MS Thesis Defense: Extension of the Sequential K Nearest Neighbors Algorithm with Predictive Mean Matching (PMMSKNN)

Chong H. Kim

Committee Members:
Dr. Kathryn Colborn
Dr. Anna Baròn
Dr. Andrew Kittelson

2020

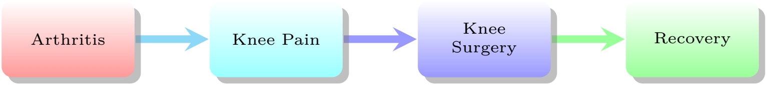
Overview

1. Background & Rationale
2. Specific Aims
3. Method
4. Results
5. Conclusion & Discussion

Overview

1. Background & Rationale
2. Specific Aims
3. Method
4. Results
5. Conclusion & Discussion

Total Knee Arthroplasty (TKA)

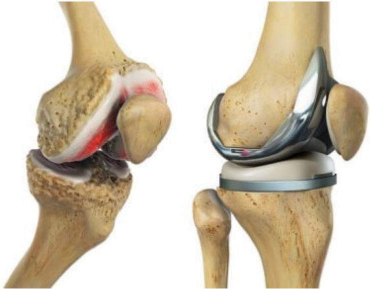


Arthritis

Knee Pain

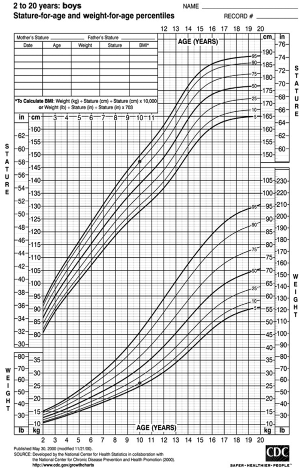
Knee Surgery

Recovery



Gap in Clinical Practice

- Consensus?
  - None! One-size-fits-all protocol
- Models?
  - Sort of! Not great
- Clinical Decision Support?
  - None! No reference charts



2 to 20 years: boys
Stature-for-age and weight-for-age percentiles

NAME ____________

RECORD # ________

Mother's Stature ________ Father's Stature ________

Date	Age	Weight	Stature	BMI*

*To Calculate BMI: Weight (kg) ÷ Stature (cm) ÷ Stature (cm) x 10,000
or Weight (lb) ÷ Stature (in) ÷ Stature (in) x 703

Published May 30, 2000 (modified 11/21/00).
SOURCE: Developed by the National Center for Health Statistics in collaboration with
the National Center for Chronic Disease Prevention and Health Promotion (2000).
http://www.cdc.gov/growthcharts

CDC
SAFER · HEALTHIER · PEOPLE™

Previous Efforts

- Sanchez-Santos et al. 2018:
  - General Linear Model, Multiple Imputation, Boostrap CV

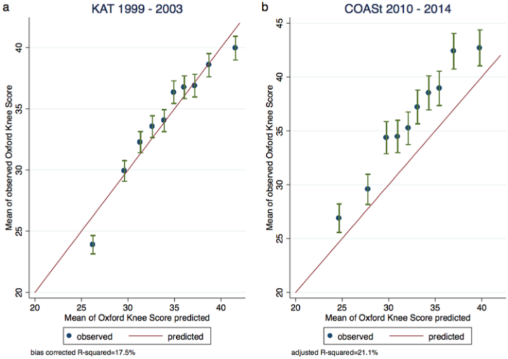
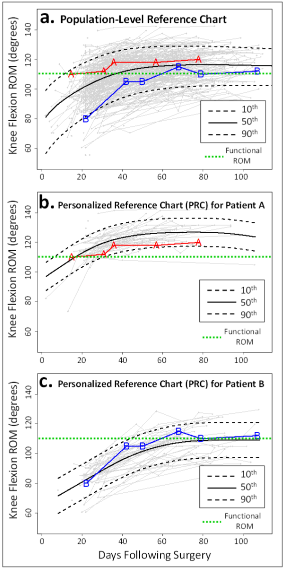


Figure 1. Calibration plot with R^2. (**a**) Calibration plot of the imputed development dataset (n = 1,649). (**b**) Calibration plot of the external dataset also imputed (n = 595). Internal and external samples used for validation were divided in 10 deciles, according to their predicted risk. For each decile, the mean predicted risk and the mean observed cases are shown on the X and Y axes, respectively. Bars indicate 95% Poisson confidence intervals.

Background & Rationale

- **Why?**: Personalized Medicine
  - **Health Outcomes**: Optimizing Health
  - **Economic Outcomes**: Optimizing Resource Allocation
- **What and How?**: Prediction using matching algorithms
  - **Sequential K Nearest Neighbors (SKNN)**: Unsupervised Learning
  - **Predictive Mean Matching (PMM)**: Supervised Learning

Better Information is Better Care



Overview

1. Background & Rationale
2. Specific Aims
3. Method
4. Results
5. Conclusion & Discussion

Specific Aims

Aim 1. Develop an algorithm for obtaining neighbors-based prediction by extending the Sequential K Nearest Neighbours algorithm with Predictive Mean Matching

Aim 2. Compare the performance of algorithms (i.e. LMM vs. SKNN vs. PMMSKNN) through simulation

Aim 3. Compare the performance of algorithms (i.e. LMM vs. SKNN vs. PMMSKNN) in case study

Aim 4. Develop software that implements the above algorithm

Overview

1 Background & Rationale

2 Specific Aims

3 Method
- Aim 1: Algorithms
- Aim 2: Simulation Study
- Aim 3: Case Study
- Aim 4: Software Development

4 Results

5 Conclusion & Discussion

Aim 1: SKNN Algorithm

Algorithm 1 SKNN Algorithm

```
procedure SKNN

    for m = 1 to M do                    ▷ let $M$ be the max No. matches specified via $d_k^*$
        for n = 1 to N do                ▷ let $N$ be the No. of subjects in data
            $\hat{\boldsymbol{y}}_n = GAMLSS$              ▷ Fit model to $n$-th person†
            Calculate $\boldsymbol{\lambda}$               ▷ $\boldsymbol{\lambda}$ can be bias, precision or coverage
        end for
        $s = \text{avg}(\boldsymbol{\lambda})$             ▷ Aggregate performance measure
    end for
    Return: $\hat{\boldsymbol{y}}_n$ with $\min s$ or $\max s$
end procedure
```

* d_k refers to euclidean (covariates) distance of k-th and n-th person

† GAMLSS model fit with $(Y_1 \dots Y_m, X_1 \dots X_m)$, $m = 1, \dots, M$

Aim 1: PMMSKNN Algorithm

Algorithm 2 PMMSKNN Algorithm

```
procedure PMMSKNN

    $\bar{\boldsymbol{y}} = \text{brokenstick}(\boldsymbol{x})$            ▷ Brokenstick prediction at time $t$
    $\tilde{\boldsymbol{y}} = f(x_1, x_2, x_3, \ldots, x_p)$    ▷ PMM using GLM; $x_i$'s are matching variables*
    for m = 1 to M do      ▷ let $M$ be the max No. matches specified via $d_k^{**}$
        for n = 1 to N do                ▷ let $N$ be the No. of subjects in data
            $\hat{\boldsymbol{y}}_n = GAMLSS$                        ▷ Fit model to $n$-th person†
            Calculate $\boldsymbol{\lambda}$             ▷ $\boldsymbol{\lambda}$ can be bias, precision or coverage
        end for
        $s = \text{avg}(\boldsymbol{\lambda})$                    ▷ Aggregate performance measure
    end for
    Return: $\hat{\boldsymbol{y}}_n$ with $\min s$ or $\max s$
end procedure
```

* The estimated $\tilde{\boldsymbol{y}}$ is based on a clinically relevant time point

** $d_k = \tilde{y}_k - \tilde{y}_n$ (difference in the predicted means for k-th and n-th person)

† GAMLSS model fit with $(Y_1 \ldots Y_m, X_1 \ldots X_m)$, $m = 1, \ldots, M$

PMMSKNN Tabular Description

Steps	Description	Details
1	Temporal Train-Test Split	Splitting of training and testing data for validation
2	Brokenstick prediction of training set outcome	Fitting of model to predict clinically relevant outcome (e.g. 90 day post-operative TUG)
3	Predictive Mean Matching: Generalized Linear Model	Fitting of model based on variables that contribute significantly to the (brokenstick) predicted outcome
4	Predictive Mean Matching: Match nearest m patients	Matching of patients based on the fitted value generated from generalized linear model
5	GAMLSS fitting of using m matched patients and evaluation of predictions	Optimize number of matches (m) based on performance measures
6	Test set Validation	Test set assessment of bias, coverage, and precision

Step 1: Temporal Train-Test Split

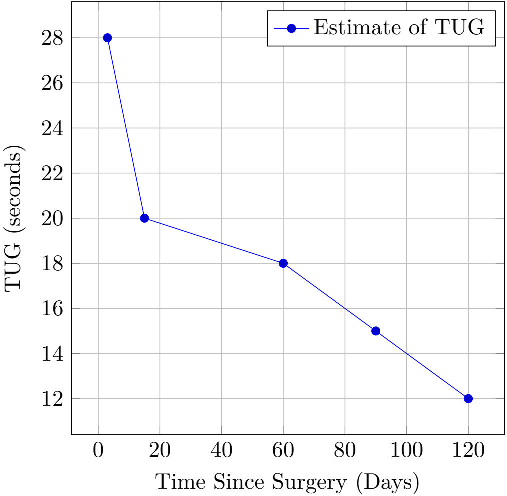
Train Data

ID	Age	Time (since surg)	TUG
1	43	-2	20
1	43	3	18
1	43	7	14
2	56	-7	34
2	56	5	20
2	56	7	17
⋮	⋮	⋮	⋮

Test Data

ID	Age	Time (since surg)	TUG
401	44	-2	21
401	44	3	18.5
401	44	7	14.5
402	55	-7	36
402	55	5	21
402	55	7	18.3

Step 2: Brokenstick



Brokenstick: Model Formulation

Follow notation of Fitzmaurice, Laird and Ware (2011)

$$Y_{ij} = \begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in_i} \end{pmatrix}, X_{ijk} = \begin{pmatrix} X_{i11} & X_{i12} & \cdots & X_{i1p} \\ X_{i21} & X_{i22} & \cdots & X_{i2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{in_i1} & X_{in_i2} & \cdots & X_{in_ip} \end{pmatrix}, \quad i = 1, \cdots, N$$

Time points t_{ij} transformed into knots x_{is} where $s = 1, \cdots, k+1$ using 2^{nd} order B-spline (De Boor 1978, 32)

$$H_s(t) = \begin{cases} (t - \kappa_{s-1})/(\kappa_s - \kappa_{s-1}) & \kappa_{s-1} < t \leq \kappa_s, \\ (\kappa_{s+1} - t)/(\kappa_{s+1} - \kappa_s) & \kappa_s \leq t < \kappa_{s+1}, \\ 0 & \text{otherwise}, \end{cases}$$

$$X_i = (x_{i1}, \cdots, x_{ik}, x_{i(k+1)})$$

$$Y_i = X_i\beta + X_ib + \epsilon_i$$

Brokenstick: Knot Specification

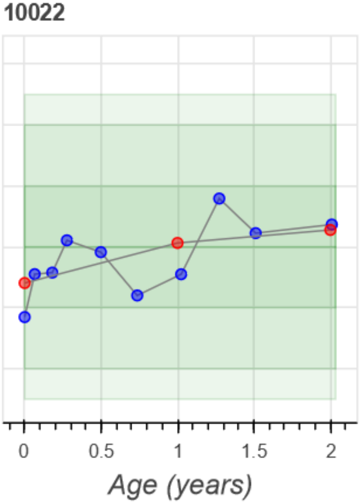


Figure: 3 knots

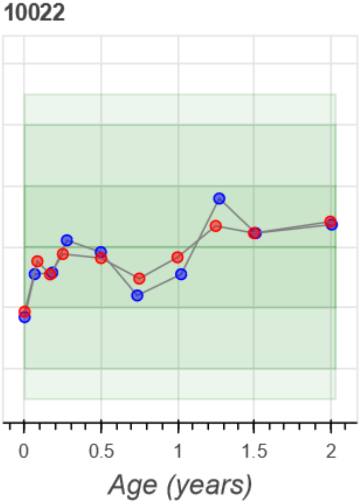


Figure: 10 knots

Step 3: PMM - Linear Model Fitting

$$\tilde{y} = \hat{\beta}_0 + \hat{\beta}_1 \text{Age} + \hat{\beta}_2 \text{Gender} + \hat{\beta}_3 \text{BMI} + \hat{\beta}_4 \text{bTUG}$$

ID	Age	$\tilde{y}^\dagger$
2	56	10
13	56	9.34
24	53	9.23
55	54	9.12
66	56	9.10
17	57	9.08
58	58	9.04
39	55	9.01
⋮	⋮	⋮

† $\tilde{y}$ may be predicted TUG at day 90

Step 4: PMM - Nearest m Matches

Matches for Patient 2

ID	Age	Time (since surg)	TUG
13	56	3	23
13	56	6	20
13	56	8	19
13	53	10	18
24	53	3	24
24	53	6	18
24	53	10	14
⋮	⋮	⋮	⋮

Matches for Patient 13

ID	Age	Time (since surg)	TUG
2	56	5	26
2	56	7	23
24	53	3	24
24	53	6	18
24	53	10	14
⋮	⋮	⋮	⋮

Step 5: GAMLSS Fitting

Within each of the N folds of the CV (in LOOCV), fit a GAMLSS model to the data (using m neighbors' data). (For the case study [with TUG outcome] we specified a Box-Cox-Cole-Green distribution with cubic spline smoother with $df = 3$ for location parameter and $df = 1$ for scale and shape parameters.)

LOOCV is used here to determine the optimal m matches with regard to bias, precision, and coverage.

Step 5: LOOCV in Training set

ID	Age	Time (since surg)	Observed TUG	Predicted TUG
2	56	5	20	18.5
2	56	7	17	16.5
⋮	⋮	⋮	⋮	⋮

Each subject has a prediction based on the model fit by nearest m matches

m	Bias	Coverage	Precision	Dropped Cases
10	0.30	0.20	0.5	5
15	0.23	0.10	0.6	3
20	0.19	0.08	0.7	1
⋮	⋮	⋮	⋮	⋮

Step 6: Test Set Validation

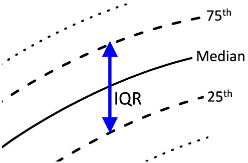
Train ID	Test ID	Time (since surg)	Observed TUG	Predicted TUG
2	401	3	18	19.5
2	401	7	14	16.5
⋮	⋮	⋮	⋮	⋮

ID 401 in Test matched to ID 2 in Train therefore predicted TUG based on the model for ID 2

Performance Measures

① Precision

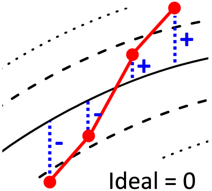
The width of the interquartile range (IQR)



Narrower is better

② Bias

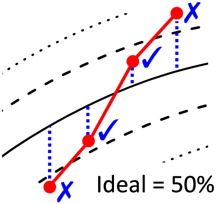
The average distance between projected and observed values



Ideal = 0

③ Coverage

The % of time the observed value is within the IQR



Ideal = 50%

Note: These 3 metrics combine to give a complete picture of PRC performance. This is preferable to a single summary metric such as Mean Squared Error (MSE).

Aim 2: Simulation Study

1. Compare the performance of algorithms (LMM, SKNN, & PMMSKNN) when increasing random noise for LMM generated y_{ij}

$$\begin{aligned} y_{ij} &= \beta_0 + b_{0i} + \beta_1 \text{time} + \beta_2 \text{age}_i + \beta_3 \text{bmi}_i + \beta_4 \text{BaselineTUG}_i + \epsilon_{ij} \\ e_{ij} &\sim N(0, \sigma) \\ \sigma &= 1 \end{aligned}$$

2. Compare the performance of algorithms (LMM, SKNN, & PMMSKNN) when increasing random noise for BCCG generated y_{ij}

$$\begin{aligned} f(y|\mu, \sigma, \nu) &= \frac{1}{\sqrt{2\pi}\sigma} \frac{y^{\nu-1}}{\mu^{\nu}} \exp(-\frac{z^2}{2}) \\ \mu_{ij} &= (\beta_0 + b_{0i} + \beta_1 \text{time} + \beta_2 \text{age}_i + \beta_3 \text{bmi}_i + \beta_4 \text{BaselineTUG}_i + \epsilon_{ij}) \times 1.2^{\beta_1 \text{time}} \\ \sigma &= 0.3 \\ \nu_{ij} &= (\beta_0 + b_{0i} + \beta_1 \text{time} + \beta_2 \text{age}_i + \beta_3 \text{bmi}_i + \beta_4 \text{BaselineTUG}_i + \epsilon_{ij}) \times 1.2^{\beta_1 \text{time}} \end{aligned}$$

Simulation 1

$$
\begin{aligned}
y_{ij} &= \beta_0 + b_{0i} + \beta_1 \text{time}_{ij} + \beta_2 \text{age}_i + \beta_3 \text{bmi}_i + \beta_4 \text{BaselineTUG}_i + \epsilon_{ij} \\
e_{ij} &\sim N(0, \sigma) \\
\sigma &= 1
\end{aligned}
$$

- y_{ij} generated from a random-intercept LMM model
- σ value varied from 2, 5, 10
- m-matches selected based on LOOCV (for SKNN & PMMSKNN) prioritizing in order of coverage, bias, then precision
- Choice of m: $m \in [10, 13, ..., 40]$

Simulation 2

$$
\begin{aligned}
f(y|\mu,\sigma,\nu) &= \frac{1}{\sqrt{2\pi}\sigma}\frac{y^{\nu-1}}{\mu^{\nu}}\exp(-\frac{z^2}{2}) \\
\mu_{ij} &= (\beta_0 + b_{0i} + \beta_1\text{time}_{ij} + \beta_2\text{age}_i + \beta_3\text{bmi}_i + \beta_4\text{BaselineTUG}_i + \epsilon_{ij}) \times 1.2^{\beta_1\text{time}_{ij}} \\
\sigma &= 0.3 \\
\nu_{ij} &= (\beta_0 + b_{0i} + \beta_1\text{time}_{ij} + \beta_2\text{age}_i + \beta_3\text{bmi}_i + \beta_4\text{BaselineTUG}_i + \epsilon_{ij}) \times 1.2^{\beta_1\text{time}_{ij}}
\end{aligned}
$$

- y_{ij} generated from a flexible GAMLSS-model with Box-Cox-Cole-Green distribution
- σ value varied from 0.3, 0.75, 1.5
- m-matches selected based on LOOCV (for SKNN & PMMSKNN) prioritizing in order of coverage, bias, then precision
- Choice of m: $m \in [10, 13, ..., 40]$

Aim 3: Case Study - Prediction of TUG trajectory

1. Data Source
   - Data from routine clinical practice & previously published longitudinal studies (surgery dates: Jan ‘13 - June ‘17)
   - Patients with pre- & post-operative Timed Up & Go were utilized
2. Measurement
   - **Primary outcome:** Timed Up & Go (TUG); a brief test of mobility
   - **Explanatory variables:** Age, gender, BMI, and TUG at time of surgery
3. Assess Performance
   - RMSE, 50% Coverage, & Precision (width of IQR)

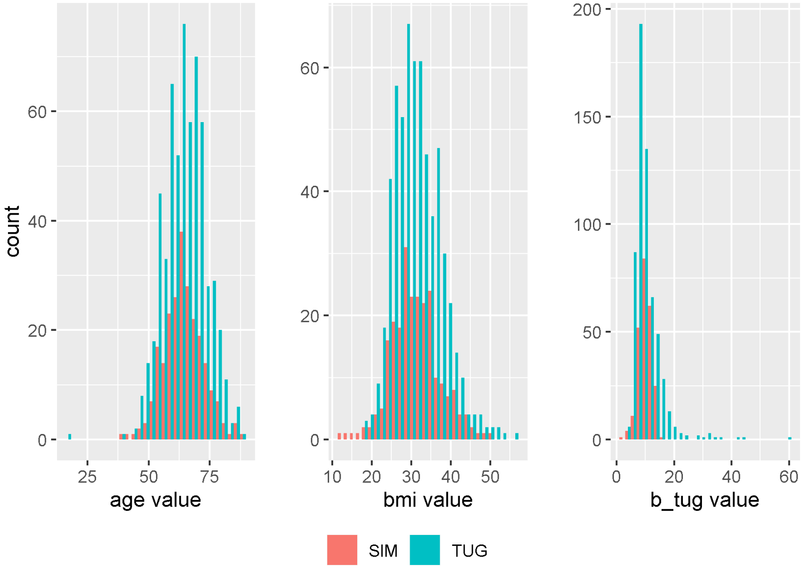
Aim 4: Software Development

- R Package - In Active Development
- Contains Procedures to fit and plot (the performance of) SKNN and PMMSKNN (via LOOCV)

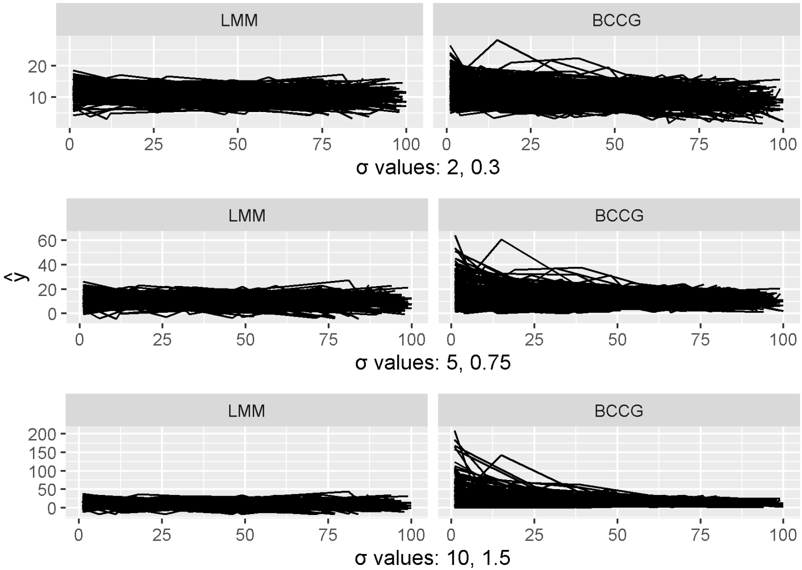
Overview

1. Background & Rationale
2. Specific Aims
3. Method
4. Results
   - Aim 2: Simulation
   - Aim 3: Case Study
   - Aim 4: Software Development
5. Conclusion & Discussion

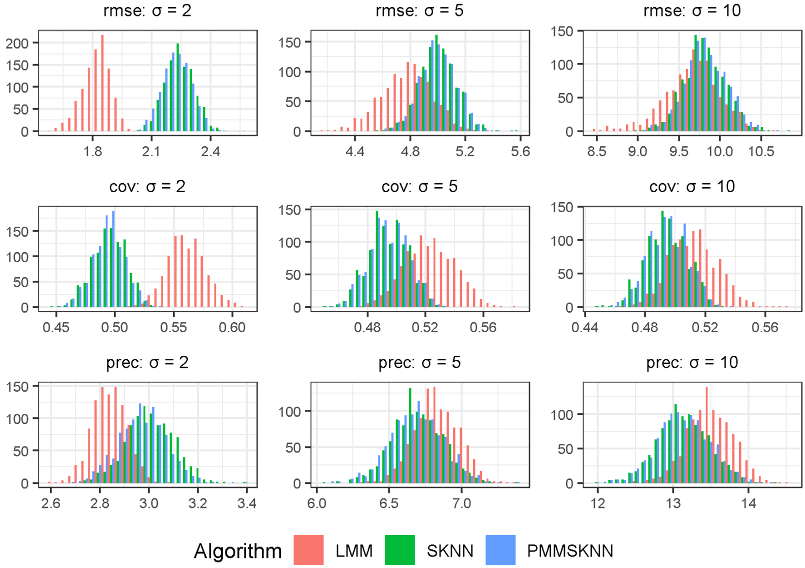
Baseline Covariates for Simulation 1 & 2



Aim 2. $\hat{y}$ over time for LMM & BCCG



Simulation 1 Results - Graphical



Simulation 1 Results - Tabular

Table: Performance Results for Simulation Study 1. Performance of LMM, SKNN, & PMMSKNN over 1,000 simulations for $\sigma = 2, 5, 10$ using $\hat{y}$ generated from LMM.

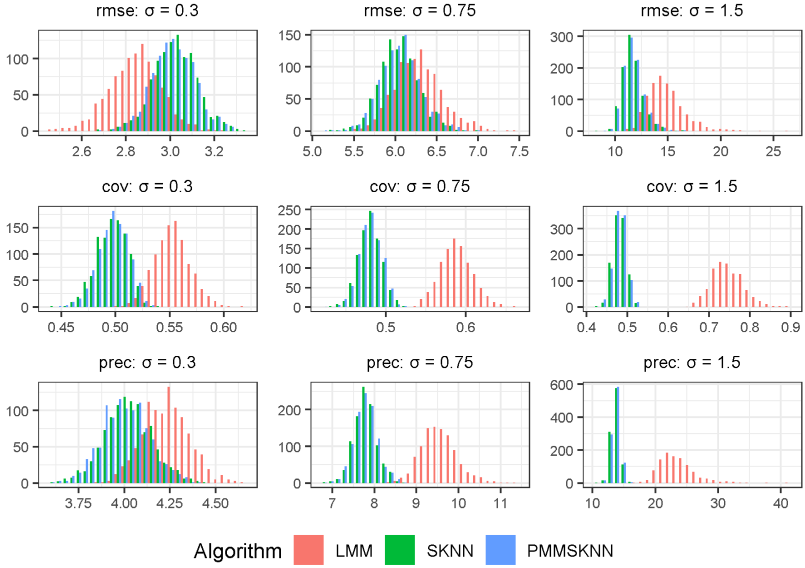
Performance	σ	LMM	SKNN	PMMSKNN
RMSE				
	2	1.836 (1.674-1.95)	2.236 (2.099-2.382)	2.22 (2.079-2.362)
	5	4.799 (4.413-5.117)	4.994 (4.747-5.254)	4.986 (4.728-5.247)
	10	9.704 (8.896-10.283)	9.797 (9.33-10.274)	9.791 (9.317-10.306)
50% Coverage				
	2	0.561 (0.532-0.595)	0.494 (0.466-0.52)	0.493 (0.466-0.519)
	5	0.524 (0.491-0.557)	0.494 (0.467-0.519)	0.493 (0.467-0.52)
	10	0.513 (0.483-0.546)	0.494 (0.468-0.519)	0.494 (0.467-0.519)
Precision[*]				
	2	2.845 (2.706-2.981)	3 (2.806-3.194)	2.963 (2.769-3.156)
	5	6.838 (6.542-7.135)	6.684 (6.34-7.038)	6.675 (6.313-7.035)
	10	13.509 (12.926-14.075)	13.118 (12.435-13.799)	13.114 (12.451-13.802)

Note:

Performance values are represented as: mean (95% CI)

[*] Width of the 50% prediction interval for LMM; Width of the Inter Quartile Range for SKNN and PMMSKNN

Simulation 2 Results - Graphical



Simulation 2 Results - Tabular

Table: Performance Results for Simulation Study 2. Performance of LMM, SKNN, & PMMSKNN over 1,000 simulations for $\sigma = 0.3, 0.75, 1.50$ using $\hat{y}$ generated from LMM.

Performance	σ	LMM	SKNN	PMMSKNN
RMSE				
	0.30	2.855 (2.616-3.075)	3.022 (2.817-3.225)	3.006 (2.822-3.21)
	0.75	6.294 (5.785-6.955)	6.033 (5.597-6.539)	6.041 (5.596-6.56)
	1.50	14.764 (12.467-19.003)	11.485 (10.016-13.95)	11.503 (10.026-13.878)
50% Coverage				
	0.30	0.553 (0.523-0.589)	0.496 (0.466-0.519)	0.497 (0.467-0.52)
	0.75	0.59 (0.552-0.633)	0.48 (0.452-0.509)	0.48 (0.452-0.505)
	1.50	0.747 (0.681-0.836)	0.479 (0.449-0.51)	0.479 (0.446-0.509)
Precision*				
	0.30	4.241 (3.994-4.487)	4.016 (3.75-4.29)	3.993 (3.753-4.266)
	0.75	9.549 (8.793-10.533)	7.753 (7.244-8.325)	7.766 (7.244-8.337)
	1.50	23.199 (19.545-30.475)	13.509 (12.329-14.886)	13.519 (12.377-14.893)

Note:

Performance values are represented as: mean (95% CI)

* Width of the 50% prediction interval for LMM; Width of the Inter Quartile Range for SKNN and PMMSKNN

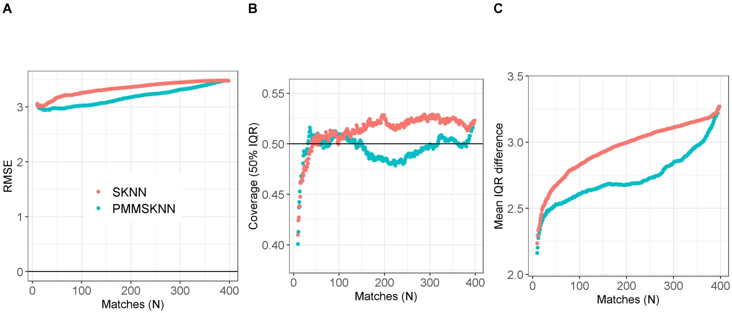
Aim 3: Case Study Patient Characteristics

Table: Baseline Characteristics of Training and Testing Set

	Train (N = 398, # TUG Obs = 1325)	Test (N = 201, # TUG Obs = 602)	p^a
Age (years) (mean (SD))	64.04 (8.43)	65.90 (8.84)	0.012
Gender = Male (%)	185 (46.6)	84 (41.6)	0.280
BMI (kg/m^2) (mean (SD))	31.33 (5.82)	31.98 (6.20)	0.208
Baseline TUG (sec) (mean (SD))	9.98 (4.95)	11.00 (5.04)	0.018

[a] Continuous variables tested with one-way analysis of variance; Categorical variables tested with χ^2 test

Aim 3: LOOCV Result



Aim 3: LMM Calibration

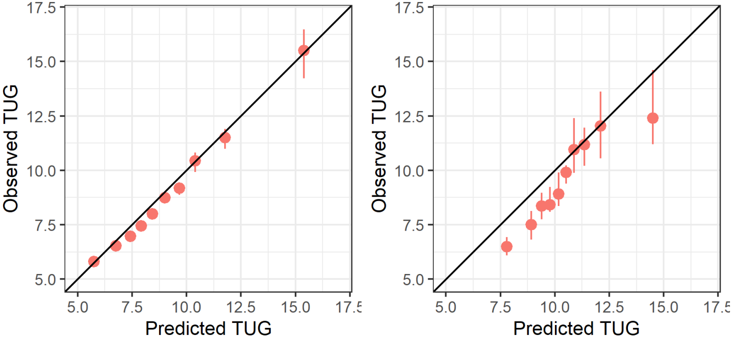


Figure: Decile-decile calibration plot for LMM predictions. **Left**: Training data. **Right**: Testing data.

Aim 3: SKNN Calibration

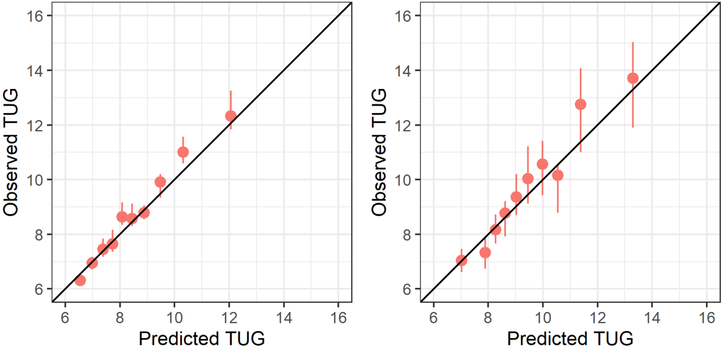


Figure: Decile-decile calibration plot for SKNN predictions. **Left**: Training data. **Right**: Testing data.

Aim 3: PMMSKNN-GAMLSS Calibration

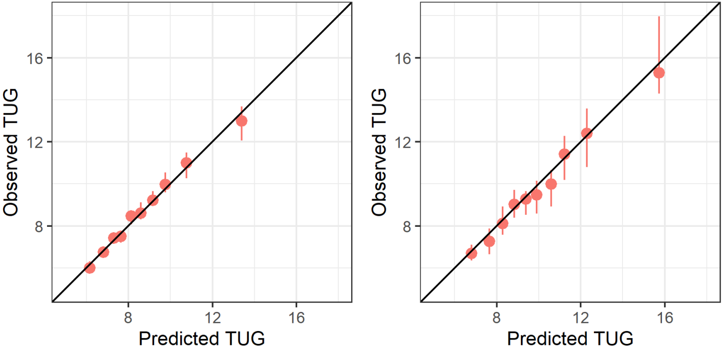


Figure: Decile-decile calibration plot for PMMSKNN-GAMLSS predictions. **Left**: Training data. **Right**: Testing data.

Aim 3: Summary of Performance

Table: Overall Performance for LMM vs. SKNN vs. PMMSKNN

	LMM	SKNN	PMMSKNN
Train			
RMSE	2.032	3.115	2.952
50% Coverage	0.78	0.50	0.50
Precision*	3.426	2.644	2.484
Test			
RMSE	6.254	4.621	4.327
50% Coverage	0.404	0.455	0.430
Precision*	3.119	3.171	3.160

Note:

Performance values are represented as: mean

* Width of the 50% prediction interval for LMM; Width of the Inter Quartile Range for SKNN and PMMSKNN

Aim 3: Trajectory Visualization

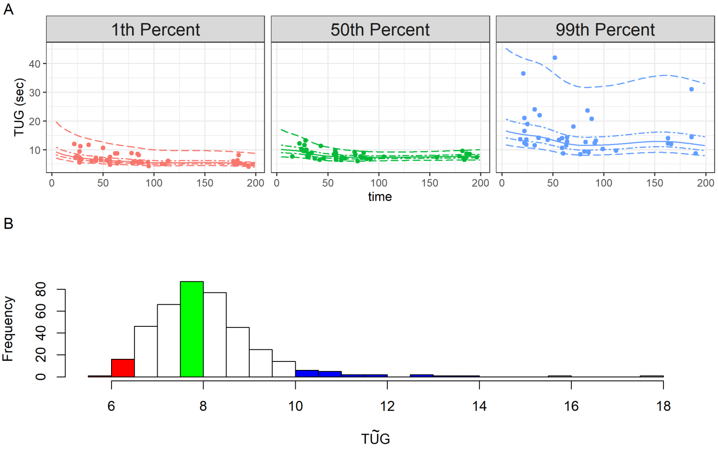
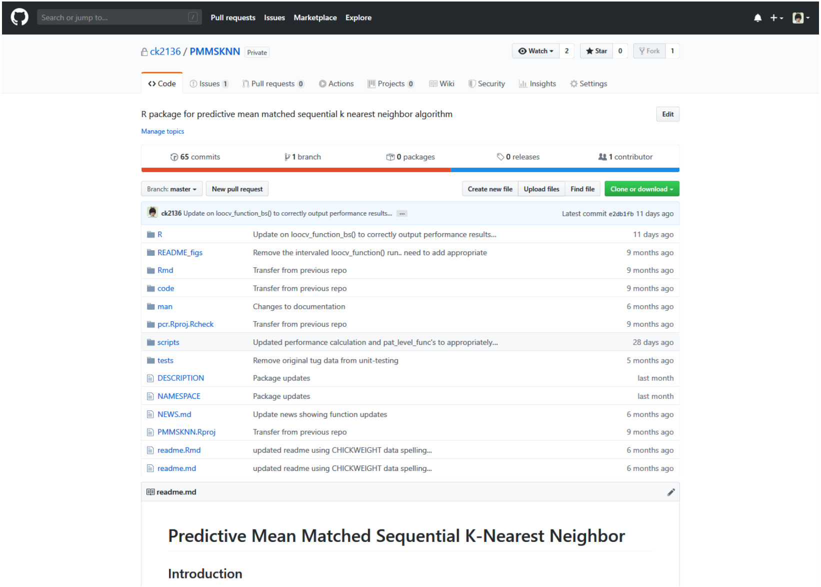


Figure: Personalized TÛG trajectory based on nearest 15 matches using PMMSKNN-GAMLSS. **A**: 1$^{\text{st}}$, 50$^{\text{th}}$ and 99$^{\text{th}}$ percentile person's predictions. **B**: TŨG values of 15 matched observations for each of the individuals in plot **A**.

Aim 4: R package PMMSKNN



R package for predictive mean matched sequential k nearest neighbor algorithm

ck2136 Update on loocv_function_bs() to correctly output performance results... … Latest commit e2db1fb 11 days ago

R	Update on loocv_function_bs() to correctly output performance results...	11 days ago
README_figs	Remove the intervaled loocv_function() run.. need to add appropriate	9 months ago
Rmd	Transfer from previous repo	9 months ago
code	Transfer from previous repo	9 months ago
man	Changes to documentation	6 months ago
pcr.Rproj.Rcheck	Transfer from previous repo	9 months ago
scripts	Updated performance calculation and pat_level_func's to appropriately...	28 days ago
tests	Remove original tug data from unit-testing	5 months ago
DESCRIPTION	Package updates	last month
NAMESPACE	Package updates	last month
NEWS.md	Update news showing function updates	6 months ago
PMMSKNN.Rproj	Transfer from previous repo	9 months ago
readme.Rmd	updated readme using CHICKWEIGHT data spelling...	6 months ago
readme.md	updated readme using CHICKWEIGHT data spelling...	6 months ago

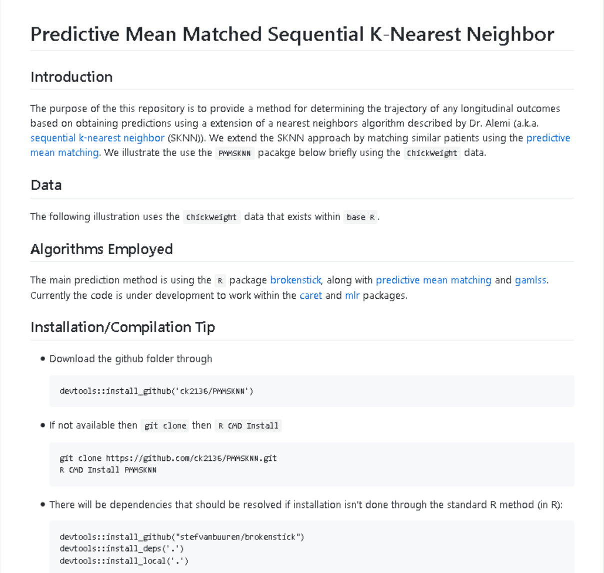
readme.md

Predictive Mean Matched Sequential K-Nearest Neighbor

Introduction

Figure: PMMSKNN Package on Private Github repository

Aim 4: R package PMMSKNN



Predictive Mean Matched Sequential K-Nearest Neighbor

Introduction

The purpose of the this repository is to provide a method for determining the trajectory of any longitudinal outcomes based on obtaining predictions using a extension of a nearest neighbors algorithm described by Dr. Alemi (a.k.a. sequential k-nearest neighbor (SKNN)). We extend the SKNN approach by matching similar patients using the predictive mean matching. We illustrate the use the `PMMSKNN` pacakge below briefly using the `ChickWeight` data.

Data

The following illustration uses the `ChickWeight` data that exists within `base R`.

Algorithms Employed

The main prediction method is using the `R` package brokenstick, along with predictive mean matching and gamlss. Currently the code is under development to work within the caret and mlr packages.

Installation/Compilation Tip

- Download the github folder through

```
devtools::install_github('ck2136/PMMSKNN')
```

- If not available then `git clone` then `R CMD Install`

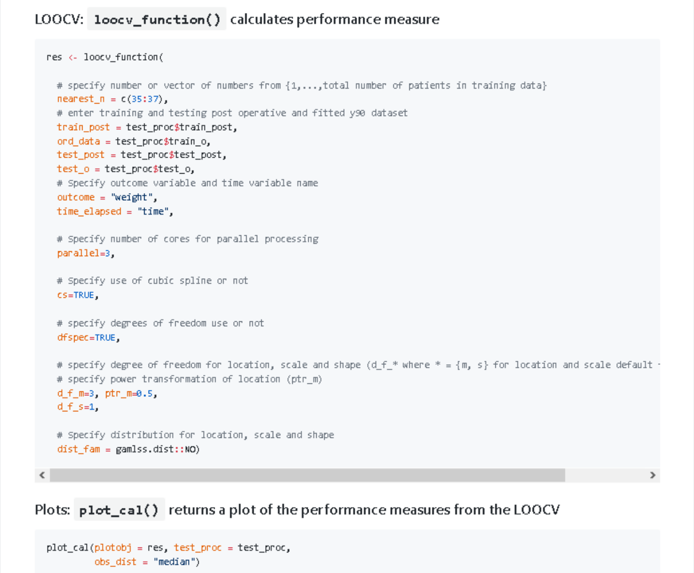
```
git clone https://github.com/ck2136/PMMSKNN.git
R CMD Install PMMSKNN
```

- There will be dependencies that should be resolved if installation isn't done through the standard R method (in R):

```
devtools::install_github("stefvanbuuren/brokenstick")
devtools::install_deps('.')
devtools::install_local('.')
```

Figure: PMMSKNN Package on Private Github repository: Readme

Aim 4: R package PMMSKNN



LOOCV: `loocv_function()` calculates performance measure

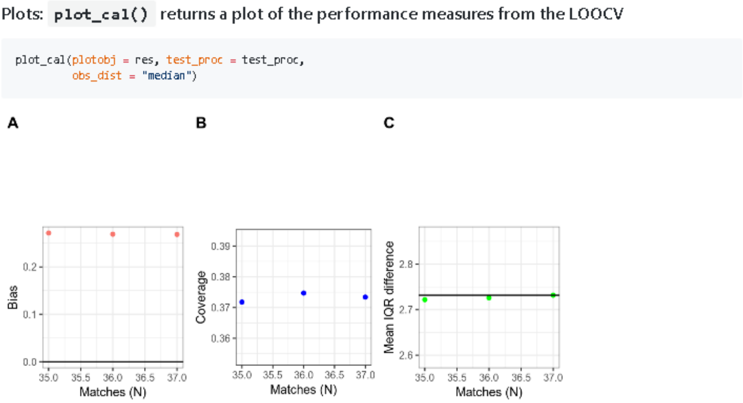
```
res <- loocv_function(

  # specify number or vector of numbers from {1,...,total number of patients in training data}
  nearest_n = c(35:37),
  # enter training and testing post operative and fitted y90 dataset
  train_post = test_proc$train_post,
  ord_data = test_proc$train_o,
  test_post = test_proc$test_post,
  test_o = test_proc$test_o,
  # Specify outcome variable and time variable name
  outcome = "weight",
  time_elapsed = "time",

  # Specify number of cores for parallel processing
  parallel=3,

  # Specify use of cubic spline or not
  cs=TRUE,

  # specify degrees of freedom use or not
  dfspec=TRUE,

  # specify degree of freedom for location, scale and shape (d_f_* where * = {m, s} for location and scale default
  # specify power transformation of location (ptr_m)
  d_f_m=3, ptr_m=0.5,
  d_f_s=1,

  # Specify distribution for location, scale and shape
  dist_fam = gamlss.dist::NO)
```

Plots: `plot_cal()` returns a plot of the performance measures from the LOOCV

```
plot_cal(plotobj = res, test_proc = test_proc,
         obs_dist = "median")
```

Figure: PMMSKNN Package on Private Github repository: Functions Description

Aim 4: R package PMMSKNN



Plots: `plot_cal()` returns a plot of the performance measures from the LOOCV

```
plot_cal(plotobj = res, test_proc = test_proc,
         obs_dist = "median")
```

Figure: PMMSKNN Package on Private Github repository: LOOCV Result Using ChickWeight Data

Overview

1. Background & Rationale
2. Specific Aims
3. Method
4. Results
5. Conclusion & Discussion
    - Discussion of Results
    - Questions & Acknowledgement

Simulation Results

- With $\hat{y}$ simulated from LMM:
  - ∗ The difference in the bias (for SKNN & PMMSKNN vs. LMM) becomes smaller as σ increases.
  - ∗ The coverage for SKNN & PMMSKNN vs. LMM is consistent even as σ increases.
  - ∗ The precision of SKNN & PMMSKNN predictions outperforms that of the LMM predictions as σ increases.
- With $\hat{y}$ simulated from GAMLSS-BCCG:
  - ∗ The bias for SKNN & PMMSKNN vs. LMM is smaller as σ increases.
  - ∗ The coverage of SKNN & PMMSKNN outperforms LMM as σ increases.
  - ∗ The precision of SKNN & PMMSKNN predictions outperforms that of the LMM predictions as σ increases.

Case Study Results

- LMM has the lowest bias in the training set but highest bias in the test set.
- PMMSKNN & SKNN both exhibit similar performance in the training and test set. Specifically:
    - ∗ In terms of the bias, PMMSKNN outperformes LMM & SKNN.
    - ∗ In terms of the coverage, SKNN outperforms both PMMSKNN-GAMLSS & LMM.
    - ∗ In terms of the precision, all three algorithms have similar performance.

Discussion

- When the distribution of outcome over time doesn't follow a simple LMM, SKNN & PMMSKNN should be considered as it may provide better predictions.
- The SKNN & PMMSKNN approach employs a flexible matching strategy that allows users to specify the characteristics of patients to match on, the time point of interest to match patients, and the distribution to fit the outcome of interest over time.
- This flexibility introduces subjectivity and therefore warrants clear documentation of procedure utilized during analysis.

Discussion

- Differences in the SKNN & PMMSKNN approach is minimal if the variation in the matching characteristics in the sample is small; If there are limited number of patients with different characteristics that have the same outcome value, predictive mean matching would match patients with similar characteristics therefore predictions from PMMSKNN would be similar to that of SKNN.
- Visualization of the trajectory of outcomes allows stakeholders to better understand the patients' current outcome status & become more engaged in the recovery process that may lead to efficient use of resources & faster recovery for patients.

Computational Note

- The SKNN & PMMSKNN approaches take much longer than the LMM if personalized estimates are obtained from N number of model fitting exercises via $m = 1, \ldots, N$, cross-validated hyperparameter ($m =$ number of matches) tuning exercise ($O(n^2)$).
- The SKNN & PMMSKNN approaches can be flexible in the model/distribution used to fit the observations from m matches. This can be optimized as a hyperparameter as well (at the cost of speed: the time complexity of the algorithm may approach $O(n^3)$ or beyond with the number of hyperparameters to optimize).

Next Steps

- Tweak the PMMSKNN package so that parts of the computation can run faster (i.e. profiling)
- Finish implementing the PMMSKNN-BS (BrokenStick)
- Improve the documentation & example section to enhance the ease of use for others
- Compare algorithms on other known longitudinal datasets (e.g. growth datasets)

Questions?

Thank You