + x_3^2}$. *Right.* Boundary of the rotated quadratic cone, $2x_1x_2 \geqslant x_3^2,\ x_1, x_2 \geqslant 0$ [1].

When writing out representations of three-dimensional rotated quadratic cones, we use the basic form

$$2ab \geqslant c^2 \iff (a, b, c) \in \mathcal{Q}_r^3.$$

The following two examples, from the MMC (2022), show how we would represent some basic inequalities as rotated quadratic cones:

Example 2.1.4 Suppose we have the inequality $|t| \leqslant \sqrt{x}$. A quadratic formulation becomes clear through the manipulation

$$|t| \leqslant \sqrt{x} \Rightarrow x \geqslant t^2 \Rightarrow 2x\left(\frac{1}{2}\right) \geqslant t^2.$$

Therefore, the inequality can be represented quadratically as

$$\left(x, \frac{1}{2}, t\right) \in \mathcal{Q}_r^3.$$

Example 2.1.5 Suppose we have the inequality $t \geqslant (1/x)$. A quadratic formulation becomes clear through the manipulation

$$t \geqslant \frac{1}{x} \Rightarrow xt \geqslant 1 \Rightarrow 2xt \geqslant \left(\sqrt{2}\right)^2.$$

Therefore, the inequality can be represented quadratically as

$$\left(x, t, \sqrt{2}\right) \in \mathcal{Q}_r^3.$$

Duality in Conic Optimization We have discussed the merits of duality theory at various points throughout this thesis, so it is worth mentioning again in the context of conic optimization. As shown in the MMC (2022), if $K \subseteq \mathbb{R}^n$ is a closed convex cone we define the dual cone K^* as

$$K^* = \{y \in \mathbb{R}^n \ : \ y^T x \geqslant 0 \ \forall x \in K\}.$$

The conic dual problem has the form

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & c - A^T y = s, \\ & s \in K^*, \end{aligned}$$

where $K \subseteq \mathbb{R}^n$ is the dual cone [1]. Some important properties of the dual cone (given in Boyd and Vandenberghe (2004) and the MMC (2022)) include

(1) K^* is convex, regardless of whether or not K is.

(2) Every vector of K^* runs perpendicular to every vector of K.

(3) The dual of the dual returns the original cone – that is, $(K^*)^* = K$.

2.2 Application to Hockey Statistics

In this section, the subjects we have covered thus far will be applied to the world of hockey. Subsection 2.2.1 briefly explains how the data was obtained, Subsection 2.2.2 gives a description of some of the R functions used, and the results are analyzed in Subsection 2.2.3.

2.2.1 Data Collection

The data we will be working with was collected from QuantHockey's 2018-19 NHL scoring leaders webpage [24]. A filter for choosing custom start and end dates was applied to obtain end of week point totals for weeks two through eight. Eight separate datasets were formed in Microsoft Excel using players' cumulative point totals at the end of each of these weeks, as well as their full season totals. To avoid any sample size issues, players that played under 60 games over the course of the 2018-19 season were excluded. Any player that did not appear in all five datasets were filtered out entirely, leaving us with a total sample size of 350. The datasets were then made into comma-separated values (csv) text files, so we could do further work with them. The R script to do this is provided in Appendix A. The table below shows a portion of the dataset corresponding to the first four weeks of the 2018-19 season:

Note: Pos refers to the position of a player i.e. forward (F) or defence (D); G, A, and P, stand for goals, assists, and points, respectively; games played is denoted by GP; and TOI (time on ice) denotes the average number of minutes a players spends on the ice per game played.

Rank	Name	Team	Age	Pos	GP	G	A	P	TOI
1	Mikko Rantanen	COL	22	F	12	5	16	21	20.58
2	Patrice Bergeron	BOS	33	F	12	7	12	19	18.82
3	Evgeni Malkin	PIT	32	F	10	6	13	19	18.07
4	Patrick Kane	CHI	30	F	12	11	7	18	21.25
5	Connor McDavid	EDM	21	F	11	9	9	18	23.58
6	Nathan MacKinnon	COL	23	F	12	9	9	18	22.07
7	Sebastian Aho	CAR	21	F	12	4	13	17	19.63
8	David Pastrnak	BOS	22	F	12	11	5	16	18.68
9	Auston Matthews	TOR	21	F	11	10	6	16	17.35
10	Gabriel Landeskog	COL	26	F	12	10	6	16	20.48
...	...	...	...	...	...	...	...	...	...

Table 2.1: 2018-19 NHL season scoring leaders after four weeks.

2.2.2 Implementation: Description of R functions used

The function `fp.R` is a straightforward function used to compute the naïve scoring estimate, as well as the leaguewide estimate, via the method of moments or MLE, where we set $\lambda_j = \lambda$. This function also includes a shrinkage element that allows us to take one of the "one-fits-all" estimates and shrink the naïve ones towards them to a specified extent. The tables in Subsection 2.2.3 include an estimator called `js2`, which is found simply by shrinking the naïve estimate to an extent of 0.5, or halfway towards the "one-fits-all" estimate.

The `fpg.R` function is used to fit the Poisson-Gamma mixture. The function essentially just does what we already discussed in Subsection 1.4.3 – that is, it calculates the MLE via `nlm()`, and solves the method of moments using the values we found for $\hat{\alpha}$ and $\hat{\beta}$.

The `kwg.R` function evaluates both the primal and dual formulations of the Kiefer-Wolfowitz estimator. Because of the complexity of this method, we employ the help of Mosek, an optimization software capable of solving difficult problems through the use of its interior-point optimizer. It should be noted that Mosek version 9 or later must be worked with because earlier versions use a different technique for minimization.

The function `fprkw.R` feeds `kwg.R` an $n \times m$ matrix, E – where each row of E corresponds to a player – in order to obtain an estimate for the prior distribution in a Poisson setting. This function also returns the posterior mean predictions, with the help of `pstmea.R`, a simple function designed to calculate the posterior mean when required.

2.2.3 Results and Analysis

Tables 2.2–2.5 list the mean squared errors of the estimators discussed in Chapter 2, to show how they performed at predicting scoring rates. For readability, the mean squared errors are multiplied by 10^6 in Tables 2.2 and 2.3, and by 10^8 in Tables 2.4 and 2.5, and subsequently rounded. Estimators included in these tables are the naïve estimator, the "one-fits-all" method of moments (`1mm`) and maximum likelihood (`1ml`) estimators, the "poor man's shrinkage"

estimator (`js2`), both Poisson-Gamma estimators – method of moments (`pgm`) and maximum likelihood (`pgl`) – and the Kiefer-Wolfowitz estimator (`rkw`). The estimators in the last four columns are also the "poor man's shrinkage" estimator, the Poisson-Gamma estimators, and the Kiefer-Wolfowitz estimator, but applied separately to forwards and defencemen.

G	naïve	1mm	1ml	js2	pgm	pgl	rkw	*js2s*	*pgms*	*pgls*	*rkws*
2 weeks	58463	19623	19325	20303	15790	**15560**	17036	*20086*	*13351*	***13154***	*14520*
3 weeks	33315	18822	18696	13854	12595	**12556**	13033	*13190*	*11029*	***10919***	*11383*
4 weeks	26793	18686	18523	12246	11640	**11596**	11634	*11408*	*10088*	***10020***	*10101*
5 weeks	22998	18481	18373	11748	**11482**	11514	11543	*10971*	*10136*	*10083*	***10069***
6 weeks	19598	18430	18351	11143	**11043**	11071	11116	*10316*	*9891*	***9875***	*9933*
7 weeks	17905	18430	18350	**10648**	10694	10716	11048	*9924*	***9795***	***9796***	*10166*
8 weeks	17405	18324	18256	**10620**	10740	10638	11017	***9790***	*9972*	*9858*	*10124*

Table 2.2: 10^6(MSE) for all estimators by week (Goals) – full season minus first 8 weeks validation set. The best values for each week are in bold font. Time epochs are games played (GP).

Table 2.2 shows the mean squared error of predictions of goal rates, where the validation set for each week is the full season minus the first eight weeks, and the time epoch is games played. For the most part, the Poisson-Gamma (PG) and Kiefer-Wolfowitz (KW) methods produce the best estimates. The MLE variants tend to be a bit better than the MM variants, but the difference is negligible. While this does not hold true for all weeks, it seems as though the PG estimators actually fare slightly better than KW's. Shockingly, given its simplicity, the "poor man's shrinkage" estimator performs remarkably well here; even outperforming other top estimates in some of the later weeks. We also notice that the naïve estimator is basically useless in the early weeks, and is still not great in the later weeks. In fact, the PG and KW methods provide significantly better results after only two weeks than the naïve method does after eight. The performance of the "one-fits-all" methods are adequate, but fail to improve as the weeks go on. Looking at the last four columns, we can easily see that separating forwards and defencemen uniformly improved our predictions. Once again, the PG methods are somewhat superior to the KW method.

A	naïve	1mm	1ml	js2	pgm	pgl	rkw	*js2s*	*pgms*	*pgls*	*rkws*
2 weeks	80009	36556	36566	32735	28367	**28291**	28308	*32996*	*29503*	***28985***	*29673*
3 weeks	51146	35478	35247	24323	23381	23213	**22768**	*24380*	*24466*	*23939*	***23433***
4 weeks	42616	35052	34916	22307	21875	21947	**21630**	*22317*	***22256***	*22365*	*22467*
5 weeks	36016	34816	34859	21576	21263	21274	**20219**	*21560*	***21250***	*21354*	*21622*
6 weeks	29874	34736	34779	19935	19652	19658	**19108**	*19914*	***19637***	*19724*	*20048*
7 weeks	25211	34668	34755	18236	17717	17726	**17449**	*18213*	***17682***	*17748*	*18707*
8 weeks	23366	34549	34605	17575	16877	**16857**	17009	*17562*	***16809***	***16802***	*18120*

Table 2.3: 10^6(MSE) for all estimators by week (Assists) – full season minus first 8 weeks validation set. The best values for each week are in bold font. Time epochs are games played (GP).

Table 2.3 shows the mean squared error of predictions of assist rates, where the validation set and time epoch are the same as they were in Table 2.2. Right away, we notice that all of the mean squared errors in Table 2.3 are substantially higher than the mean squared errors in Table 2.2. Overall, it appears that the KW method actually does a better job at predicting assist rates than the PG model, albeit not by much. The "one-fits-all" prediction methods still fail to improve from week to week; so much so that the naïve estimator surpasses them in efficacy by week six. The "poor man's shrinkage" estimator does not perform better than PG and KW in this case, but interestingly, after separating forwards and defencemen, it only takes three weeks for its estimates to become better than the separated KW ones. Separating forwards and defencemen does not lead to improved estimates, and even seems to have a negative effect on the KW estimates.

One interesting development is that while the mean squared errors leveled off fairly quickly when predicting goal rates, for assist rates they continually improved as the weeks went on. A possible reason for this is that while there is a certain amount of randomness to all events in hockey, there is inherently more so in assists because of the inclusion of secondary assists, and because passers do not have any control over the shooting ability of their teammates. Therefore, it is easy to imagine that we may need a larger sample size before these values will begin to normalize. While purely speculatory, one may assume this is also the likely cause of the higher mean squared errors compared

to Table 2.2.

The similarity in performance of the Poisson-Gamma parametric model and the Kiefer-Wolfowitz nonparametric method are illustrated in Figures 2.6 and 2.7. Figure 2.6 shows the relationship between predictions of PG and KW when forwards and defencemen are combined, while Figure 2.7 shows the relationship between them when forwards and defencemen are separated.

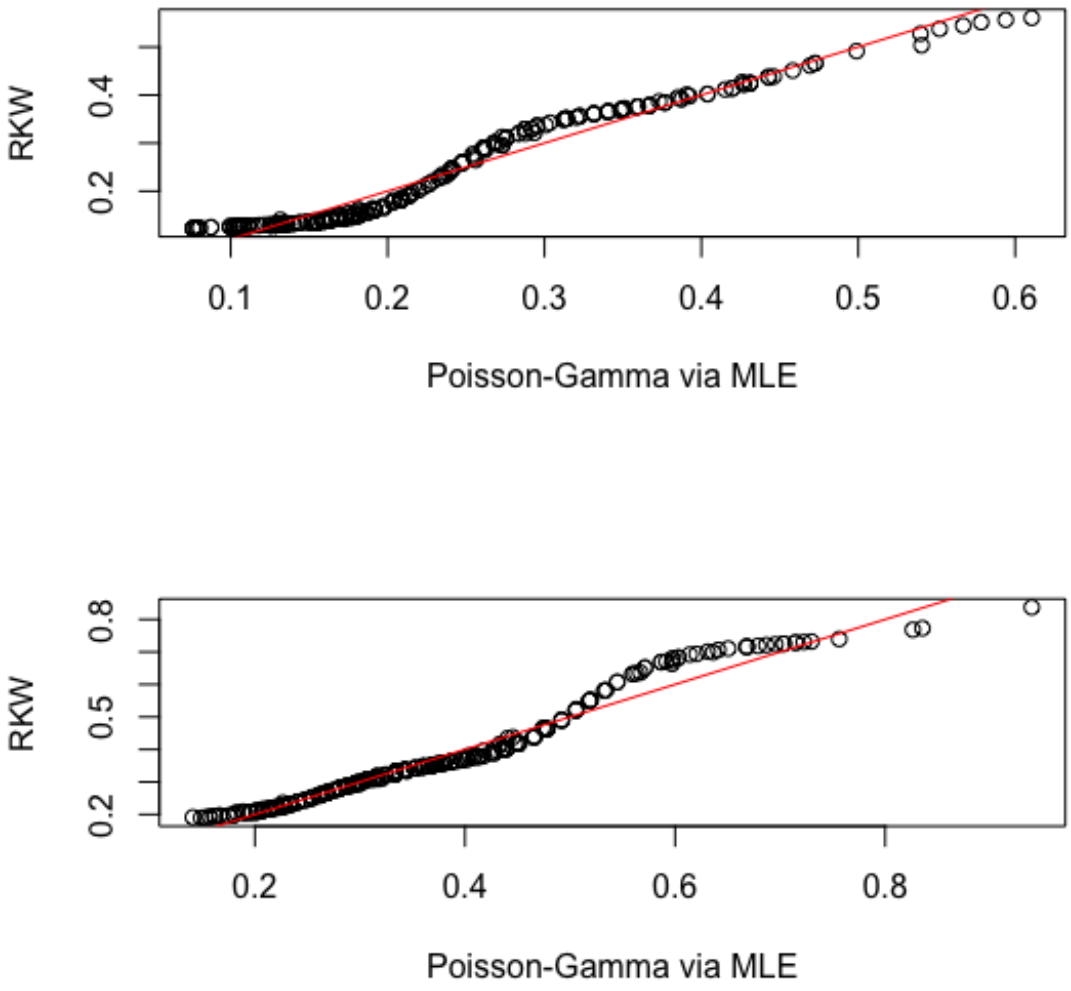

Figure 2.6: *Top.* Comparison of each player's mean squared error for KW vs PG via MLE (Goals). *Bottom.* Comparison of each player's mean squared error for KW vs PG via MLE (Assists).

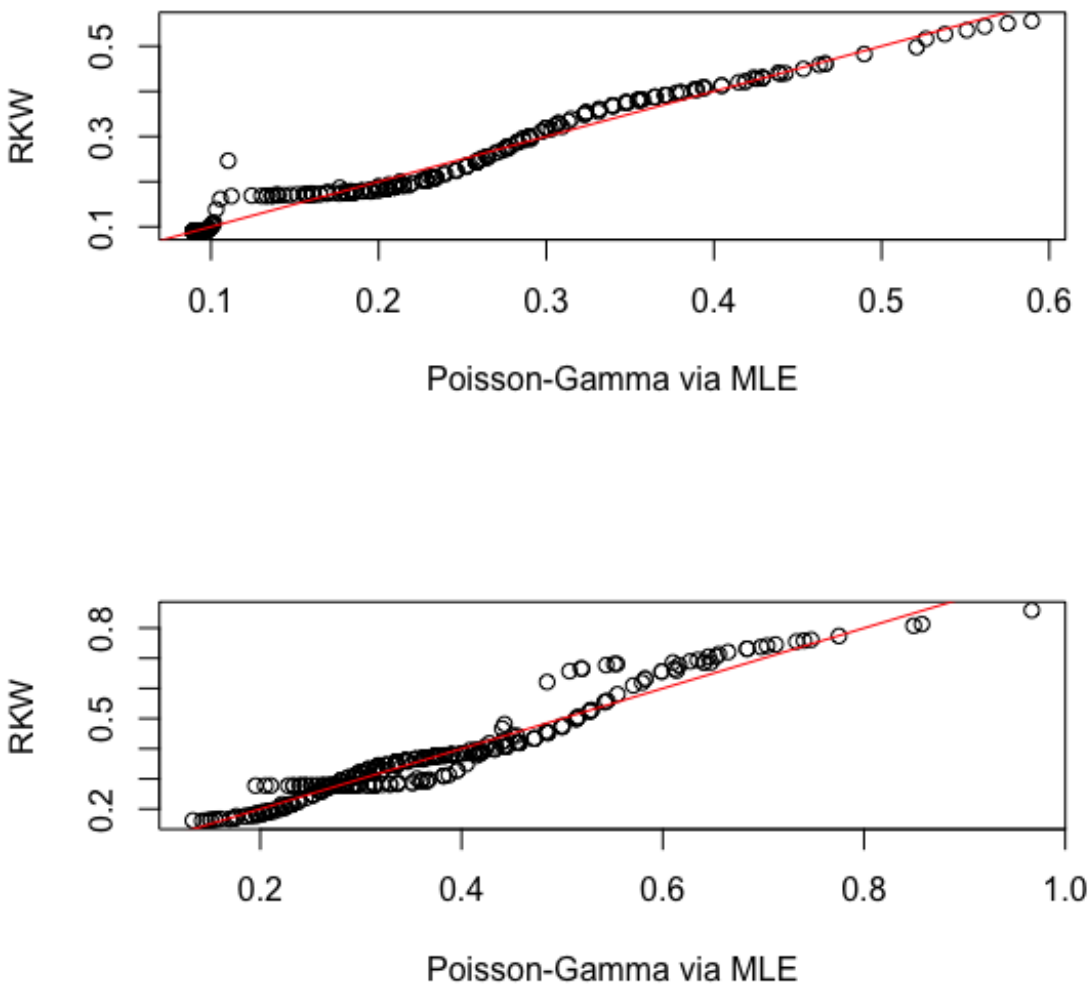

Figure 2.7: *Top.* Comparison of each player's mean squared error for KW vs PG via MLE with F and D separated (Goals).*Bottom.* Comparison of each player's mean squared error for KW vs PG via MLE with F and D separated (Assists).

Tables 2.4 and 2.5 mirror Tables 2.2 and 2.3, except the time epoch equal to games played has been changed to the total time on ice (TOI*GP). Looking at this is a worthwhile venture because it has the added nuance of accounting for the number of minutes a player plays in a game, which theoretically should lead to better predictions. The results line up to those seen in previous tables, with the PG and KW estimators continuing to lead the pack.

G	naïve	1mm	1ml	js2	pgm	pgl	rkw	*js2s*	*pgms*	*pgls*	*rkws*
2 weeks	20643	5836	5563	7019	**4716**	4866	5320	*7117*	***3661***	*3792*	*4389*
3 weeks	12059	5449	5345	4800	3872	3890	**3827**	*4647*	***3093***	*3216*	*3268*
4 weeks	9666	5373	5281	4235	3620	3668	**3594**	*3971*	***2854***	*2935*	*2987*
5 weeks	8347	5310	5225	3945	3545	3638	**3508**	*3678*	***2844***	*2876*	*2890*
6 weeks	7162	5300	5218	3720	**3449**	3546	3519	*3434*	***2817***	*2856*	*2879*
7 weeks	6662	5302	5213	3574	**3409**	3517	3445	*3321*	***2852***	*2863*	*2958*
8 weeks	6552	5266	5189	3554	3463	3496	**3378**	*3246*	*2924*	***2853***	*2960*

Table 2.4: 10^8(MSE) for all estimators by week (Goals) – full season minus first 8 weeks validation set. The best values for each week are in bold font. Time epoch are total time on ice.

A	naïve	1mm	1ml	js2	pgm	pgl	rkw	*js2s*	*pgms*	*pgls*	*rkws*
2 weeks	26274	8289	8550	9984	7266	7325	**7249**	*9841*	***6907***	*7232*	*7204*
3 weeks	16606	7832	79645	7215	6180	6130	**6039**	*7074*	***5949***	*8416*	*6513*
4 weeks	13890	7657	7791	6561	**5705**	5763	5780	*6431*	***5586***	*5695*	*5981*
5 weeks	11294	7560	7730	6100	5555	5607	**5423**	*5989*	***5402***	*5494*	*5638*
6 weeks	9305	7543	7691	5517	5245	5286	**5167**	*5474*	***5167***	*5272*	*5289*
7 weeks	7995	7500	7663	5024	**4815**	4864	4930	*4974*	***4760***	*4865*	*4913*
8 weeks	7343	7468	7604	4796	**4639**	4673	4681	*4755*	***4587***	*4666*	*4781*

Table 2.5: 10^8(MSE) for all estimators by week (Assists) – full season minus first 8 weeks validation set. The best values for each week are in bold font. Time epochs are total time on ice.

These predictions can also be done where the validation set differs for each dataset. In particular, we made predictions where the validation set for z weeks was the full season totals minus first z weeks totals. The results were not exactly equal, but akin to the tables shown above – the structure of winning methods were mostly the same as we have already shown, except in a few cases where the outcomes differed very little. Therefore, for the sake of brevity and clarity we chose not to present any tables with the updated validation set here.

As a bonus topic of interest, we look briefly at scoring rate predictions for players playing in their first NHL season; otherwise known as rookies. It would be easier to make predictions about a player's scoring rates with data on how they performed in past years, but when a player is a rookie we do not have access to such information. Therefore, we would like to know if empirical Bayes approaches like the Poisson-Gamma model and the Kiefer-Wolfowitz

method represent an improvement when estimating without knowledge of past scoring rates. Given in Table 2.6 is a comparison of the mean squared error in predictions for all players vs. predictions for rookies. We use all players – including rookies – to estimate the prior. A sample of 21 rookies was used to obtain these predictions.

G	naïve	1mm	1ml	js2	pgm	pgl	rkw	*js2s*	*pgms*	*pgls*	*rkws*
All Players	25170	17569	17418	10882	10348	10307	**10262**	*9940*	***8569***	*8633*	*8657*
Rookies	43413	10764	10318	10994	6685	6657	**5789**	*12158*	*7243*	*7366*	***6793***
A											
All Players	41181	33672	33549	20906	20482	20557	**20436**	*20918*	*20976*	***20867***	*20986*
Rookies	28335	24088	23617	13776	11783	11690	**11328**	*13734*	*10993*	*11002*	***10900***

Table 2.6: 10^6(MSE) of all players vs. rookies for all estimators using data from week 4 (Goals and assists). The best values are in bold font. Time epochs are games played (GP).

Overall, the predictions for rookies seems to fare better than those for all players. In fact, in some cases the predictions for rookies do almost twice as well. This could be because of the small sample we chose, but the results are interesting nonetheless. Further exploration of this is left to future work.

Conclusion

Future work might wish to consider goals and assists as dependent variables, as well as isolating for the effect of primary and secondary assists. Secondary assists often do not have as much to do with the goal being scored as primary ones do, and are therefore less repeatable. Because of this, secondary assists are likely to blame for some of the noise present in our results. Going forward we might prefer to focus exclusively on primary points (goals and first assists) to get a more accurate prediction of a player's scoring rates.

Goals and assists hardly provide the full picture of a player's overall talent level, but they are good baseline statistics to use. In future work, we would like to focus on other, more robust statistics, like expected goals (xG). This statistic eliminate some of the noise caused by shooting luck, so we theorize that predictions would improve even further if we used it for our estimates instead.

In this thesis, we set out to show that empirical Bayes methods demonstrate a superior alternative with regards to estimation of scoring rates in hockey; specifically given data obtained over a relatively short time period. We were able to find that the Poisson-Gamma parametric model and the Kiefer-Wolfowitz nonparametric method displayed better predictive performance than a variety of other prediction methods. Surprisingly, we found that the "poor man's shrinkage" estimator provided good predictions as well.

There was no clear winner between the Poisson-Gamma and Kiefer-Wolfowitz methods, so in practice a linear combination of the two might be the best estimate to use. This linear combination could be suboptimal occasionally, but it would prevent egregiously bad estimates from occurring. Alternatively, one might decide to exclusively use the Poisson-Gamma model because it is

relatively easy to compute (especially the method of moments), and does not require Mosek. Of course this is just a recommendation based on one season of data; data from more seasons could be included in the future to see if our results are repeatable.

References

[1] M. ApS, *Mosek modeling cookbook*, 2020.

[2] A. Birnbaum, "On the foundations of statistical inference," *Journal of the American Statistical Association*, vol. 57, no. 298, pp. 269–306, 1962.

[3] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[4] B. P. Carlin and T. A. Louis, "Empirical Bayes: Past, present and future," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1286–1289, 2000.

[5] B. P. Carlin and T. A. Louis, *Bayesian methods for data analysis*. CRC Press, 2008.

[6] G. Casella and R. L. Berger, *Statistical inference*. Cengage Learning, 2021.

[7] M. L. Clevenson and J. V. Zidek, "Simultaneous estimation of the means of independent Poisson laws," *Journal of the American Statistical Association*, vol. 70, no. 351a, pp. 698–705, 1975.

[8] B. Efron, "Why isn't everyone a bayesian?" *The American Statistician*, vol. 40, no. 1, pp. 1–5, 1986.

[9] B. Efron, "Two modeling strategies for empirical bayes estimation," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 29, no. 2, p. 285, 2014.

[10] Glen, Stephanie, *Local Minimum (Relative Minimum); Global*, [Online; accessed June 27, 2022]. [Online]. Available: `https://www.calculushowto.com/local-minimum/`.

[11] Hasani, Heliya, [Online; accessed June 27, 2022], 2022. [Online]. Available: `https://medium.com/@heliyahasani/1-convex-sets-and-functions-modern-optimization-techniques-687736d78f80`.

[12] J. He, J. Rexford, and M. Chiang, "Design for optimizability: Traffic management of a future internet," in *Algorithms for Next Generation Networks*, Springer, 2010, pp. 3–18.

[13] W. Jiang and C.-H. Zhang, "General maximum likelihood empirical Bayes estimation of normal means," *The Annals of Statistics*, vol. 37, no. 4, pp. 1647–1684, 2009.

[14] J. Kiefer and J. Wolfowitz, "Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters," *The Annals of Mathematical Statistics*, pp. 887–906, 1956.

[15] R. Koenker and I. Mizera, "Convex Optimization, Shape Constraints, Compound Decisions, and Empirical Bayes Rules," *Journal of the American Statistical Association*, vol. 109, no. 506, pp. 674–685, 2014.

[16] N. Laird, "Nonparametric maximum likelihood estimation of a mixing distribution," *Journal of the American Statistical Association*, vol. 73, no. 364, pp. 805–811, 1978.

[17] G. F. Lawler, *Introduction to stochastic processes*. Chapman and Hall/CRC, 2018.

[18] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.

[19] D. V. Lindley and L. D. Phillips, "Inference for a Bernoulli process (a Bayesian view)," *The American Statistician*, vol. 30, no. 3, pp. 112–119, 1976.

[20] J. S. Maritz and T. Lwin, *Empirical Bayes methods*. Chapman and Hall/CRC, 2018.

[21] I. Mizera, "Lecture 6: Empirical Bayes methods,"

[22] L. Mou. "Lili Mou Machine Learning Course - Class 4: Convexity," Youtube. (2020), [Online]. Available: `https://www.youtube.com/watch?v=jAhBhan6Ybg&list=PLKlhhkvvU8-bEzt3Cu1xcvdoyNtZWoNok&index=4`.

[23] J. Orloff and J. Bloom, "Conjugate priors: Beta and normal. Class 15, 18.05, Spring 2014,"

[24] QuantHockey, "NHL Scoring Leaders 2018-19," 2019. [Online]. Available: `https://www.quanthockey.com/nhl/nationality-totals/nhl-players-career-stats.html`.

[25] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: `http://www.R-project.org/`.

[26] H. E. Robbins, "The empirical Bayes approach to statistical decision problems," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 1–20, 1964.

[27] H. E. Robbins, "An empirical Bayes approach to statistics," in *Breakthroughs in statistics*, Springer, 1992, pp. 388–394.

[28] D. J. Spiegelhalter, L. S. Freedman, and M. K. Parmar, "Bayesian approaches to randomized trials," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 157, no. 3, pp. 357–387, 1994.

[29] S. Tao, "Convex Duality in Nonparametric Empirical Bayes Estimation and Prediction," M.S. thesis, University of Alberta, 2014.

[30] S. Tao, "Theoretical and computational aspects of mixture models, with applications to empirical Bayes methods," Ph.D. dissertation, University of Alberta, 2018.

Appendix A

R Code

The following R script is what was explained in Subsection 2.2.1:

```
# import excel files
library(readxl)
library(tidyverse)
NHL_SL_2_Weeks <- read_excel("NHL_SL_2_Weeks.xlsx")
NHL_SL_3_Weeks <- read_excel("NHL_SL_3_Weeks.xlsx")
NHL_SL_4_Weeks <- read_excel("NHL_SL_4_Weeks.xlsx")
NHL_SL_5_Weeks <- read_excel("NHL_SL_5_Weeks.xlsx")
NHL_SL_6_Weeks <- read_excel("NHL_SL_6_Weeks.xlsx")
NHL_SL_7_Weeks <- read_excel("NHL_SL_7_Weeks.xlsx")
NHL_SL_8_Weeks <- read_excel("NHL_SL_8_Weeks.xlsx")
NHL_SL_fs <- read_excel("NHL_SL_fs.xlsx")

NHL_SL_2_Weeks$TOI <-
sapply(strsplit(NHL_SL_2_Weeks$TOI,":"), function(x) {
                                          x <- as.numeric(x)
                                          x[1]+x[2]/60 })

NHL_SL_3_Weeks$TOI <-
sapply(strsplit(NHL_SL_3_Weeks$TOI,":"), function(x) {
                                          x <- as.numeric(x)
                                          x[1]+x[2]/60})
```

```
NHL_SL_4_Weeks$TOI <-
sapply(strsplit(NHL_SL_4_Weeks$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60 })

NHL_SL_5_Weeks$TOI <-
sapply(strsplit(NHL_SL_5_Weeks$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60})

NHL_SL_6_Weeks$TOI <-
sapply(strsplit(NHL_SL_6_Weeks$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60 })

NHL_SL_7_Weeks$TOI <-
sapply(strsplit(NHL_SL_7_Weeks$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60})

NHL_SL_8_Weeks$TOI <-
sapply(strsplit(NHL_SL_8_Weeks$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60 })

NHL_SL_fs$TOI <-
sapply(strsplit(NHL_SL_fs$TOI,":"), function(x) {
                                        x <- as.numeric(x)
                                        x[1]+x[2]/60})
```

```
# filter out all players that played less than 60 games
NHL_SL_fs <- NHL_SL_fs %>% filter(GP >=60)

#filter out all players that aren't in all 5 tables
NHL_SL_fs <- NHL_SL_fs %>% filter(Name %in% NHL_SL_2_Weeks$Name)
NHL_SL_8_Weeks <- NHL_SL_8_Weeks %>% filter(Name %in% NHL_SL_fs$Name)
NHL_SL_7_Weeks <- NHL_SL_7_Weeks %>% filter(Name %in% NHL_SL_8_Weeks$Name)
NHL_SL_6_Weeks <- NHL_SL_6_Weeks %>% filter(Name %in% NHL_SL_7_Weeks$Name)
NHL_SL_5_Weeks <- NHL_SL_5_Weeks %>% filter(Name %in% NHL_SL_6_Weeks$Name)
NHL_SL_4_Weeks <- NHL_SL_4_Weeks %>% filter(Name %in% NHL_SL_5_Weeks$Name)
NHL_SL_3_Weeks <- NHL_SL_3_Weeks %>% filter(Name %in% NHL_SL_4_Weeks$Name)
NHL_SL_2_Weeks <- NHL_SL_2_Weeks %>% filter(Name %in% NHL_SL_3_Weeks$Name)

NHL_SL_fs <- NHL_SL_fs %>% filter(Name %in% NHL_SL_4_Weeks$Name)
NHL_SL_8_Weeks <- NHL_SL_8_Weeks %>% filter(Name %in% NHL_SL_fs$Name)
NHL_SL_7_Weeks <- NHL_SL_7_Weeks %>% filter(Name %in% NHL_SL_8_Weeks$Name)
NHL_SL_6_Weeks <- NHL_SL_6_Weeks %>% filter(Name %in% NHL_SL_7_Weeks$Name)

# filter out any irrelevant statistics
two_week_NHL_data <- NHL_SL_2_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
three_week_NHL_data <- NHL_SL_3_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
four_week_NHL_data <- NHL_SL_4_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
five_week_NHL_data <- NHL_SL_5_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
six_week_NHL_data <- NHL_SL_6_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
seven_week_NHL_data <- NHL_SL_7_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
eight_week_NHL_data <- NHL_SL_8_Weeks %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)
fs_NHL_data <- NHL_SL_fs %>% select(Name,Team,Age,Pos,GP,G,A,P,TOI)

# write as csv text files
write.csv(two_week_NHL_data,file = "two_week_NHL_data")
write.csv(three_week_NHL_data,file = "three_week_NHL_data")
write.csv(four_week_NHL_data,file = "four_week_NHL_data")
```

```
write.csv(five_week_NHL_data,file = "five_week_NHL_data")
write.csv(six_week_NHL_data,file = "six_week_NHL_data")
write.csv(seven_week_NHL_data,file = "seven_week_NHL_data")
write.csv(eight_week_NHL_data,file = "eight_week_NHL_data")
write.csv(fs_NHL_data,file = "fs_NHL_data")
```

The following R Scripts contain the functions used to obtain the results:

kwg.R

```
kwg <- function(E, ny = rep(1,dim(E)[1]),
                method="primal", opts=list(verbose = 5))
## ---------------------------ConeMosek-------------------------------
##  Kiefer-Wolfowitz estimator of mixture distribution/prediction
##  likelihood is evaluated at the "rows of E"
##      hence the number of rows of E must be y., the length of y
##      vector n.y records multiple occurrences of components of y
##        - multiplicity 0 should be taken special care of
##  the output is mixing distribution p, a vector with length p.
##      equal to the number of columns of E
##
##  (a) general primal formulation: vars  z  f  p  ons (=1)
##                                        y. y. p.  y.
##
##      sum -z -> min  -z >= -log f   f = Ep   sum p_i = 1  p >= 0
##                       z <= log f              (and also f >= 0)
##
##  (b) general dual formulation: vars    z   w   ons (=1)
##                                        nz. nz. nz.
##
##      sum z -> max   z <= log w    E'w <= n  w >= 0
## -----------------------------------------------------------------
```

```
{
  require(Matrix)
  require(Rmosek)

  y. <- dim(E)[1]
  p. <- dim(E)[2]
  nn <- sum(ny)
  nz <- ny > 0     # where ny is not zero; only those enter the dual
  nz. <- sum(nz)   # how many of those; lengths of vars in the dual
  nzy <- ny[nz]    # ny[ny > 0]; only essential ny then

#### Dual objective in the old-style Mosek
##   cns <- matrix(list(), nrow=3, ncol=sum(nzi))
##   opro[1,] <- "LOG"
##   opro[2,] <- (1:y.)[nz]
##   opro[3,] <- -nzy
##   opro[4,] <- 1/nzy
##   opro[5,] <- 0

  if (method == "dual") {
    prn <- list(sense="max")
    prn$c <- c(nzy,rep(0,2*nz.))
    prn$A <- cbind(spMatrix(p.,nz.),               # [0 Eo' 0]
                   t(E[nz,]) %*% diag(nzy), # more precise than t(E[nz,])
                   spMatrix(p.,nz.))
    prn$bc <- rbind(rep(-Inf,p.),                  # lower
                    rep(nn,p.))                    # upper
    prn$bx <- rbind(c(rep(-Inf,nz.),rep(0,nz.),
                      rep(1,nz.)),                 # lower
                    c(rep(Inf,2*nz.),
                      rep(1,nz.)))                 # upper
    prn$cones <- matrix(list(), nrow=2, ncol=nz.)
```

```
  rownames(prn$cones) <- c("type","sub")
  for (k in 1:nz.)                              # (w,ons,z)
    prn$cones[,k] <- list("MSK_CT_PEXP",c(nz.+k,2*nz.+k,k))
} else {
  prn <- list(sense="min")
  prn$c <- c(-ny,rep(0,y.+p.+y.))
  prn$A <- rbind(cbind(spMatrix(y.,y.),             # [0 -I E 0]
                       -Diagonal(y.), E,
                       spMatrix(y.,y.)),
                 c(rep(0,2*y.),rep(1,p.),rep(0,y.))) # [00 00 11 00]
  prn$bc <- rbind(c(rep(0,y.),1),                   # = [00 1]
                  c(rep(0,y.),1))
  prn$bx <- rbind(c(rep(-Inf,y.),rep(0,y.+p.),      # -ooo 00 00 11
                    rep(1,y.)),
                  c(rep(Inf,2*y.+p.),rep(1,y.)))    # +ooo +ooo +ooo 11
  prn$cones <- matrix(list(), nrow=2, ncol=y.)
  rownames(prn$cones) <- c("type","sub")
  for (k in 1:y.)                                # (f,ons,z)
    prn$cones[,k] <- list("MSK_CT_PEXP",c(y.+k,2*y.+p.+k,k))
}
#### Need to find equivalent of this in new Mosek
## prn$dparam$intpnt_nl_tol_rel_gap <- 1e-12

MOB = mosek(prn, opts=opts)

if (method == "dual") {
  w <- MOB$sol$itr$xx[(nz.+1):(2*nz.)]
  wc <- MOB$sol$itr$skc
  xc <- MOB$sol$itr$xc
  p <- -MOB$sol$itr$suc
  f <- E %*% p
  kwg <- list(p=p, f=f, w=w, wc=wc, xc=xc, E=E, ny=ny, method="dual")
```

```
  } else {
    f <- MOB$sol$itr$xx[(y.+1):(2*y.)]
    p <- MOB$sol$itr$xx[(2*y.+1):(2*y.+p.)]
    kwg <- list(p=p, f=f, E=E, ny=ny, method=method)
  }
}
```

postmea.R

```
pstmea <- function(x,mus)
  {
    (x$E %*% (mus*x$p))/(x$E %*% x$p)
  }
```

fprkw.R

```
fprkw <- function(cnts, epchs, lmbs=seq(0,max(cnts/epchs)+1,len=3000),
                  method="primal", opts=list(verbose=5))
### feeds the kwg.R with the appropriate A matrix, to estimate
### the mixing distribution in the setting of Poisson mixture, and
### then returns the posterior mean predictions (and other things)
  {
    cnts. <-  length(cnts)
    A <- matrix(0, nrow=cnts., ncol=length(lmbs))
    for (k in 1:nrow(A))
      A[k,] <- dpois(cnts[k], lmbs*epchs[k])
    hp <- kwg(A, method=method, opts=opts)
    prd <- pstmea(hp, lmbs)
    fprkw <- list(A=hp$E, prds=prd, pros=hp$p, lmbs=lmbs)
  }
```

fp.R

```
fp <-  function(cnts, epchs, shrink=1, method="mle")
```

```
### computes naive estimates (cnts/epchs) and
### estimates of compound lambda via method of moments or mle
### and shrinks them toward those depending on shrink parameter
{
  naiv <- cnts/epchs

  if (method == "mle")
    lam <- sum(cnts)/sum(epchs)
  else
    lam <- mean(naiv)

  fp <- list(lam=lam, shrink=shrink,
             prds=(1-shrink)*naiv+shrink*rep(lam,length(cnts)))
}
```

fpg.R

```
fpg <-  function(cnts, epchs, method="mle")
### fitting Poisson-Gamma mixture via method of moments or mle
{
  ## this is method of moments
  ratpg <- mean(cnts/epchs)
  scndm <- mean(cnts*(cnts-1)/epchs^2)
  bercp <- scndm/ratpg - ratpg
  bet <- if (bercp <=0) Inf else 1/bercp
  alp <- bet*ratpg

  ## which for the mle via nlm() provides starting values...
  if (method == "mle") {
    ## negbinomial -loglikelihood
    ff <- function(x)
      sum(-log(dnbinom(cnts,size=x[1],prob=x[2]/(x[2]+epchs))))
```

```
    if (bercp <= 0)
      pest <- nlm(ff,p=c(ratpg,1))$estimate # ...if beta is Inf
    else
      pest <- nlm(ff,p=c(alp,bet))$estimate # ...if both are finite
    alp <- pest[1]
    bet <- pest[2]
  }

  fpg <- list(alpha=alp, beta=bet, method=method,
              prds=(as.vector(alp)+cnts)/(as.vector(bet)+epchs))
}
```