SmartBP - Blood Pressure Access For Everyone: An approach through online learning of ranking functions for cuffless devices

Technical Report

Jessica De Souza*
Department of Electrical & Computer Engineering
University of California San Diego
La Jolla, CA 92093
jdesouza@eng.ucsd.edu

Nathan E. Liittschwager
Department of Computer Science & Engineering
University of California San Diego
La Jolla, CA 92093
nliittsc@eng.ucsd.edu

Abstract

This work aims to develop a low-cost finger-based Blood Pressure (BP) assessment tool for adults. This solution is based on an apparatus with a heart rate sensor associated with a force sensor for measure the finger. Our premise is that the association of heart rate and force, both gathered from the finger, will provide enough information for accurate BP measurement. Currently, available blood pressure cuff devices are precise however, still inaccessible by a majority of people in developing countries, and they can be inconvenient for carrying around for specific measurements. In this study, we verified the accuracy of our solution from parametric fitting and machine learning, which showed optimistic results when compared to a regular cuff-based device. In particular, we adapt a new algorithm normally used for image search, from Google Research, and show that it can robust estimate blood pressure through a nearest-neighbors approach. We call the device SmartBP, and the algorithm RankBP.

Keywords Blood Pressure · Finger Oscillometry · Machine Learning

1 Introduction

High blood pressure, or hypertension, is defined as systolic blood pressure (BP) with values above 130 mmHg and/or diastolic BP above 80 mmHg, is common and increases with age. This disease remains the leading preventable cause of premature death and disability worldwide, killing almost eight million people every year and is projected to increase by 60% to affect 1.6 billion adults worldwide by 2025 [4]. High-income countries have stable or decreasing rates, hypertension prevalence rates are increasing in low and middle income countries (LMICs) related to ageing of the population and increases in exposure to lifestyle risk factors including unhealthy diets and lack of physical activity. The access to blood pressure (BP) devices in these countries continues to be poor, often less than 10% [4], and it is concerning because BP assessment is an important factor to prevent hypertension. Migrants from LMICs, especially refugees, are especially vulnerable to poor BP control due to many post-migratory challenges such as navigating new healthcare systems, cultural and language barriers, and low socioeconomic status which make them more likely to have undetected or uncontrolled BP. Home BP monitoring for better BP control in refugees is believed to be a cost-effective and vital technique that can deal with the barriers of accessing and being retained in care. Moreover, there is growing research linking perceived stress, a common condition in refugees, to hypertension development and poor management. Besides ethnicity-based hypertension disparities such as in the case of refugees, hypertension management during pregnancy in low-resource settings has been a driver of maternal mortality in various LMICs [1]. Therefore, there is a need for innovative techniques that make measuring BP at home or in stressful, resource-limited settings and in socially disadvantaged populations cheaper, easier, safer and more accurate.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Hypertension is exceedingly common, particularly in older adults, where more than 70% of people with age 65 years and older had BP meeting the definition of hypertension in the National Health and Nutrition Examination Survey (NHANES) [5], yet hypertension control rates are lower among older people. In the same NHANES study, BP control to below 130/80 mmHg was achieved by 54, 50, 46, and 33% of people aged 20 to 54 years, 55 to 64 years, 65 to 74 years and 75 or more years, respectively [5]. Uncontrolled hypertension is associated with an increase in adverse cardiovascular and renal outcomes and is the most important modifiable risk factor for premature cardiovascular disease [7]. Older adults have additional challenges with advancing age including increasing multimorbidity and limitations that affect mobility, dexterity, vision and hearing. Given the prevalence of uncontrolled hypertension and additional challenges in the older adult population, it is essential that measurement of BP be done easily and accurately, but at the same time usable by themselves with little to no training and can be operated with limited dexterity.

Heart rate and force sensors can be utilized in low-resource settings globally and in the US with vulnerable populations experiencing hypertension disparities, such as refugees and other low income groups. Those sensors are easily found online and can be used combined with simple microcontrollers. Some existing solutions to date are built in a way that require extensive calibration per user [6] or require additional electronic devices [2]. The objective of this work is to implement a BP measurement device using embedded devices and sensors, and we will replicate the work from Chandrasekhar et al as a start point and for validation of the hardware setup. Also, we aim to implement machine learning algorithms that will also perform blood pressure measurement using the same device as a data source. We envision that such a tool would be cheap enough to be provided even in low-resource settings such as mobile clinics and refugee camps directly to anyone who needs to regularly monitor their blood pressure. Our work is summarized in the following contributions:

- Replication of the work from Chandrasekhar et al [2].
- Assembly of an apparatus for blood pressure measurement with real time data acquisition on the computer.
- Implementation of several machine learning algorithms: *Random Forest*, *k-nearest neighbors* and an implementation of a new online-learning ranking algorithm from Google Research, called *OASIS*.

2 Technical Material

The development of the SmartBP device is split into hardware and software implementation, data collection and analysis. The following subsections will better explain the process of each step.

2.1 Hardware

For creating the SmartBP device, we used the pulse oximeter and heart-rate sensor for health MAX30102, and the calibrated force sensor from Single Tact 15MM, 4.5N/1.0LB. Both sensors require I2C communication for sending the sensor readings, therefore we chose a microcontroller Arduino Yun for communicating with the sensors. In Figure 1 we can see the model of the sensors used. The blood pressure is measured by using the applied force and heart rate collected at the same time, therefore, we are creating our setup in a way that both sensors are placed in the top of each other. In Figure 2 we can see the scheme of the hardware implementation with the Arduino.

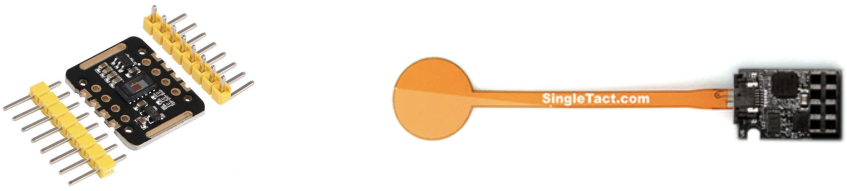

Figure 1: Heart rate sensor MAX30102 (left), and Single Tact force sensor (right).

Between the force sensor and the PPG sensor we placed a rubber material in the same diameter of the force sensor, so the applied force would be centered in the correct place. Also, for stabilizing the finger placement and to reduce the noises due to motion and finger misplacement, we made a thin plastic cover that also helped to concentrate the LED light on the finger.

The sampling rate of the sensors were at 10 samples per second, with microcontroller baud rate at 57600 symbols per second. The way we used the I2C communication was by setting each sensor at a different address, and then the

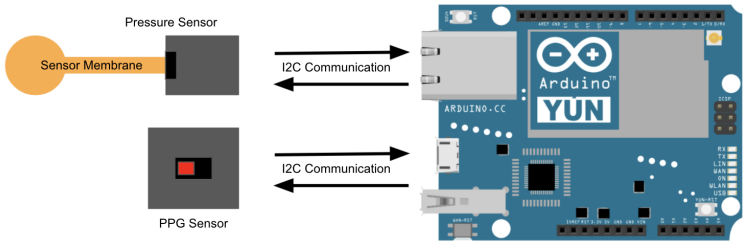

Figure 2: Components used in the implementation.

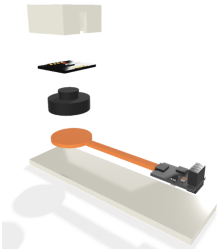

Figure 3: Exploded view of the SmartBP structure for sensor placement.

Arduino could know where is the data coming from. Parameters like LED brightness, ADC sampling and other details for improving the PPG sensor were customized accordingly.

2.2 Methodology

In this section, we provide an overview of our methodology for estimating blood pressure given an oscillometric signal. We provide two novel methods, building on the work of the authors in [2]. First, we deploy a *Random Forest* predictive model that is trained on a combination of data we acquired from our experiments, as well as publicly available data set. Second, we consider the problem of telemedicine more generally, and show that blood pressure estimation can be solved as an information retrieval task. We argue that such a point of view is favorable in our situation, as it can be applied more generally to other biomedical signals generated from telemedicine applications.

2.2.1 Learning to Estimate Blood Pressure

Estimating blood pressure from an oscillometric PPG signal is a common task, and the most popular methods employed in cuff-based devices tend to either be based on physical models, or empirical methods [8]. The authors show that when implemented in cuff-devices, these mathematical models and simple regression formulas can be extremely accurate, with biases within ± 1 mmHg. It stands to reason that such models of blood pressure can be used in the finger-pressing oscillometric method, as the underlying anatomical processes are similar, as argued in previous work [2].

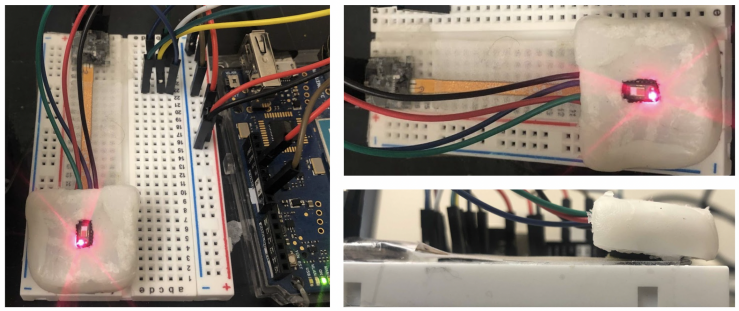

Figure 4: Real hardware implementation of the SmartBp system.

However, while similar, the underlying anatomical processes in the finger *are* different than in the arm, and application of physics-based mathematical models lacks formal justification. Empirical methods based on linear regression formulas are promising, but such methods are estimated with step-wise regression, which has no formal justification for its use, and is well known to not solve the problem it claims to, and is out of favor among serious statisticians and computer scientists [9], [10], [11].

Estimating a true measurement $\mathbf{y}$ from a noisy signal $\mathbf{x}$ is a classical problem for modern machine learning methods. In most cases, with signal processing data, the vector $\mathbf{x}$ is a noisy measurement of $\mathbf{y}$, and we are interested in predicting the *true measurement given the data*: $p(\mathbf{y}|\mathbf{x})$, where $p(\cdot|\cdot)$ is a conditional probability density. In frequentist terms (which we will use throughout the rest of the paper), the problem is to estimate $\mathbf{y}$ with a vector $\hat{\mathbf{y}}$ such that $\ell(\mathbf{y}, \hat{\mathbf{y}})$ is minimized, where $\ell(\cdot, \cdot)$ is a *loss function*. However, the setting of blood pressure estimation is slightly different. With blood pressure measurement through an oscillometric method, $\mathbf{y}$ is a small real-valued vector: $\mathbf{y} \in \mathbb{R}^2$, but $\mathbf{x} \in \mathbb{R}^n$ is a sequence of n PPG sensor measurements sampled through time that are causally related to cuff-pressure, at each time step. The estimation problem is to *learn* how a long sequence $\mathbf{x}$ can relate to a vector $\mathbf{y}$, which is different from merely trying to solve an estimation problem. In the deep learning, this prediction task is called *sequence-to-sequence* modeling or *seq2seq* for short [12]. While being able to deploy such deep learning models would be ideal, the problem of trying to accurately estimate blood pressure from an finger-pressing oscillometric signal is so new, that there is not enough data to justify such a powerful learning method.

In our case, the signal $\mathbf{x}$ is indexed by time, and can be written as $\mathbf{x}_t$, and the task of predicting (but not *forecasting* a scalar or small vector $\mathbf{y}$ has very recently gained traction in the machine learning community as *Time Series Extrinsic Regression* [13], and is expected to be a fruitful area of research.

The authors in the aforementioned work showed that *Random Forests* were a powerful method for this class problems. Random forests are a versatile and powerful modern machine learning method for tabular data [14]. Because they form an *ensemble* of decision trees on bootstrapped subsections of data sample and data features, they are robust to overfitting, and often provide high level performance with default settings. Since Random Forests are a commonly used machine learning method, we direct the interested reader to [14] for more details.

2.2.2 Blood Pressure Estimation as Information Retrieval

In this section, we consider the problem of blood pressure estimation as *information retrieval*.

Information retrieval has a large body of work, and a large number of applications, such as speech processing, music-retrieval, web search, etc [15], [16], [17]. In this paper, it is the following task: given a query vector $\mathbf{q}$, and database $\mathcal{D}$, retrieve the set of k-nearest neighbors $\mathbf{x}_1, ..., \mathbf{x}_k$ according to some similarity metric $S(\cdot, \cdot) \in \mathbb{R}$, where if $S(\mathbf{q}, \mathbf{x}_i) > S(\mathbf{q}, \mathbf{x}_j)$, then the query vector $\mathbf{q}$ is *more similar* to the vector $\mathbf{x}_i$ than $\mathbf{x}_j$.

If $N_k(\mathbf{q}, S, \mathcal{D})$ is the set of k nearest neighbors according to S, then in machine learning settings, information retrieval is often used to predict a hidden target label $\mathbf{y}_q$ of the query as a function of the retrieved neighbors – usually each neighbor $\mathbf{x}_i$ also comes with a *known* target label $\mathbf{y}_i$. In regression settings, the prediction is an average of the k-nearest neighbors:

$$\mathbf{y}_q \approx \frac{1}{k} \sum_{i \in N_k(\mathbf{q}, S, \mathcal{D})} \mathbf{y}_i.$$

Under the assumption that *people with similar blood pressures have similar oscillograms* (which is the foundation of oscillometric blood pressure measurement), it can be seen that blood pressure estimation can be reduced to information retreival: given a query oscillometric PPG signal $\mathbf{q}$, a similarity metric S, and a database $\mathcal{D}$, we can predict the blood pressure $\mathbf{y}_q$ of the patient corresponding to $\mathbf{q}$ by finding the k-nearest neighbors $N(\mathbf{q}, S, \mathcal{D})$. This is often called a *ranking problem.*

2.2.3 RankBP: Learning to Rank Blood Pressure

While standard similarity metrics S such as the *euclidean distance* exist, it is more interesting (and more useful) to try and *learn* a similarity metric from data, so that it may give the best results for each application at hand. The approach taken in this section is largely inspired by a work from Google Research [18]. We provide a brief overview, adapting their notation to our setting. From this point on, we write symbols $\mathbf{q}$ and $\mathbf{x}$ as q and x, understanding that they refer to d-dimensional vectors.

Given a query q_i, we would like to retrieve the k-nearest neighbors corresponding to the bilinear similarity function $S_{\mathbf{W}}$. I.e. $S_{\mathbf{W}}(q_i, x_j) = q_i^T \mathbf{W} x_j$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$. The authors at Google Research proposed *OASIS* (Online Algorithm for Large Scale Similarity Learning) for the purposes of image search retrieval. Image search often consists of flattening

a tensor of pixels into a vector, and then embedding the vector into a feature space. This procedure is analagous to the situation we find ourselves in: retrieving similar signals from a database. We adopt *OASIS* to our setting.

The algorithm works as follows: given q_i, there are examples $q_i^+ \in \mathcal{D}^+$ and $q_i^- \in \mathcal{D}^-$ denoting vectors that are considered *more* similar to q_i and *less* similar to q_i. I.e., $S_{\mathbf{W}}(q_i, q_i^+) > S_{\mathbf{W}}(q_i, q_i^-) + 1$, where the 1 acts as a 'safety margin'. The goal is to *learn* the similarity matrix $\mathbf{W}$ that maximizes the similarity of q_i and q_i^+ and minimizes the similarity of q_i and q_i^-. Thus each learning example is a triple: (q_i, q_i^+, q_i^-).

The ranking is *relative*, meaning that exact labels are not required. This is a useful condition in our setting: where data is scarce, noisy, but expected to accumulate rapidly over time as *SmartBP* eventually gets adapted for general use. Labeling data sets is costly, but since later iterations of *SmartBP* can be expected to interact with doctors, the expertise of doctors can provide helpful *relative* ranking of patients: a doctor need only say that patient i has 'high blood pressure' and that patient j has 'low blood pressure' (say through interacting with an app), and we obtain a useful ranking without any need for a doctor to painstakingly label datasets.

To learn $\mathbf{W}$, a scoring function is needed, and we use the one given in the paper:

$$\ell_{\mathbf{W}}(q_i, q_i^+, q_i^-) \equiv \max(0, 1 - S_{\mathbf{W}}(q_i, q_i^+) + S_{\mathbf{W}}(q_i, q_i^-))$$

The global loss over the database $\mathcal{D}$ is defined as $L_{\mathbf{W}} \equiv \sum_{(q_i, q_i^+, q_i^-) \in \mathcal{D}^3} \ell_{\mathbf{W}}(q_i, q_i^+, q_i^-)$.

Minimization of this loss is done with the online *passive aggressive* algorithm similar to that in [19]. First, $\mathbf{W^0}$ is initialized as the $d \times d$ identity matrix. Then, on each step of the algorithm, a triplet $(q_i, q_i^+, q_i^-) \in \mathcal{D}^3$ is randomly drawn, and the following optimization problem is solved:

$$\mathbf{W}^i = \text{argmin}_{\mathbf{W}} ||\mathbf{W} - \mathbf{W}^{i-1}||_{Fro}^2 + C\xi \quad \text{s.t.} \quad \ell_{\mathbf{W}}(q_i, q_i^+, q_i^-) \leq \xi \quad \text{and} \quad \xi \geq 0. \tag{1}$$

This is a convex problem with a *soft-margin* and admits an analytical solution. We skip the derivations, but point out that it can be done with lagrange multipliers, as shown in [19] and [18]. Concretely, we just need to make the following updates:

$$\mathbf{V}_i = [q_i^1(q_i^+ - q_i^-), ..., q_i^d(q_i^+ - q_i^-)]^T \tag{2}$$

$$\tau = \min \left\{ C, \frac{\ell_{\mathbf{W}^{i-1}}(q_i, q_i^+, q_i^-)}{||\mathbf{V}_i||^2} \right\} \tag{3}$$

$$\mathbf{W}^i = \mathbf{W}^{i-1} + \tau \mathbf{V}_i. \tag{4}$$

These updates can largely be seen as online stochastic gradient descent, with a gradient $\mathbf{V}_i$ with step-size τ. The parameter C is an *aggressiveness* parameter (hence *passive-aggressive* learner – the *passive* part is from the choice of loss function that is *passive* when predictions are correct), which enforces how quickly the algorithm learns $\mathbf{W}$. We also follow the advice given in [18], and add a projection step to force $\mathbf{W}$ after learning. This can be done with any standard method, but the authors recommend to use the eigenvalue decomposition: $\mathbf{W} = \mathbf{VDV}^T$ and forcing the eigenvalue matrix $\mathbf{D}$ to have positive values.

The next question is: how do we generate (q_i, q_i^+, q_i^-)? With a lack of domain knowledge, relative rankings suffice. However, one important thing to note wrt blood pressure is that the medical community at large considers blood pressure to be an ordinal categorical variable. The following bullet point list comes from American College of Cardiology [20], giving new guidelines on blood pressure:

- Normal: Less than 120/80 mm Hg;
- Elevated: Systolic between 120-129 and diastolic less than 80;
- Stage 1: Systolic between 130-139 or diastolic between 80-89;
- Stage 2: Systolic at least 140 or diastolic at least 90 mm Hg;
- Hypertensive crisis: Systolic over 180 and/or diastolic over 120, with patients needing prompt changes in medication if there are no other indications of problems, or immediate hospitalization if there are signs of organ damage.

Ultimately we would like this order to be maintained. Thankfully, a simple euclidean distance of the blood pressure vectors $y_i \in \mathbb{R}^2$ maintains the total order. While we do not report the results here, we would like to point out that categorical encodings with $y_i \in \{0, 1, 2, 3, 4\}$ (e.g., $y_i = 0$ means 'normal" and $y_i = 3$ means 'stage 1') were competitive. While worse than having exact ground truth, this encoding scheme easily defeats the baseline we consider later. The ease of labeling large amounts of data with a relative ranking rather than taking exact ground truth remains a strength of our approach.

2.2.4 Experiments

The objective of performing experiments is to validate the accuracy of the device we developed and also compare the blood pressure measurements with the work from Chandrasekhar et al, the cuff-based device that is our ground truth, and the veracity of our approaches.

Dataset Collection. We used the device from this work (the SmartBP) combined with a cuff-based device (Omron 7 Series Wireless Upper Arm Blood Pressure Monitor) which is our ground truth. The participant needed to be seated with symmetric body position, and at least 5 minutes rested. The procedure for the data acquisition occurred as follows: (1) the participant after understanding how the device works and performed two practice sessions with the SmartBP device, places the index finger (top knuckle) on the sensing area, where the base of the nail touches the PPG LED. We then start the timer and the data collection, where there is 30 seconds of data acquisition with the patient steady and slowly increasing the applied pressure from the finger on the sensing area. Once the 30 seconds are over, the algorithm shows the estimated blood pressure, the user removes the finger from the device and we start measuring the blood pressure from the ground truth.

We obtained two datasets. The first is a dataset of 62 measurements from 5 volunteers. Their oscillometric signals were recorded with our device, and their ground truth labels obtained directly after. The second is a publicly available dataset from *IEEE DataPort* [21], which contains 350 oscillometric measurements from cuff-devices, as well as the corresponding estimated blood pressures. In order to address the scarcity and lack of variety in our data, we use this extra dataset to supplement our own, giving a total of 412 PPG signals, and blood pressure labels. While not drawn from the same distribution, the justification is that the oscillometric finger pressing method is analagous to the oscillometric recordings of the cuff device. Thus, we can think of the extra data set as a way to inject *bias* into our models (thus allowing learning), as well as *variance* in our labels (thus getting an estimate of generalization error). Put another way, combining the two data sets can be justified as a form of *transfer learning.* The advantage of this approach is that it empirically shows that good predictions can be made using data from other sources – and not just data recorded on our device. This helps alleviate what is commonly called a *cold start* problem in the machine learning community.

Data Preprocessing and Modeling. Before doing any learning, the data is preprocessed. All signals are centered with zero mean and unit variance, then passed through a 2nd order butterworth bandpass filter, with cutoff frequencies of $[0.8, 3.5]$ Hz. Centering and scaling was primarily done to avoid numerical issues with the butterworth filter.

We compare 5 methods for blood pressure estimation. First, is the baseline oscillometric algorithm described in the *software* section of [2]. We follow their algorithm almost exactly, up to some minor differences in the filtering and preprocessing of the signals, to accomodate the differences between our device and theirs.

Second, we utilize a *Random Forest*. While in principle, a random forest could predict a the blood pressure $y_i \in \mathbb{R}^2$ directly from a ppg signal $\mathbf{x}_i \in \mathbb{R}^d$, this has been shown to not work well [13]. Instead, we slice each signal $\mathbf{x}_i$ into 10 'windows' $\tilde{\mathbf{x}}_{i_1}, ..., \tilde{\mathbf{x}}_{i_{10}}$, and from each window, extract a vector $\mathbf{z}_{i_j}$ of features. In particular, we extract the mean, standard deviation, min, max, skew, kurtosis, difference in start and end value, the most common spectral frequency, as well as autocorrelation coefficients on the first 4 lags. Thus each PPG signal $\mathbf{x}_i$ results a feature vector $\mathbf{v}_i$ with 120 features. Abusing notation, we construct the data set $\tilde{\mathbf{X}} = [\mathbf{v}_1, ..., \mathbf{v}_{412}]^T$ with ground truth labels $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_{412}]^T$, and the random forest is learned using 100 random trees and no limit on max-depth, the number of estimators is chosen by cross-validation.

Third, since we are interested in ranking problem, we also consider a k-Nearest Neighbor model using the extracted features $\tilde{\mathbf{X}}$, and the standard Euclidean distance. While we don't expect this method to beat the random forest, it will provide a suitable baseline for our next two methods. Note that $\tilde{\mathbf{X}}$ is normalized to the real-valued interval $[0, 1]$ to allow fair comparison by Euclidean distance. We choose $K = 10$ neighbors by leave-one-out cross-validation.

Finally, we consider two versions of *RankBP*. In the first version, RankBP uses the extracted feature vectors $\tilde{\mathbf{X}}$ (after normalization), with an aggressiveness parameter of $C = 0.10$. The second version of RankBP uses simply the PPG signals themselves: $\mathbf{X}$. To accommodate for differences in length, a minor resampling step is done to ensure all PPG signals have the same length. In general, we recommend sampling at a higher granularity than intended by the model, then downsampling to the needed size. Training was done very fast, with roughly 2000 samples drawn from the data set (roughly equivalent to 5 entire passes with gradient descent – we noticed more iterations destroyed results).

Evaluation. We are interested in several metrics:

- Avg. Bias of each blood pressure measurement j: $\mu_j = \frac{1}{N} \sum_{i=1}^{N} y_{i_j} - \hat{y}_{i_j}$. In particular, systolic and diastolic blood pressure.

- Std. Deviations of each bias: $\sigma_j = \sqrt{\frac{\sum_i (y_{i_j} - \hat{y}_{i_j}) - \mu_j}{N-1}}$
- Precision: $\frac{tp}{tp+fn}$, Recall : $\frac{tp}{tp+fn}$, F_1-score: $\frac{tp}{tp+(fp+fn)/2}$.

Notice that the last three are classification metrics. The classification is made by first predicting a real-valued blood pressure $\hat{\mathbf{y}}_i \in \mathbb{R}^2$ for the ith signal. Then the prediction is compared to the recommendations given by the ACC, and a class in $\{0, 1, 2, 3, 4\}$ is predicted. These class predictions are compared to the class labels derived from the ground truth BP measurements $\mathbf{y}_i$. The classification is then binarized: $y_i \mapsto 0$ if $y_i \leq 1$ and otherwise $y_i \mapsto 1$. Thus classification is made on the basis of whether someone has stage 1 BP or higher.

We justify this choice as follows: when a doctor takes a patients blood pressure, they do not care about whether a patient has $123/82$ mmHg, or $121/79$ mmHg. They care whether the patient has $123/82$ mmHg, or $138/96$ mmHg. Thus fine-tuned 'granularity' is less important, but whatever prediction our model makes, *it must be mostly correct, most of the time.* One could argue that this is really a classification problem, but it is unlikely doctors would trust a learning algorithm that simply outputs "normal" or "hypertensive" without some measurement of how 'close' the patient is to "normal" or "hypertensive". Thus, we opt for this approach, as it is more realistic to the goals of medical professionals, even if it is less clean to analyze.

We can interpret the classification metrics as follows: *precision* is the positive predictive value: the probability that the model is *correct* when it says a patient has "high blood pressure". The recall is the *true positive rate*: the rate at which people are correctly identified as having high blood pressure. The F_1 score is a balance of both (higher is better).

For our statistical analysis of the results, we take a Bayesian point of view: we compute the posterior distribution of the bias estimates using a conjugate prior over a multivariate normal distribution. Using this, we compute 95% credible intervals for the bias: $0.95 = p(\ell < \mu < u \,; \mathcal{D}, \Theta)$, and the posterior probabilities that the machine learning methods saw a real improved upon the baseline wrt to the bias: $p(||\mu_i|| < ||\mu_{baseline}|| \,; \mathcal{D}, \Theta)$ where μ_i is the bias of the ith algorithm. This can be thought of as a simple AB-test to compare algorithm performance.

Note on evaluation. To be clear, it should be said that while the methods were *trained* with both datasets, *testing* was done with our personal dataset collected from volunteers. The extra dataset is used as a kind of transfer learning – each model sees more data and with larger variation, and so learns the neccesary patterns, but we evaluate our methods on our 'real' data. All results that follow are computed in this way, with leave-one-out cross validation, and bayesian posterior updates with weak priors.

2.3 Results

The statistics were computed using leave-one-out cross validation, a frequentist method of estimating generalization error, in each case. While there is some correlation in our voluntarily collected data set with repeated measurements, we believe the inclusion of the extra data as a form of transfer learning will make our results more robust and reduce overfitting.

Our results were both surprising and unsurprising in various ways.

First, each machine learning method was able to handily beat the baseline method, this includes the versions of RankBP which used categorical labels to encode the triplets (q_i, q_i^+, q_i^-) instead of real-valued labels. However, those versions of the model were the weakest so far – likely from lack of data – so we omit them.

What's interesting is that our implementation of the baseline algorithm from [2] does slightly worse: they reported biases of $3.3/-5.6$ mmHg for systolic/diastolic and smaller standard deviations. However, upon close inspection of their paper, we see that a crucial difference between their algorithm and ours is that their algorithm can repeatedly ask patients to start-over with the estimation process if the algorithm deems their data to be poor quality. This makes sense, as we had to throw out nearly 20% of the predictions from the baseline algorithm, due to numerical stability issues arising from noise in the data (even after filtering). Because the methods described in [2] rely on essentially computing an oscillogram, then using ratios of a fitted curve to estimate blood pressure, it makes sense that even small amounts of noise jeopardize this process (see Figure 5 for an example of a good and a bad oscillogram). Despite this, the precision and recall of their method compares well to the machine learning methods, indicating that the baseline algorithm is "wrong, but in the right direction".

On the other hand, the random forest does as expected, it achieves levels of bias and standard deviation lower than in [2]. It also gives the some of the highest precision, recall, and F_1 score of all the methods tried. This was expected, as it was shown in prior work that the random forest excels on this class of problem [13].

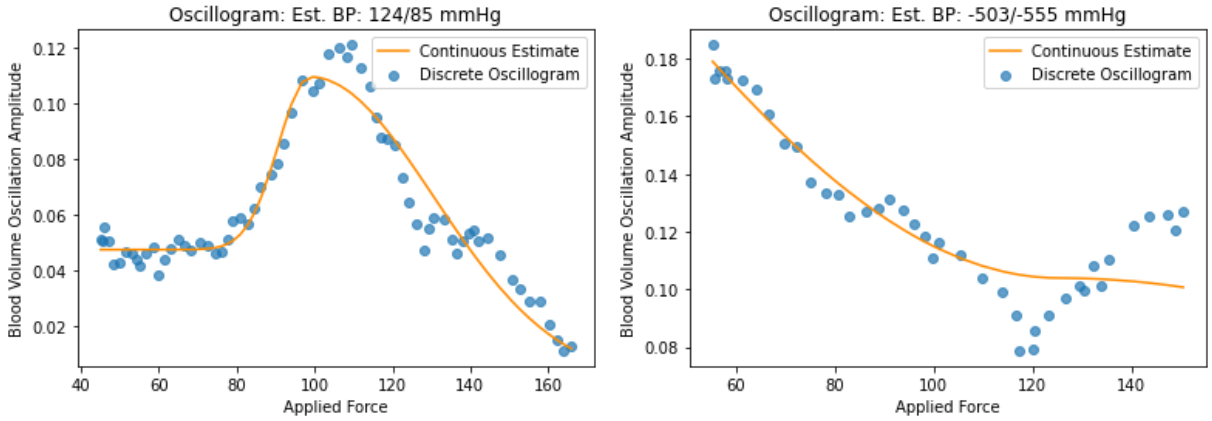

Figure 5: **(a)** An ideal oscillogram with the baseline method. **(b)** An example of what happens when an unstable signal is passed to the oscillogram. This prediction had to be thrown out, for obvious reasons.

Perhaps one of the more surprising results is the performance of k-nearest neighbors. We chose this model as a simple baseline to compare with our $RankBP$ method, but the results were stunning: It handily is one of the best methods for this problem, with very low biases and errors. However, the precision and recall is lacking compared to the other methods. This suggests that, while 'accurate' in an absolute sense, the model is 'inaccurate' in how well it identifies whether people really have high blood pressure or not. This would improve with data, as it is well known that k-nearest neighbors converges to the *Bayes error rate* in the limit of data. This model has the second highest probability of a real improvement (in terms of bias) over the baseline at 97%.

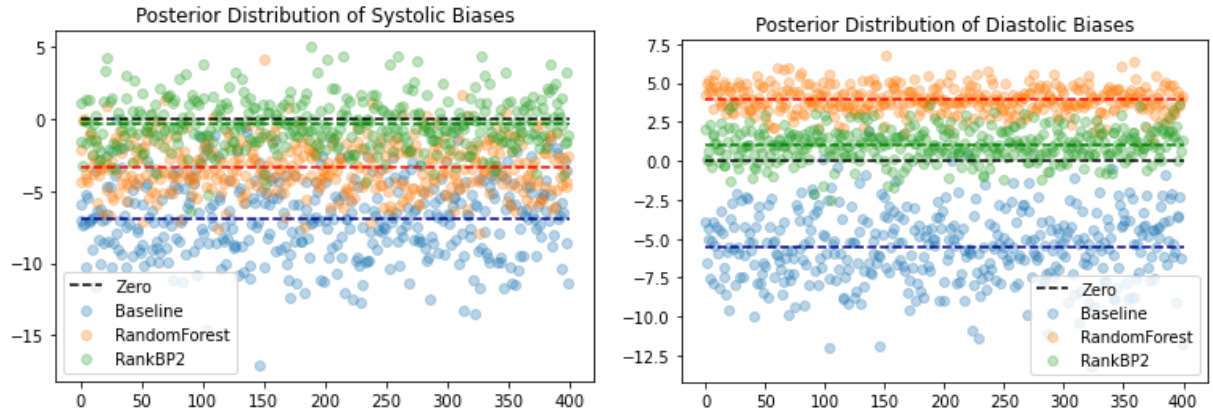

Figure 6: A visual representation of the posterior distribution of the systolic and diastolic biases, estimated after learning the algorithms. We only plot a subset of the models, for simplicity. Closer and 'tighter' to 0 is better.

Finally, the results of *RankBP* are encouraging, with very low biases and acceptable standard deviations. What is most heartening is the very passable recall rate of RankBP2, which uses the (normalized) data vectors $\mathbf{X}$ instead of the extracted feature vectors $\tilde{\mathbf{X}}$. What's most encouraging is the extremely low systolic and diastolic bias of the method. Figure 8 gives most of the relevant results: the 95% central credible intervals are tight, and the probability of a real improvement is 97.2%.

Training of RankBP One noticable effect of *RankBP* was its efficiency. RankBP converges in roughly 2000 iterations on a dataset of 412 signals. Since each iteration is a stochastic gradient descent update, this is faster than it would seem: roughly only 5 passes over the data was needed to achieve good results. In the future, when SmartBP is an actual phone application, this has distinct advantages: noisy gradients can be computed on an individuals phone, then messaged to a central location. The model updates can then be messaged out, resulting in potentially *personalized online learning* of biomedical information retrieval systems. This can be a huge benefit as the hypothetical user base of SmartBP grows.

Method	Systolic Bias	Diastolic Bias	Systolic Std.Dev	Diastolic Std.Dev	Precision	Recall	F1
Baseline	-6.936	-5.553	18.604	15.979	0.642	0.692	0.667
RF	-3.323	3.930	13.712	6.855	**0.700**	**0.778**	**0.736**
KNN	-2.483	1.020	13.023	7.705	0.619	0.722	0.666
RankBP	**-2.105**	6.655	12.761	8.314	**0.705**	0.666	**0.685**
RankBP_2	**-0.353**	**1.040**	13.680	7.923	0.642	**0.750**	**0.692**

Figure 7: The results of our experiments. Care should be taken when interpreting the baseline method: nearly 20% of the predictions had to be removed due to numerical problems.

Method	Sys. Bias C.I.	Dia. Bias C.I.	Sys. Prob. Improve	Dia. Prob. Improve	Prob. Overall Improve
Baseline	[-11.535, -2.34]	[-9.332, -1.839]	N/A	N/A	N/A
RF	[-6.053, -0.449]	[2.544, 5.341]	0.866	0.749	0.838
KNN	[-5.318, 0.183]	[-0.577, 2.586]	**0.991**	0.918	0.970
RankBP	[-4.873, 0.599]	[4.817, 8.419]	0.947	0.345	0.699
RankBP_2	**[-3.103, 2.576]**	**[-3.103, 2.576]**	**0.973**	**0.959**	**0.972**

Figure 8: **(a)** & **(b)** The 95% central **credible** intervals on the biases for systolic and diastolic mmHg, respectively. **(c)** & **(d)** The probability of algorithm improvement: $p(|\mu_i| < |\mu_{baseline}| \; |\mathcal{D}, \mathbf{\Theta})$ where $\mathbf{\Theta}$ are the posterior parameters of the multivariate gaussian distribution. **(e)** Probability of an overall improvement: $[(||\mu|| < ||\mu_{baseline}|| \; |\mathcal{D}, \mathbf{\Theta})$.

3 Milestones

In this section we briefly describe our milestones, what we achieved, and what we did not.

1. We *achieved* building our Minimum Viable Product: building a small device that can estimate blood pressure using just an oscillometric signal from the finger.
2. We *achieved* replicating prior work: our baseline method had comparable levels of bias to prior work. While our standard deviations were inflated, we believe this was because our implementation was not as willing to make users try multiple times to obtain a good measurement.
3. We *achieved* a novel adaptation of the prior work. We implemented several machine learning methods, two of them completely standard (Random Forests and KNN), and one of them known in the greater literature, but a novel take on the blood pressure estimation problem. Estimation via information retrieval, RankBP. We believe the latter approach will be more generally useful for other forms of estimation and search in telemedicine.
4. We *did not achieve* a smartphone application. While we initially wanted to try to achieve this as a stretch goal, we realized that such a task would take at least another 10-20 weeks, and likely more collaborators, as these authors are not familiar with building smartphone applications.
    - *What to do differently*: It would be more pertinent to start the project from scratch with a smartphone application in mind, rather than building a device, and hoping the smartphone application would somehow 'come together'. On the other hand, we have learned a great deal about what makes this idea work, and how it could be improved.
5. We *did not achieve* at least one hard deadline. In particular, this paper took at least 7 more hours to finish than we thought.
    - *What happened?* There was a very serious pivot right up to the deadline. The authors realized that a fundamental component in the data-collection algorithm was incorrect, and leading to very noisy, unusable results when trying the device on those who were not the authors. This had to be corrected, and required a significant time investment. Since the data collection phase was wrong, *all* of our prior results were wrong. We had to fix the problem, then collect data *all over again*, then analyze it *all over again.*
    - *What else happened?* The idea of *RankBP* was another big pivot in design. Close to the deadline, while trying a variety of machine learning methods, viewing the problem as a classification problem, it became obvious that no doctor or patient would accept an algorithm that simply answers "normal" or "not normal". The idea of *RankBP* was cooked up in the final days leading to our deadline. While late with the paper, we are happy we did it. The method is modern, interesting, powerful, scalable, and generalizable.
    - *How could we have avoided this?* This is a difficult question. On the one hand, we could not have predicted such a critical bug, so close to the deadline. On the other hand, we did not have a robust plan

for feature testing of the hardware. Likely, we could have avoided the issue had we extensively tested hardware before moving on to data-modeling.

4 Conclusion

This paper is the result of a 10-week long project to replicate prior work. While we named our project *SmartBP*, we really were replicating the work of Chandrasekhar et al. in [2], and then adding our own twist, which we call *RankBP*.

While the learning algorithm for *RankBP* is not new, we have applied it to a completely novel domain: blood pressure estimation through information retrieval in telemedicine. Our experiments provide a robust set of evidence that the improvements we saw over the baseline algorithm are real, and not a result of a statistical anamoly. We believe this type of learning scheme can also be applied more generally: information retrieval can be used to obtain relevant metrics, information, and predictions for *any* biomedical signal, and since the ranking algorithm learns in an efficient online manner, costly model training steps may be avoided. Moreover, the model is able to learn a similarity matrix $\mathbf{W}$ using *approximate, relative* ranking functions. This will be a boon in the early phases of telemedicine, where data is abound, but data labels are not. A doctor could simply provide noisy updates of "doing better than before" or "blood pressure is increasing over time", and that is relevant information for methods like *RankBP*.

Ultimately, we were able to achieve most of our goals, though not without some bloodshed - there were serious mishaps along the way with the construction and implementation of this device. However, we feel the project was a success, and hope that it can be iterated upon in the future, either by ourselves or others.

References

[1] Brown, M.A., Magee, L.A., Kenny, L.C., Karumanchi, S.A., McCarthy, F.P., Saito, S., Hall, D.R., Warren, C.E., Adoyi, G. and Ishaku, S. 2018. Hypertensive disorders of pregnancy: ISSHP classification, diagnosis, and management recommendations for international practice. Hypertension. 72, 1 (2018), 24–43.

[2] Chandrasekhar, A., Kim, C.S., Naji, M., Natarajan, K., Hahn, J.O. and Mukkamala, R. 2018. Smartphone-based blood pressure monitoring via the oscillometric finger-pressing method. Science Translational Medicine. 10, 431 (2018), 1–12.

[3] Liu, J., Cheng, H.M., Chen, C.H., Sung, S.H., Hahn, J.O. and Mukkamala, R. 2017. Patient-Specific Oscillometric Blood Pressure Measurement: Validation for Accuracy and Repeatability. IEEE Journal of Translational Engineering in Health and Medicine. 5, December 2016 (2017).

[4] Mills, K. T. et al. 2016. Global disparities of hypertension prevalence and control: a systematic analysis of population-based studies from 90 countries. Circulation 134, 441–450.

[5] Munter P, Carey RM, Gidding S, Jones DW, Taler SJ, Wright JT,Jr, Whelton PK, 2018. Potential US Population impact of the 2017/ACC/AHA High BLood Pressure Guideline. Circulation 137 (2) 109-118.

[6] Wang, E.J., Zhu, J., Jain, M., Lee, T.-J., Saba, E., Nachman, L. and Patel, S.N. 2018. Seismo: Blood pressure monitoring using built-in smartphone accelerometer and camera. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (2018).

[7] Wilson PW. Established risk factors and coronary artery disease: The Framingham Study 1994 Am J Hypertens Jul 7 (7 Pt 2): 7S-12S.

[8] Chandrasekhar A, Yavarimanesh M, Hahn JO, et al. Formulas to Explain Popular Oscillometric Blood Pressure Estimation Algorithms. Front Physiol. 2019;10:1415. Published 2019 Nov 21. doi:10.3389/fphys.2019.01415

[9] https://statmodeling.stat.columbia.edu/2014/06/02/hate-stepwise-regression

[10] Andrew Gelman, Matt Stevens & Valerie Chan (2003) Regression Modeling and Meta-Analysis for Decision Making, Journal of Business & Economic Statistics, 21:2, 213-225, DOI: 10.1198/073500103288618909

[11] Flom, P. and D. Cassell. "Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use." (2007).

[12] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14). MIT Press, Cambridge, MA, USA, 3104–3112.

[13] C. Tan, C. Bergmeir, F. Petitjean, and G. Webb, "Time Series Extrinsic Regression: Predicting numeric values from time series data" in Data Mining and Knowledge Discovery 2021

[14] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[15] Goodrum, Abby A. (2000). "Image Information Retrieval: An Overview of Current Research". Informing Science.

[16] Jansen, B. J. and Rieh, S. (2010) The Seventeen Theoretical Constructs of Information Searching and Information Retrieval Archived 2016-03-04 at the Wayback Machine. Journal of the American Society for Information Sciences and Technology. 61(8), 1517-1534.

[17] Mark Sanderson & W. Bruce Croft (2012). "The History of Information Retrieval Research". Proceedings of the IEEE. 100: 1444–1451. doi:10.1109/jproc.2012.2189916.

[18] Chechik, Gal, Varun Sharma, Uri Shalit, and Samy Bengio. "Large scale online learning of image similarity through ranking." (2010).

[19] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. J. Mach. Learn. Res. 7 (12/1/2006), 551–585.

[20] https://www.acc.org/latest-in-cardiology/articles/2017/11/08/11/47/mon-5pm-bp-guideline-aha-2017

[21] https://ieee-dataport.org/documents/auscultatory-and-oscillometry-blood-pressure-data#files