Gaussian Process Models for Mortality Rates and Improvement Factors: An Interactive R Markdown Approach

Longevity 14

Jimmy Risk

Dept of Mathematics & Statistics Cal Poly Pomona

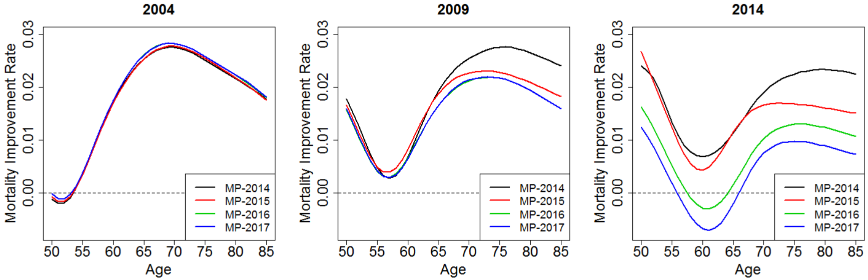
September 20 2018

Joint with M. Ludkovski (UCSB) and H. Zail (Elucidor)

US Mortality Improvement

MP-2014 / MP-2015

- Published by SOA, uses “RPEC_2014" model
- US CDC Data
- MP-2014 uses years 1950-2009
- Plans to update scales at least triennially; two years of additional CDC data shows drastic change in later years
  - MP-2015 emerges
  - (Also showing MP-2016, MP-2017, came after this paper finished)



Goal:

Model US Mortality data using Gaussian Process (GP) regression

- Bayesian
- Provides posterior Gaussian distribution for input of any age and year
- Offers easy analysis of both mortality and mortality improvement simultaneously
- Gaussian distribution implies one-year mortality improvement factors remains Gaussian
- Differentiable: can provide instantaneous mortality improvement (still Gaussian)
- Spatial approach inherently handles missing and edge data
- Provide simple to use code with output through an R notebook

Typical Regression Assumption

Hypothesis:

$$y = f(x) + \varepsilon$$

- Observe $\boldsymbol{y} = y^{1:N}$ for input locations $\boldsymbol{x} = x^{1:N}$
- Want to understand the function f
  - e.g. $f(x) = \beta_0 + \beta_1 x$ (simple linear regression)
- ε is noise
  - e.g. measurement error
  - can't observe $f(x)$ directly
- Assume $\varepsilon \sim \mathcal{N}\left(0, \sigma^2(x)\right)$ (often $\sigma(x) \equiv \sigma \in \mathbb{R}^+$)
- Often in mortality modeling: $f(x)$ is based on an ARIMA process(es) or on splines
- Our assumption: f is a Gaussian Process (modeling log-mortality, $x = (x_{ag}, x_{yr})$)

Gaussian Process

- Defined as a collection of random variables $\{f(x)|x \in \mathbb{R}^d\}$
- Any finite subset has a multivariate Gaussian distribution with covariance $C(\cdot, \cdot)$:

$$f(x_1), \ldots, f(x_n) \sim \mathcal{N}\left((m(x_1), \ldots, m(x_n)), C(\boldsymbol{x}, \boldsymbol{x}^T)\right).$$

- Fix mean function m and covariance kernel C; this provides a prior distribution

Modeling with Gaussian Processes

1. Declare prior mean function and covariance kernel
    - Mean function can also be parametric and fitted with data; useful in extrapolation
    - Covariance kernel governs spatial relation between points
    - Hyperparameters can be specified using expert knowledge or fitted from data

Modeling with Gaussian Processes

1. Declare prior mean function and covariance kernel
   - Mean function can also be parametric and fitted with data; useful in extrapolation
   - Covariance kernel governs spatial relation between points
   - Hyperparameters can be specified using expert knowledge or fitted from data
2. Output can be easily evaluated at any location
   - Output is a random variable with mean and covariance depending on neighboring inputs

Posterior

- Observe pairs $(\boldsymbol{y}, \boldsymbol{x}) = ((y, x)^{1:N})$
  - (e.g. $y =$ historic log–mortality and $x =$ (age, year))
- Gaussian assumptions imply that marginally for any input x

$$f(x)|(\boldsymbol{y}, \boldsymbol{x}) \sim \mathcal{N}\left(m_*(x), s_*^2(x)\right)$$

Posterior

- Observe pairs $(\boldsymbol{y}, \boldsymbol{x}) = ((y, x)^{1:N})$
  - (e.g. y = historic log–mortality and x = (age, year))
- Gaussian assumptions imply that marginally for any input x

$$f(x)|(\boldsymbol{y}, \boldsymbol{x}) \sim \mathcal{N}\left(m_*(x), s_*^2(x)\right)$$

- m_* and s_*^2 are the posterior mean and variance functions

$$\begin{cases} m_*(x) \doteq \boldsymbol{c}(x)^T(\boldsymbol{C} + \Sigma)^{-1}\boldsymbol{y}; \\ s_*^2(x) \doteq C(x,x) - \boldsymbol{c}(x)^T(\boldsymbol{C} + \Sigma)^{-1}\boldsymbol{c}(x), \end{cases} \tag{1}$$

where

$$\begin{cases} \boldsymbol{c}(x) \doteq \left(C(x, x^i)\right)_{1\le i\le N} & \text{(covariances between } x \text{ and inputs } \boldsymbol{x}) \\ \boldsymbol{C} \doteq \left(C(x^i, x^j)\right)_{1\le i,j\le N} & \text{(covariances between inputs } \boldsymbol{x}) \\ \Sigma \doteq \operatorname{diag}\left(\sigma^2(x^i)\right) & \text{(diagonal matrix of noise variance)} \end{cases}$$

Covariance Kernels & Parameter Estimation

- Common choice is squared-exponential (or Gaussian) covariance kernel

$$C(x, x') = \eta^2 \exp\left(-\frac{(x_{ag} - x'_{ag})^2}{2\theta_{ag}^2} - \frac{(x_{yr} - x'_{yr})^2}{2\theta_{yr}^2} \right).$$

- Knowing mortality at x will greatly influence mortality at “neighboring" x’s
  - e.g. knowing mortality for a 80 year old in 2015 greatly aids in prediction of a 85 year old’s mortality in 2016; knowing a 50 year old’s mortality in 2000 has a nearly non-existent effect
- Implies hyperparameter family of $\Theta \doteq (\theta_{ag}, \theta_{yr}, \eta^2, \sigma^2)$
  - Also mean function hyperparameters (if included)
- Estimates are fit using MLE; likelihood can be written out explicitly due to Gaussian assumptions
  - Done using `R` package `DiceKriging`
- Alternatively, can use Bayesian approach with priors on Θ
  - Separate package using STAN language
  - Leads to non-Gaussian posterior

Illustrative Example

Goal: Learn $f(x) = \sin(x)$ over domain $[0, 1.5\pi]$

- Observe realizations of

$$y = \sin(x) + \varepsilon$$

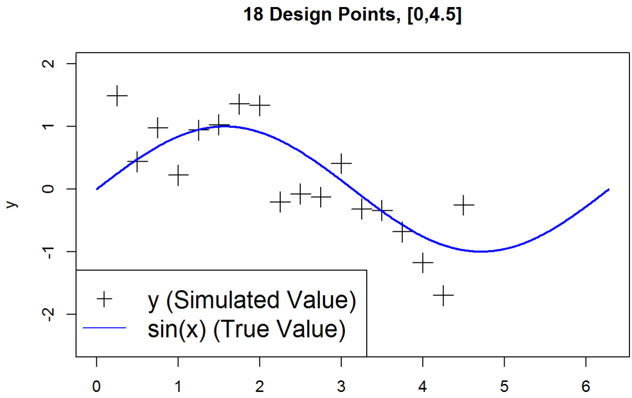
  where $\varepsilon \sim N(0, \sigma = 0.5)$
- Try:
  - $x = 0.25, 0.5, 0.75, \ldots, 2.75, 4.5$ ($N = 18$)
- Then update model with data on $(1.5\pi, 2\pi]$ to see how the overall fit changes
  - add $x = 4.75, \ldots, 6$ (total $N = 24$)

Illustrative Example

$$y = \sin(x) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.01x)$$

1. Generate data from random process

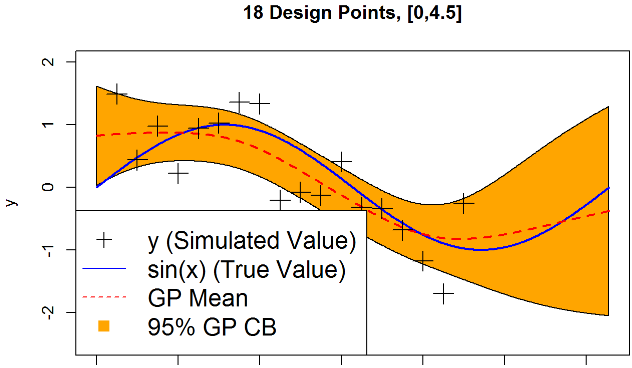
$$y = \sin(x) + \varepsilon, \qquad e \sim N(0, 0.5)$$



Illustrative Example

Fit GP to $N = 18$ (x, y) pairs

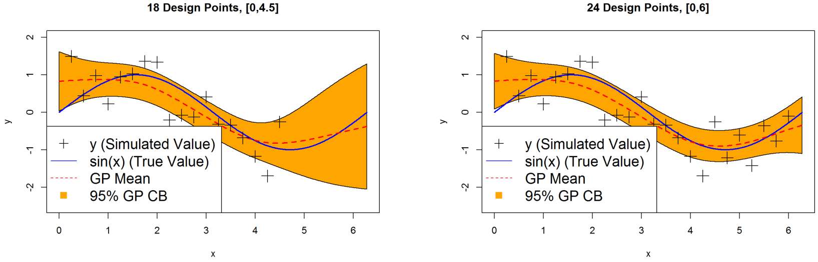
- Estimate hyperparameters (θ, η, σ)
- Produce posterior mean, covariance matrix (provides credible intervals)
- Observe naturally increasing uncertainty at edge data



Illustrative Example

Update GP with $N_{new} = 6$ (x, y) pairs on $[4.75, 6]$

- (Optional) update hyperparameters (θ, η, σ)
- Produce posterior mean, covariance matrix (provides credible intervals)



R Notebook Code

```
x <- seq(0.25,4.5,0.25)
n <- length(x)
y <- sin(x)+rnorm(n,0,0.5)
```

```
library(DiceKriging)
fit_nug <- km(formula = ~1,
              design = data.frame(x=x), response = y,
              nugget.estim = TRUE,
              covtype= "gauss",
              optim.method="gen")
```

```
nug <- fit_nug@covariance@nugget

fit <- km(formula = ~1,
          design = fit_nug@X, fit_nug@y,
          noise.var = rep(nug, fit_nug@n),
          coef.trend = fit_nug@trend.coef, #re-use obtained hyperparameters
          coef.cov = fit_nug@covariance@range.val)
```

```
##
## Call:
## km(formula = ~1, design = fit_nug@X, response = fit_nug@y, coef.trend = fit_nug@trend.coef,
##     coef.cov = fit_nug@covariance@range.val, noise.var = rep(nug,
##         fit_nug@n))
##
## Trend  coeff.:
##
##  (Intercept)    -0.0346
##
## Covar. type  : matern5_2
## Covar. coeff.:
##
##     theta(x)     1.3373
##
## Variance: 0.5803238
```

Comments

- $m(x) = m$ assumed *(clearly not true)*
- In practice,
  - Data is usually detrended, or
  - Parametric trend function e.g. $f(x) = \beta_0 + \beta_1 x$ assumed (and fitted alongside)
- Example is one-dimensional ($x \in \mathbb{R}$)
  - Framework naturally extends to multi-dimensional case ($x \in \mathbb{R}^d$), for example
    - $f(x, y) = \sin(x)\cos(y) + 2xy$
    - $f(\text{age}, \text{year}) = (\text{mortality rate depending on age, year})$

R Notebook For GPs on Mortality Data

- Provide R “code blocks" along with explanation of what it does and discussion of results
- Practitioner can choose to modify as much as needed
  - For example, simply change `cdcMale.csv` to `myInsuranceCompanyData.csv`
  - Plots have changeable ranges (easy to choose what years to plot)
  - ALL code is available, so programmers can easily modify as needed

Data

CDC Data

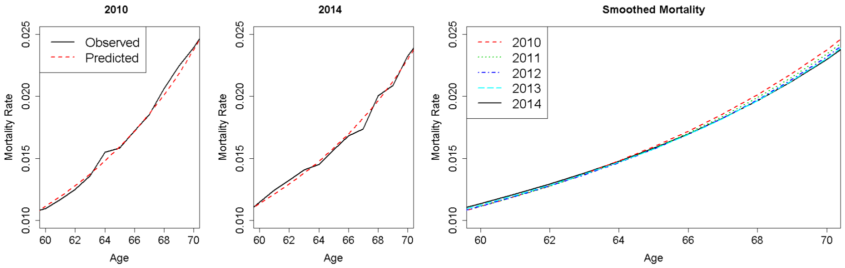
- United States
- Ages 50–84, Years 1999–2014
  - 1360 Data Points ($x = (x_{ag}, x_{yr})$)
  - 84 is maximal age for CDC data
  - 50 chosen as cutoff to minimize mixing lower age behavior
  - 1999 earliest year available on wonder.cdc.gov
  - Could add earlier years, but our analysis suggests they have little effect
  - Most relevant for longevity risk

GP Model Assumptions

- Observe central mortality rate $e^{-\mu(x_{ag},x_{yr})} = D(x_{ag,yr})/E(x_{ag,yr})$
- Fit log-mortality rate y to $x = (x_{ag}, x_{yr})$ pairs
- Can try $\sigma(x)$ based on Binomial assumption
  - Overdispersion issues ($\mu_{ag,yr}$ is unknown)
  - Minimal change in final model from simply choosing $\sigma := \sigma(x)$
- Use Gaussian covariance kernel
  - Implies f is differentiable
  - Minimal change in final model from other kernel choices
- Changed $m(x)$ trend function based on application
  - In-sample analysis generally used $m(x) = \beta_0$
  - Out-of-sample generally used $m(x) = \beta_0 + \beta_1 x_{ag} + \beta_2 x_{ag}^2 + \beta_3 x_{yr}$ (like Gompertz)

Posterior Predicted Mortality Rates

- Showing $m_*(x)$ for each ages 60–70
- Left panels include historic observations
- Right panel suggests mortality improvement



Goals

- In-sample smoothing
- Extrapolation (both in calendar year and age)
- Mortality Improvement

$$MI^{obs}_{back}\left(x_{ag}; yr\right) \doteq 1 - \frac{\exp\left(\mu(x_{ag}, yr)\right)}{\exp\left(\mu(x_{ag}, yr-1)\right)}$$

compare with SOA MP-2015 results

Loading Data

- Input data should be an R data frame with
  - age, calendar year, deaths, exposure
- The corresponding log mortality rates are computed as

$$y^n = \log(D^n/L^n)$$

  - D^n is the number of deaths and for the nth age/year pair $x^n = (x^n_{ag}, x^n_{yr})$,
  - L^n midyear count of lives

```
mortData <- read.csv("cdcMale.csv",header=T)
mortData$rate <- mortData$D / mortData$L
mortData$y <- log(mortData$rate)
head(mortData)
```

```
##   X age year     D       L        rate         y
## 1 1  50 1999  9775 1847555 0.005290776 -5.241790
## 2 2  51 1999 10470 1762492 0.005940452 -5.125970
## 3 3  52 1999 11509 1900702 0.006055131 -5.106849
## 4 4  53 1999  9885 1355175 0.007294261 -4.920667
## 5 5  54 1999 10717 1413117 0.007583944 -4.881722
## 6 6  55 1999 11728 1390616 0.008433673 -4.775523
```

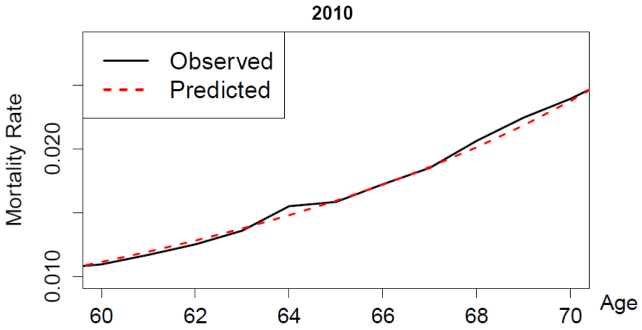
Fitting the Model

```
xMort <- data.frame(age = mortData$age, year = mortData$year)
yMort <- mortData$y
mortModel_nug <- km(formula = ~1,
                    design = data.frame(x = xMort), response = yMort,
                    nugget.estim=TRUE,
                    covtype="gauss",
                    optim.method="gen",
                    # the "control" parameters below handle speed versus risk of
                    # converging to local minima.  See "rgenoud" package for details
                    control=list(max.generations=100,pop.size=100,wait.generations=8,
                                 solution.tolerance=1e-5))
mortModel_nug
```

```
##
## Call:
## km(formula = ~1, design = data.frame(x = xMort), response = yMort,
##     covtype = "gauss", nugget.estim = TRUE, optim.method = "gen",
##     control = list(max.generations = 100, pop.size = 100, wait.generations = 8,
##         solution.tolerance = 1e-05))
##
## Trend  coeff.:
##                Estimate
##  (Intercept)    -3.8710
##
## Covar. type  : gauss
## Covar. coeff.:
##                 Estimate
##   theta(x.age)    15.8250
## theta(x.year)    15.5361
##
## Variance estimate: 1.842994
##
## Nugget effect estimate: 0.0002808436
```

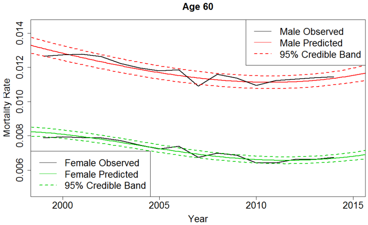
In-Sample Smoothing

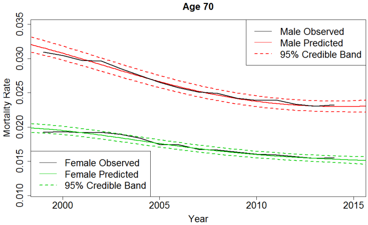
```
agesForecast <- seq(58,72,1)
yearsForecast <- 2010

rateForecastTrue <- dplyr::filter(mortData,age %in% agesForecast, year %in% yearsForecast)$rate
# build data frame for desired forecasts, then call predict
xPred <- data.frame(age=agesForecast,year=yearsForecast)
mortPred <- predict(mortModel, newdata=data.frame(x=xPred),
                    cov.compute=TRUE,
                    se.compute=TRUE,type="UK")

# plot mortality as a function of age
plot(agesForecast,rateForecastTrue,type="l",lwd=2, main="2010",
     xlab="Age",ylab="Mortality Rate",cex.axis=1.3,cex.lab=1.3,cex=1.3,
     xlim=c(60,70))
lines(agesForecast,exp(mortPred$mean),col=2,lty=2,lwd=2)
legend("topleft",c("Observed","Predicted"),lwd=c(2,2),lty=c(1,2),col=c(1,2),cex=1.5)
```

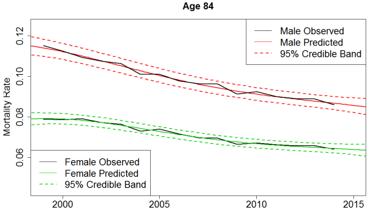


Mortality Over Time with Credible Bands

- Posterior mean and 95% credibility bands for f_* over calendar year
- Can observe increasing uncertainty at edges
- Observe mortality improvement then decline





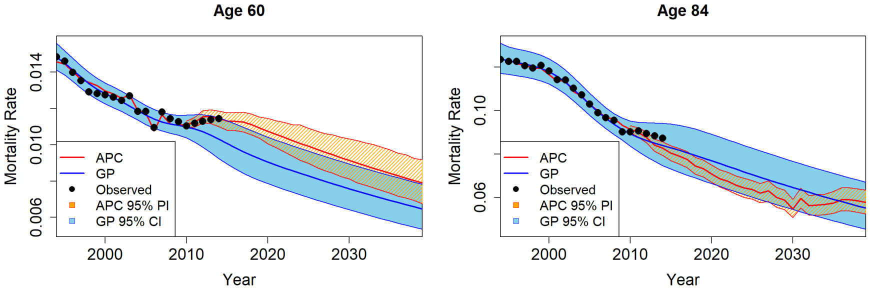


Extrapolation

- Compare to `apc` model from `R` package `stmomo`

$$\mu(x,t) = \alpha(x) + \kappa(t) + \gamma(t-x), \qquad \sum_c \gamma(c) = 0, \sum_c c\gamma(c) = 0$$

- GP produces similar forecasts with more desirable smoothness properties



Mortality Improvement

- Typical way is to look at the annual backward improvement

$$MI^{obs}_{back}\left(x_{ag}; yr\right) \doteq 1 - \frac{\exp\left(\mu(x_{ag}, yr)\right)}{\exp\left(\mu(x_{ag}, yr-1)\right)}$$

- $f_*(x_{ag}, yr)$ is a random variable, so we have the predicted mean improvement

$$m^{GP}_{back}\left(x_{ag}, yr\right) = \mathbb{E}\left[MI^{GP}_{back}\left(x_{ag}, yr\right)\right] \doteq \mathbb{E}\left[1 - \frac{\exp\left(f_*(x_{ag}, yr)\right)}{\exp\left(f_*(x_{ag}, yr-1)\right)}\right]$$

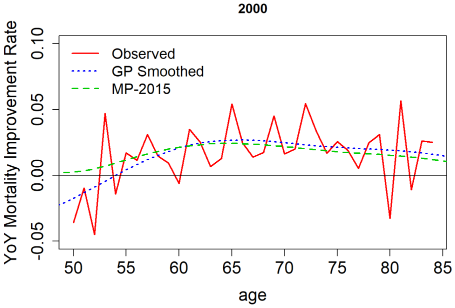
  - Available in closed form (lognormal distribution)
- Also have $MI^{MP}_{back}\left(x_{ag}; yr\right)$ (published MP-2015 improvement factors)

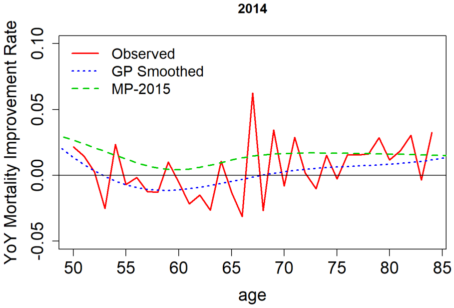
YoY Mortality Improvement Plots

```
# compare to MP-2015 tables
mp2015 <- read.csv("mp2015.csv")

agesForecast <- 48:86
agesObserved <- 50:84
yearsForecast <- c(2000,2014)
yearsPred <- c(1999,2000,2013,2014) # need to add one year back for improvement

nYr <- length(yearsPred)
nAg <- length(agesForecast)

# predict
xPred <- data.frame(age=rep(agesForecast,each=nYr),year=rep(yearsPred,nAg))
mortPred <- predict(mortModel, newdata=data.frame(x=xPred),cov.compute=TRUE,
             se.compute=TRUE,type="UK")

xPred$m <- mortPred$mean
for(yr in yearsForecast){
  forwardObs <- dplyr::filter(mortData, age %in% agesObserved, year == yr)$rate
  backwardObs <- dplyr::filter(mortData, age %in% agesObserved, year == yr-1)$rate
  MIbackobs <- 1-forwardObs/backwardObs    # raw observed improvement rates

  forwardPred <- filter(xPred, age %in% agesForecast, year == yr)$m
  backwardPred <- filter(xPred, age %in% agesForecast, year == yr-1)$m
  # smoothed improvement rates using the GP model
  mibackgp <- 1-exp(forwardPred)/exp(backwardPred)

  plot(agesObserved,MIbackobs, type="l", lwd=2, main = yr, ylim=c(-0.05, 0.1), xlab="age",
       ylab="YoY Mortality Improvement Rate", cex.axis=1.5,cex.lab=1.5,cex=1.5, col=2)
  lines(agesForecast,mibackgp, col=4, lwd=2, lty=3)
  lines(c(0,100),c(0,0))

  # MP-2015 rates
  mpRate <- dplyr::filter(mp2015, gender=="male", age %in% agesForecast, year == yr)$improvement
  lines(agesForecast, mpRate, col=3, lwd=2, lty=2)

  legend("topleft",c("Observed","GP Smoothed","MP-2015"), col=c(2,4,3),lwd=rep(2,3),
```

Comparing Mortality Improvement Methods

- Raw improvements extremely noisy (unsurprising)
- Smoothed methods both follow data well
- GP implies a stronger decline
  - Additional data suggests mortality deceleration

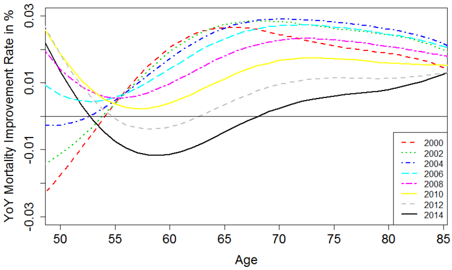




GP Improvement Over Time

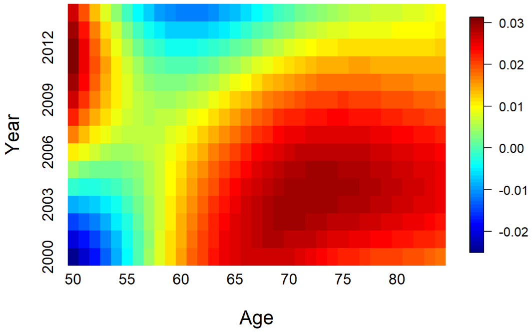
GP Improvements from 2000–2014 (in 2 year increments)

- Shape changes (flips) over time
- Consistent with MP-2015
- Generally decelerating after age 55



GP Mortality Improvement Heatmap

- Heatmap indicates possible cohort type relation with mortality improvement



Backward Difference & Derivatives

f_* denotes the fitted GP

$$1 - \left(\frac{\exp\left(f_*(x_{ag}, yr)\right)}{\exp\left(f_*(x_{ag}, yr - h)\right)} \right)^{1/h} \approx -\frac{f_*(x_{ag}, yr) - f_*(x_{ag}, yr - h)}{h} \quad (3)$$

- As defined, the typical annual mortality improvements are backward differences with $h = 1$
- Right side remains a GP by linearity
- Taking limit as $h \to 0$ yields derivative
  - Exists (depending on covariance kernel)
- Closed form expressions for distribution of $\frac{\partial f_*}{\partial x_{yr}}$

GP Derivative

Proposition

For the Gaussian Process f_* with a twice differentiable covariance kernel C, the limiting random variables

$$\frac{\partial f_*}{\partial x_{yr}}(x_{ag}, yr) \doteq \lim_{h \to 0} \frac{f_*(x_{ag}, yr+h) - f_*(x_{ag}, yr)}{h} \tag{4}$$

exist in mean square and form a Gaussian process $\frac{\partial f_*}{\partial x_{yr}} \sim GP(m_{diff}, s_{diff})$. Given the training set $\mathcal{D} = (\mathbf{x}, \mathbf{y})$, the posterior distribution of $\frac{\partial f_*}{\partial x_{yr}}(x_*)$ has mean and variance

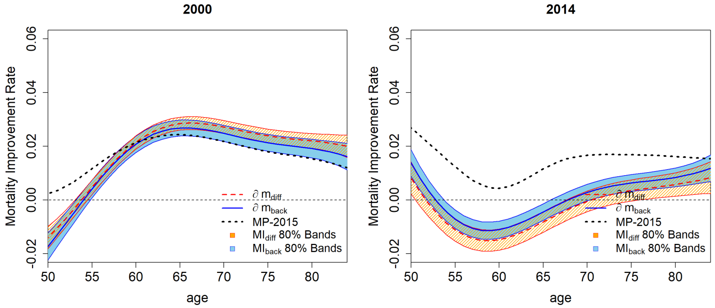
$$\begin{cases} m_{diff}(x_*) = \mathbb{E}\left[\frac{\partial f_*}{\partial x_{yr}}(x_*) \middle| \mathbf{x}, \mathbf{y} \right] = \frac{\partial C}{\partial x'_{yr}}(\mathbf{x}, x_*)(\mathbf{C} + \Sigma)^{-1}\mathbf{y}, \\ s^2_{diff}(x_*) = Var\left(\frac{\partial f_*}{\partial x_{yr}}(x_*) \middle| \mathbf{x}, \mathbf{y} \right) = \frac{\partial^2 C}{\partial x_{yr} \partial x'_{yr}}(x_*, x_*) - \frac{\partial C}{\partial x'_{yr}}(\mathbf{x}, x_*)(\mathbf{C} + \Sigma)^{-1} \frac{\partial C}{\partial x_{yr}}(x_*, \mathbf{x}), \end{cases}$$

where $\frac{\partial C}{\partial x'_{yr}}(\mathbf{x}, x_*) = \left[\frac{\partial C}{\partial x'_{yr}}(x^1, x_*), \ldots, \frac{\partial C}{\partial x'_{yr}}(x^N, x_*) \right]$ and each component is computed as the partial derivative of $C(x, x')$.

See Theorem 2.2.2 in Adler (2010) for more details/proof.

Comparing Other Methods with GP Derivative

- Blue is backwards mortality difference (as before); red is GP derivative; black is MP-2015
- Analysis of other years shows deceleration begins around 2010
  - Implies mortality evolution is convex
    - Justifies accelerating divergence between yearly difference and derivative methods
  - MP-2014 and MP-2015 begin to diverge around 2010
    - Suggests that later years are crucial to mortality forecasts



R Notebook Comments

- R notebook approach illustrates ease of use of GP's
- Practitioners have many options:
  - can simply change `.csv` file to their own data
  - can change output ranges for plots (e.g. plot 2016 instead of 2015)
  - have access to each plot and piece of code so programmers can specialize if needed
- Applied to US Females, Japan Male/Female, UK Male/Female data
  - Showed plausible forecasts for mortality and mortality improvement

Conclusions

- GP's provide a variety of benefits to modeling mortality and mortality improvement
  - Bayesian approach (data driven)
  - Posterior distribution for any location
    - Including distribution of mortality improvement (both yearly difference and instantaneous)
    - Credible bands (historic and forecasting)
- Relatively consistent results with MP-2015
  - Four years of additional data pushes GP results in the direction that MP-2015 took compared to MP-2014 (and MP-2016, 2017 found later)
  - Differences in results is likely due to data differences than model issues
- GP framework easily handles joint analysis of mortality rates and mortality improvement

Future Work

- Modeling annual mortality improvement directly with GP
- Monotonicity constraint: $f \Big|_{\frac{\partial f}{\partial f_{age}} > 0}$
- Multiple populations
  - Jointly modeling male & female mortality
  - Multivariate GP of multiple countries and factors
- Modeling by cause of death

References

M. Ludkovski, J. Risk, H. Zail, 2018
Gaussian Process Models for Mortality Rates and Improvement Factors
ASTIN Bulletin, 1–41. doi:10.1017/asb.2018.24

J. Risk, M. Ludkovski 2018
Code for Analyzing Mortality Rates and Improvements using Gaussian Processes.
GitHub repository https://github.com/jimmyrisk/GPmortalityNotebook

Roustant, O., Ginsbourger, D., Deville, Y., et al. 2012.
Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization.
Journal of Statistical Software, 51(1):1–55.

Williams, C. K. and Rasmussen, C. E. 2006.
Gaussian processes for machine learning, the MIT Press.

Adler, Robert J. 2010
The geometry of random fields, Siam

References

M. Ludkovski, J. Risk, H. Zail, 2018
Gaussian Process Models for Mortality Rates and Improvement Factors
ASTIN Bulletin, 1–41. doi:10.1017/asb.2018.24

J. Risk, M. Ludkovski 2018
Code for Analyzing Mortality Rates and Improvements using Gaussian Processes.
GitHub repository https://github.com/jimmyrisk/GPmortalityNotebook

Roustant, O., Ginsbourger, D., Deville, Y., et al. 2012.
Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization.
Journal of Statistical Software, 51(1):1–55.

Williams, C. K. and Rasmussen, C. E. 2006.
Gaussian processes for machine learning, the MIT Press.

Adler, Robert J. 2010
The geometry of random fields, Siam

THANK YOU!