ماشین حساب اعداد فازی

عباس پرچمی[۱]

تاریخ دریافت: ۱۳۹۵/۵/۱۱
تاریخ پذیرش: ۱۳۹۵/۱۲/۲۵

چکیده:

در این مقاله، مرور و مقایسه‌ای بر دو بستهٔ نرم‌افزاری FuzzyNumbers و Calculator.LR.FNs انجام شده است. این بسته‌ها قابلیت نصب بر روی نرم‌افزار R را دارند و در واقع ابزار و توابعی در اختیار کابران‌شان قرار می‌دهند که به‌وسیلهٔ آنها رسم و انجام عملیات حسابی بر روی اعداد فازی LR به‌سادگی میسر می‌شود. برای درک بهتر خوانندگان، در این مقاله سعی شده است تا دستورالعمل‌ها و مقایسات به‌وسیلهٔ مثال‌های عددی زیادی مطرح و توضیح داده شوند و پیشنهاد می‌شود که خوانندگان دستورات مثال‌ها را در نرم‌افزار R عملاً اجرا و دنبال کنند.

واژه‌های کلیدی: بستهٔ نرم‌افزاری R، عدد فازی LR ، عملگر حسابی، اصل گسترش.

۱ معرفی مقدماتی نرم‌افزار R

R، یک زبان برنامه‌نویسی و نرم‌افزار قدرتمند برای اجرای محاسبات ریاضی و انجام تجزیه و تحلیل‌های آماری (بر روی اعداد، کاراکترها، بردارها، ماتریس‌ها و آرایه‌ها) می‌باشد که بر اساس زبان‌های S و +S پیاده‌سازی شده است. این نرم‌افزار تحت اجازه‌نامهٔ انتشار همگانی گنو[۲] عرضه شده و به‌رایگان در دسترس عموم است. R، حاوی محدوده‌ی گسترده‌ای از فنون آماری و همچنین قابلیت‌های گرافیکی است. در محیط R، کدهای C، C++ و Fortran قابلیت اتصال و فراخوانی هنگام اجرای برنامه را نیز دارند. یکی از ویژگی‌های مهم این نرم‌افزار، امکان توسعه‌ی قابلیت‌های R، با افزودن بسته‌های[۳] نرم‌افزاری ایجاد شده توسط کاربران آن است. مجموعه‌ای از بسته‌های اصلی R، هنگام نصب همراه برنامه وجود دارند و در مجموع ۷۸۰۰ بسته (تا ژانویهٔ ۲۰۱۶ میلادی) در شبکهٔ بایگانی فراگیر R (CRAN)[۴] ثبت شده است. این بسته‌ها که توسط R، C++، Java و Fortran نوشته شده‌اند، طیف وسیعی از قابلیت‌ها را در زمینه‌های مختلف تحلیل داده‌ها به نرم‌افزار R اضافه می‌کنند و در لینک زیر فهرست شده‌اند.

https://cran.r-project.org/web/packages

۲ بسته‌های فازی در R

گرچه نرم‌افزار R اغلب به‌منظور اجرای محاسبات آماری به کار می‌رود، اما این نرم‌افزار قابلیت انجام برخی از محاسبات پیچیده‌ی فازی را نیز دارد که البته بسته‌های موجود در خصوص مجموعه‌های فازی موجبات تسهیل در این امر را فراهم آورده‌اند. در حال حاضر تعداد ۲۲ بسته‌ی نرم‌افزاری بر روی CRAN به ثبت رسیده که در زیر به ترتیب حروف الفبای لاتینی فهرست شده‌اند:

anfis (Adaptive Neuro Fuzzy Inference System in R)

Calculator.LR.FNs (Calculator for LR Fuzzy Numbers)

fclust (Fuzzy Clustering)

FCMapper (Fuzzy Cognitive Mapping)

FLR (Fuzzy Logic Rule Classifier)

frbs (Fuzzy Rule-Based Systems for Classification and Regression Tasks)

fso (Fuzzy Set Ordination)

fugeR (Fuzzy Genetic, a machine learning algorithm

[۱] بخش آمار دانشگاه شهید باهنر کرمان

[۲] GNU General Public License
[۳] packages
[۴] Comprehensive R Archive Network

to construct prediction model based on fuzzy logic)
fuzzyFDR (Exact calculation of fuzzy decision rules for multiple testing)
fuzzyforest (Fuzzy Forests)
FuzzyLP (Fuzzy Linear Programming)
FuzzyNumbers (Tools to Deal with Fuzzy Numbers)
Fuzzy.p.value (Computing Fuzzy p-Value)
fuzzyRankTests (Fuzzy Rank Tests and Confidence Intervals)
FuzzyStatProb (Fuzzy Stationary Probabilities from a Sequence of Observations of an Unknown Markov Chain)
FuzzyToolkitUoN (Type 1 Fuzzy Logic Toolkit)
lfl (Linguistic Fuzzy Logic)
Rfmtool (Fuzzy Measure Tools for R)
RoughSets (Data Analysis Using Rough Set and Fuzzy Rough Set Theories)
SAFD (Statistical Analysis of Fuzzy Data)
vegclust (Fuzzy clustering of vegetation data)
Weighted.Desc.Stat (Weighted Descriptive Statistics)

از میان این بسته‌ها در ادامه به معرفی دو بستهٔ نرم‌افزاری FuzzyNumbers و Calculator.LR.FNs می‌پردازیم.

۳ بستهٔ نرم‌افزاری FuzzyNumbers

نام کامل این بسته Tools to Deal with Fuzzy Numbers است و نسخهٔ 0.4-1 آن در سال ۲۰۱۵ میلادی توسط مارک گاگلوزکی و کاها[۵] از هلند تهیه و بر روی CRAN بارگذاری شده است [۴]. در ابتدا به بررسی روش‌های معرفی و شناسایی یک عدد فازی در بستهٔ FuzzyNumbers می‌پردازیم.

۱.۳ معرفی عدد فازی

این بسته از قابلیت معرفی اعداد فازی به روش‌های گوناگون برخوردار است که در ادامه با ذکر مثال به آن‌ها اشاره می‌شود.

۱.۱.۳ عدد فازی LR

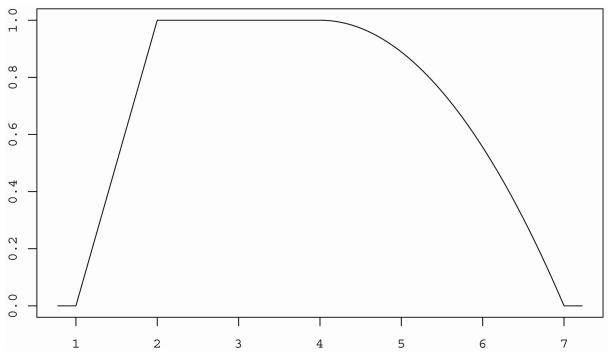
تعریف ۱.۳. (عدد فازی در بستهٔ FuzzyNumbers) عدد فازی A یک زیرمجموعهٔ فازی از R با تابع عضویت

$$\mu_A(x) = \begin{cases} , & x < a \\ left\left(\frac{x-a}{a-a}\right), & a \le x < a \\ , & a \le x < a \\ right\left(\frac{x-a}{a-a}\right), & a \le x < a \\ , & a < x \end{cases}$$

است که در آن $a, a, a, a \in R$، $left : [,] \to [,]$ تابعی غیرنزولی و $right : [,] \to [,]$ تابعی غیرصعودی است.

مثال ۲.۳. فرض کنید می‌خواهیم عددی فازی با هستهٔ $(a, a) = (,)$، دامنهٔ $(a, a) = (,)$ و توابع شکل چپ و راست $left(x) = x$ و $right(x) = -x$ را به بستهٔ *FuzzyNumbers* معرفی و تابع عضویت آن را رسم کنیم. بدین منظور، می‌توان نوشت:

```
A1 <- FuzzyNumber(1, 2, 4, 7,
left=function(x) x,
right=function(x) 1-x^2
)
plot(A1)
```

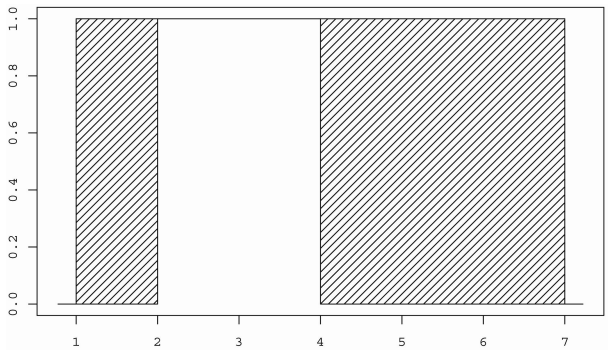


شکل ۱. نمودار تابع عضویت عدد فازی در مثال ۲.۳

تذکر ۳.۳. در صورت حذف توابع شکل در تابع *FuzzyNumber*، یک مجموعهٔ فازی اصطلاحاً سایه‌دار به‌دست می‌آید که در آن مقدار تابع عضویت در بازه‌های (a, a) و (a, a) مشخص نیست. به‌عنوان نمونه، از اجرای دستور زیر شکل ۲ حاصل می‌شود.

```
A2 <- FuzzyNumber(1, 2, 4, 7)
```

[۵] Gagolewski and Caha



شکل۲. نمودار تابع عضویت عدد فازی سایه‌دار مربوط به تذکر ۳.۳

۲.۱.۳ اعداد فازی مثلثی و ذوزنقه‌ای

مثال ۴.۳. چند روش برای معرفی اعداد فازی مثلثی، ذوزنقه‌ای و همچنین معرفی توابع نشانگر در این مثال مطرح شده است.

```
T1 <- TrapezoidalFuzzyNumber(1, 1.5, 4, 7)
plot(T1)
T2 <- TrapezoidalFuzzyNumber(1.4, 3, 3, 7)  # or
equivalently:  T2 <- TriangularFuzzyNumber(1.4,3,7)
plot(T2, add=TRUE, col=2, lty=3)
TrapezoidalFuzzyNumber(2,2,3,3) # 'crisp' interval
as.TrapezoidalFuzzyNumber(c(2,3)) # is the same
TrapezoidalFuzzyNumber(5,5,5,5) # 'crisp' real
as.TrapezoidalFuzzyNumber(5) # is the same
```

۳.۱.۳ عدد فازی قطعه‌قطعه‌ای خطی

به‌وسیلهٔ تابع PiecewiseLinearFuzzyNumber می‌توان نوع دیگری از اعداد فازی (معروف به عدد فازی قطعه‌قطعه‌ای خطی) را در بستهٔ FuzzyNumbers تولید کرد. یکی از مؤلفه‌های این تابع knot.n است که تعداد برش‌های قطعه‌قطعه‌کنندهٔ عدد فازی را تعیین می‌کند. با استفاده از مؤلفه‌های knot.alpha، knot.left و knot.right به‌ترتیب می‌توان ارتفاع، ابتدا و انتهای برش‌ها (گره‌ها) را نیز تعیین کرد.

مثال ۵.۳. (معرفی عدد فازی قطعه‌قطعه‌ای خطی)

```
P1 <- PiecewiseLinearFuzzyNumber(1, 2, 3, 4,
knot.n=1, knot.alpha=0.25, knot.left=1.5,
knot.right=3.25)
plot(P1, xlim=c(0.5,4.5))
P1
P2 <- PiecewiseLinearFuzzyNumber(1, 2, 3, 4,
knot.n=2, knot.alpha=c(0.25,0.6),
knot.left=c(1.5,1.8), knot.right=c(3.25, 3.5))
plot(P2, col=2, lty=2, pch=2, add=TRUE)
P2
```

۴.۱.۳ عدد فازی توانی

اگر توابع شکل در عدد فازی LR برابر با $left(x) = x^{p.left}$ و $right(x) = (-x)^{p.right}$ در نظر گرفته شوند، آنگاه عدد حاصل را در این بسته «عدد فازی توانی» می‌نامند. عدد فازی توانی به‌وسیلهٔ تابع PowerFuzzyNumber قابل معرفی است و بدیهی است که هر عدد ذوزنقه‌ای حالت خاصی از اعداد فازی توانی به‌ازای $p.left = p.right =$ می‌باشد (مثال ۱۱.۳ را ببینید).

۲.۳ محاسبهٔ هسته، دامنه و برش‌های عدد فازی

هسته، دامنه و α-برش‌های یک عدد فازی به‌ترتیب با توابع core، supp و alphacut قابل محاسبه‌اند. همچنین مقدار تابع عضویت در یک نقطهٔ خاص با تابع evaluate به‌دست می‌آید. مثال ۶.۳ مطلب را روشن می‌سازد.

مثال ۶.۳.

```
A <- TrapezoidalFuzzyNumber(1, 1.5, 4, 7)
plot(A)
cuts <- alphacut(A, c(0, 0.5, 1))
cuts
cuts[, 1] #  or equivalently  cuts[, "L"]
supp(A)
core(A)
evaluate(A, 0)
evaluate(A, c(-3,0,3))
evaluate(A, seq(-1, 2, by=0.5))
```

۳.۳ عملگرهای حسابی برای اعداد فازی

عملگرهای دوبعدی جمع، تفریق، ضرب و تقسیم به‌ترتیب با علایم $+$، $-$، $*$ و $/$ در بستهٔ FuzzyNumbers تعریف می‌شوند که عملاً در میان دو عدد فازی نوشته می‌شوند. همچنین عملگر

ضرب اسکالر نیز با علامت * در این بسته قابل شناسایی است که میان یک عدد فازی و یک عدد حقیقی قرار می‌گیرد.

مثال ۷.۳. (جمع دو عدد مثلثی و ذوزنقه‌ای)

```
A <- TrapezoidalFuzzyNumber(0, 1, 1, 2)
B <- TriangularFuzzyNumber(1, 2, 3)
plot(A, xlim=c(0,6))
 plot(B, add=TRUE, col=2, lty=2)
 plot(A+B, add=TRUE, col=4, lty=4)
```

گرچه اعمال جمع و تفریق برای اعداد مثلثی و ذوزنقه‌ای به‌طور مستقیم به‌آسانی قابل استفاده است، اما در صورتی که قصد استفاده از سایر عملگرها یا انواع دیگری از اعداد فازی را در این بسته دارید، باید در ابتدا اعداد فازی مورد نظر را به‌وسیلهٔ تابع PiecewiseLinearFuzzyNumber قطعه‌قطعه کرده، سپس عملگرها را بر روی اعداد قطعه‌قطعه‌شده اعمال نمایید.

تذکر ۸.۳. اعداد قطعه‌قطعه‌شدهٔ خطی را با سه روش: **تولید** (به‌وسیلهٔ دستور *PiecewiseLinearFuzzyNumber*)، **تبدیل** (به‌وسیله‌ی دستور *as.PiecewiseLinearFuzzyNumber*) و **تخمین** (به‌وسیله‌ی دستور *piecewiseLinearApproximation*) می‌توان در بستهٔ *FuzzyNumbers* ایجاد کرد. تفاوت دو دستور نخست آن است که با اولین تابع می‌توان یک عدد فازی قطعه‌قطعه‌ای خطی را به‌طور مستقیم تولید کرد، اما با دومین تابع می‌توان یک عدد فازی مثلثی/ذوزنقه‌ای تولید شده را به‌طور دقیق (و نه تخمینی) به یک عدد فازی قطعه‌قطعه‌ای تبدیل کرد.

تذکر ۹.۳. در بستهٔ *FuzzyNumbers*، سه روش مختلف زیر برای تخمینِ قطعه‌قطعه‌ای یک عدد فازی در نظر گرفته شده است:

- **روش کمترین فاصلهٔ اقلیدسی (*NearestEuclidean*):** در این روش، تخمین یک عدد فازی به‌وسیلهٔ یک عدد فازی قطعه‌قطعه‌ای به‌گونه‌ای انجام می‌گیرد که کمترین فاصلهٔ اقلیدسی را از عدد فازی اصلی داشته باشد [۲، ۳]. تعریف فاصلهٔ اقلیدسی بین دو عدد فازی در زیربخش ۵.۳ آورده شده و همچنین توجه داشته باشید که روش کمترین فاصلهٔ اقلیدسی، روش پیش‌فرض تابع *piecewiseLinearApproximation* است.
- **روش محافظ هسته و دامنه (*SupportCorePreserving*):** همان‌گونه که از نام این روش بر می‌آید، هسته و دامنهٔ تخمین قطعه‌قطعه‌ای یک عدد فازی (به‌روش محافظ هسته و دامنه)، معادل هسته و دامنهٔ همان عدد فازی اصلی است. ملاک دوم این روش، که در حال حاضر تنها به‌ازای *knot.n=1* فعال است، کمینه کردن فاصلهٔ اقلیدسی بین عدد فازی و تخمینش می‌باشد.
- **روش ساده (*Naive*):** در این روش، تخمین قطعه‌قطعه‌ای یک عدد فازی غیرخطی به‌گونه‌ای صورت می‌گیرد که (الف) هستهٔ عدد فازی اصلی برابر با هستهٔ عدد فازی تخمینی باشد، (ب) دامنهٔ عدد فازی اصلی برابر با دامنهٔ عدد فازی تخمینی شود، و (ج) برش‌های معرفی‌شده در دستور برای عدد فازی اصلی، برابر با برش‌های متناظر عدد فازی تخمینی باشند.

مثال ۱۰.۳. (تخمین یک عدد فازی با دو روش ساده و کمترین فاصلهٔ اقلیدسی)

```
A <- FuzzyNumber(-5, 3, 6, 20, left=function(x)
 pbeta(x,0.4,3),
right=function(x) 1-x^(1/4),
lower=function(alpha) qbeta(alpha,0.4,3),
upper=function(alpha) (1-alpha)^4)
plot(A, xlim=c(-6,21))
P1 <- piecewiseLinearApproximation(A, method=
'Naive', knot.n=1)
P1['allknots']
plot(P1, add=TRUE, col=2, lty=2, lwd=2)
print(distance(A, P1), 3)
P2 <- piecewiseLinearApproximation(A, method=
'NearestEuclidean', knot.n=1)
P2['allknots']
plot(P2, add=TRUE, col=3, lty=3, lwd=2)
print(distance(A, P2), 3)
```

مثال ۱۱.۳.

```
A <-
piecewiseLinearApproximation(PowerFuzzyNumber
```

```
(1,2,3,4, p.left=2, p.right=0.5), knot.n=20)
A+A
plot(A, xlim=c(0,8))
plot(A+A, col=2, add=TRUE)
plot(2*A, col=3, lty=2, lwd=2, add=TRUE)
# the same as A+A
```

اجرای مثال ۱۱.۳ را با $knot.n = $ تکرار و سپس نتیجه را مقایسه کنید.

مثال ۱۲.۳. (چهار عمل اصلی)

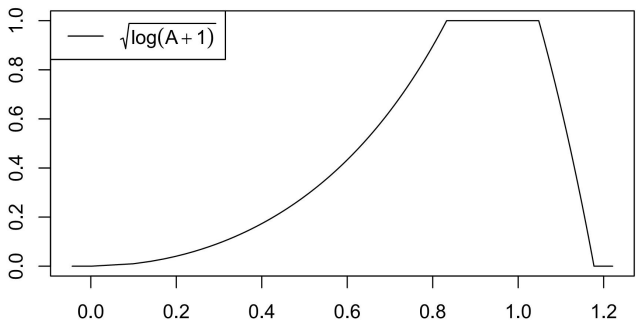
```
A <-
piecewiseLinearApproximation(PowerFuzzyNumber
(-1, 1, 1, 3, p.left=1, p.right=1),
method="Naive", knot.n=150)
B <-
piecewiseLinearApproximation(PowerFuzzyNumber
(1, 3, 3, 5, p.left=1, p.right=1),
method="Naive", knot.n=150)
plot(A,xlim=c(-6,15), lwd=2)
plot(B, add=TRUE, col=1, lwd=2)
plot(A+B, add=TRUE, col=2, lwd=2)
plot(A-B, add=TRUE, col=3, lwd=2)
plot(A*B, add=TRUE, col=4, lwd=2)
plot(A/B, add=TRUE, col=5, lwd=2)
```

۴.۳ اِعمال یک تابع یکنوا بر عدد فازی

یکی از مهم‌ترین قابلیت‌های این بسته، اعمال اصل گسترش [۱] برای توسیع و گسترش توابع تک‌متغیره (و البته یک‌به‌یک) به‌گونه‌ای است که بتوان به‌جای یک عدد حقیقی، یک عدد فازی را در تابع قرار داد. این کار در بستهٔ FuzzyNumbers توسط تابع fapply میسر می‌شود و شایان ذکر است که عدد فازی مورد نظر باید از نوع قطعه‌قطعه‌شدهٔ خطی باشد.

مثال ۱۳.۳. (محاسبهٔ تابع عضویت عدد فازی $\sqrt{log(A+)}$)

```
A <- as.PiecewiseLinearFuzzyNumber(
TrapezoidalFuzzyNumber(0,1,2,3), knot.n=100)
plot(A, xlim=c(0,3))
plot( fapply(A, function(x) sqrt(log(x+1))))
```



شکل۳. نمودار تابع عضویت عدد فازی $\sqrt{log(A+)}$ در مثال ۱۳.۳

۵.۳ فاصلهٔ بین دو عدد فازی

تعمیمی از فاصلهٔ اقلیدسی (L) که برای دو عدد فازی A و B به‌صورت

$$d_E(A,B) = \left\{\int [A_L(\alpha) - B_L(\alpha)]\, d\alpha + \int [A_U(\alpha) - B_U(\alpha)]\, d\alpha\right\}$$

تعریف می‌شود، در بستهٔ FuzzyNumbers به‌وسیلهٔ تابع distance قابل فراخوانی است. این تابع به دو روش اقلیدسی (Euclidean) (که پیش‌فرض تابع است) و روش مربع اقلیدسی (EuclideanSquared) همانند مثال ۱۴.۳ قابل استفاده است.

مثال ۱۴.۳.

```
T1 <- TrapezoidalFuzzyNumber(-5, 3, 6, 20)
T2 <- TrapezoidalFuzzyNumber(-4, 4, 7, 21)
distance(T1, T2, type='Euclidean') #L2 distance
distance(T1, T2, type='EuclideanSquared')
# Squared L2 distance
```

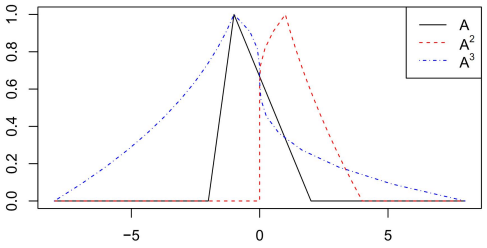
۶.۳ چند مثال محاسباتی

در این بخش به‌کمک توابع مطرح‌شده در بستهٔ FuzzyNumbers، چند مثال محاسباتی ارایه می‌شود.

مثال ۱۵.۳. (محاسبهٔ توان دوم و سوم عدد فازی $Tr(-,-,-,)$ و رسم نمودار توابع عضویت آن‌ها)

```
A <- as.PiecewiseLinearFuzzyNumber(
TrapezoidalFuzzyNumber(-2,-1,-1,2), knot.n=10)
```

```
plot(A, xlim=c(-8,8))
plot(A^2, add=TRUE, col=2, lty=2)
plot(A^3, add=TRUE, col=4, lty=4)
```



شکل ۴. نمودار توابع عضویت اعداد فازی A، A و A در مثال ۱۵.۳

مثال ۱۶.۳. (محاسبهٔ توابع عضویت میانگین، واریانس و انحراف معیار چند عدد فازی ذوزنقه‌ای)

```
x1 <-
as.PiecewiseLinearFuzzyNumber(
TrapezoidalFuzzyNumber(0,1,2,3), knot.n=20)
x2 <-
as.PiecewiseLinearFuzzyNumber(
TrapezoidalFuzzyNumber(-1,2,2,2.5), knot.n=20)
x3 <-
as.PiecewiseLinearFuzzyNumber(
TrapezoidalFuzzyNumber(2,3,4,4.5), knot.n=20)
plot(x1, col=1, xlim=c(-2,15))
plot(x2, col=2, add=TRUE)
plot(x3, col=3, add=TRUE)
mean = (x1+x2+x3)/3
plot(mean, col=1, lwd=4, lty=3, add=TRUE)
s2 =((x1-mean)^2+(x2- mean)^2+(x3-mean)^2)/3
plot(s2, col=2, lwd=3, lty=4, add=TRUE)
s = fapply( s2, function(x) sqrt(x))
plot(s, col=4, lwd=3, lty=6, add=TRUE)
legend( "topright", c("X1 = Tr(0,1,2,3)",
"X2 = Tr(-1,2,2,2.5)", "X3 = Tr(2,3,4,4.5)",
"Mean", "Var", "s"), col = c(1,2,3,1,2,4),
text.col = 1, lwd = c(2,2,2,4,3,3),
lty = c(1,1,1,3,4,6))
```

شکل ۵ را ببینید، که برای وضوح بیشتر در انتهای مقاله رسم شده است.

مثال ۱۷.۳. عدد فازی ذوزنقه‌ای زیر را در نظر بگیرید

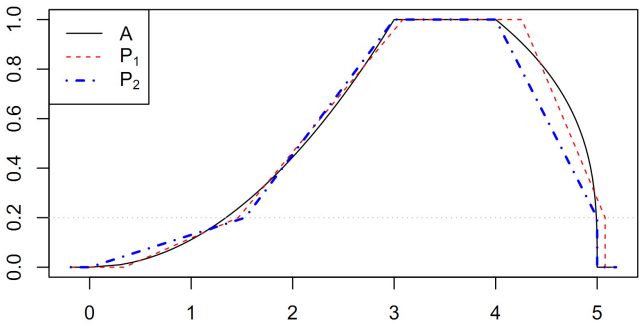
```
A <- FuzzyNumber(0, 3, 4, 5,
lower=function(x) qbeta(x, 2, 1),
upper=function(x) 1-x^3
)
```

به‌منظور یافتن بهترین تخمین‌زنندهٔ قطعه‌قطعه‌ای خطی برای A، قصد داریم مقدار α را به‌گونه‌ای بیابیم که تخمین‌یافتهٔ قطعه‌قطعه‌ای خطی A (با $knot.n$ و $knot.alpha = \alpha$) به‌روش SupportCorePreserving کمترین فاصلهٔ اقلیدسی را از عدد فازی A داشته باشد. برای این منظور، در ابتدا فاصلهٔ اقلیدسی عدد فازی A را از تخمینش، به‌صورت تابع $D(\alpha) = d_E(A, P_\alpha(A))$ در زیر معرفی می‌کنیم و سپس به دنبال αای در بازهٔ $(,)$ می‌گردیم که این تابع را کمینه سازد.

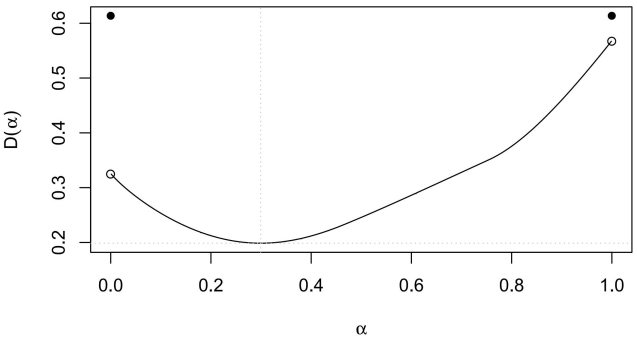
```
D <- function(a) distance(A,
piecewiseLinearApproximation(A,
method="SupportCorePreserving", knot.alpha=a))
optimize(D, lower=0, upper=1)
```

حال بعد از مشخص شدن مقدار دقیق α بهینه، می‌توان ادعا کرد که در ردهٔ همهٔ برآوردگرهای قطعه‌قطعه‌ای خطی با $knot.n =$، عدد فازی زیر بهترین برآورد قطعه‌قطعه‌ای خطی برای A است:

```
piecewiseLinearApproximation(A,
method="SupportCorePreserving", knot.alpha=
alpha_0 )
```



شکل ۶. نمودار تابع عضویت A و دو تخمین برای آن در مثال ۱۷.۳



شکل۷. نمودار فاصلهٔ اقلیدسی بین A و تخمینش در مثال ۱۷.۳

۴ بستهٔ نرم‌افزاری $Calculator.LR.FNs$

نام کامل این بسته Calculator for LR Fuzzy Numbers است و همان طور که از نامش بر می‌آید این بسته یک ماشین حساب مقدماتی برای اعداد فازی LR می‌باشد. قابلیت اصلی نسخهٔ 1.1 این بسته که در سال ۲۰۱۶ میلادی بر روی CRAN بارگذاری شد [۵]، انجام چهار عمل اصلی و همچنین ضرب اسکالر بر روی ردهٔ اعداد فازی LR است.

۱.۴ معرفی عدد فازی

تابع عضویت عدد فازی LR در بستهٔ Calculator.LR.FNs با آنچه عدد فازی LR در بستهٔ FuzzyNumbers نامیده می‌شود، از نظر تعریف و طرز نمایش متفاوت است. لذا در ابتدا به تعریف عدد فازی LR در بستهٔ Calculator.LR.FNs می‌پردازیم که منطبق با تعریف عدد فازی LR در [۱] است.

تعریف ۱.۴. (عدد فازی LR در بستهٔ Calculator.LR.FNs) فرض کنید عدد فازی N یک زیرمجموعهٔ فازی از R با تابع عضویت

$$\mu_N(x) = \begin{cases} left.fun\left(\frac{n-x}{\alpha}\right), & x \le n \\ right.fun\left(\frac{x-n}{\beta}\right), & x > n \end{cases}$$

باشد به‌طوری که در آن $left.fun : R^+\cup \to [,]$ و $right.fun$: $[,] R^+\cup \to$ توابعی غیرنزولی، $right.fun() = left.fun() =$ و $\alpha, \beta >$ باشند. در این صورت N را یک عدد فازی LR نامیده، آن را با نماد $N = LR(n, \alpha, \beta)$ نشان می‌دهند که در آن n، α، β، $left.fun$ و $right.fun$ به‌ترتیب مقدار هسته، پهنای چپ، پهنای راست، تابع شکل چپ و تابع شکل راست نامیده می‌شوند.

تذکر ۲.۴. از توابع مختلفی می‌توان برای $left.fun$ و به‌طور مشابه برای $right.fun$ استفاده کرد؛ مثلاً $left.fun(x) =$ $exp(-x^p)$ و $right.fun(x) = max\{, -|x|^q\}$ که در آن‌ها $p, q \ge$ است.

تعریف ۳.۴. اگر $right.fun(x) = left.fun(x)$ باشد، آنگاه عدد فازی $LR(n, \alpha, \beta)$ را یک عدد فازی L می‌نامند و آن را با نماد $L(n, \alpha, \beta)$ نشان می‌دهند.

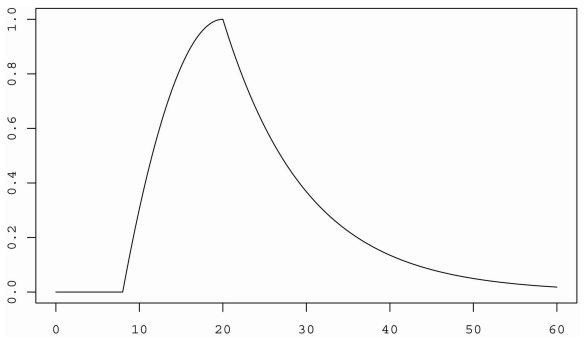
تذکر ۴.۴. اعداد فازی مثلثی و نرمال رایج‌ترین اعداد فازی L هستند که به‌ترتیب به‌ازای توابع شکل زیر به‌دست می‌آیند

$$left.fun(x) = \begin{cases} -x, & \le x \le \\ , & < x \end{cases}$$
$$left.fun(x) = e^{-x}, \quad x \ge .$$

توجه داشته باشید که اعداد فازی ورودی و خروجی در بستهٔ Calculator.LR.FNs فقط باید در ردهٔ اعداد فازی LR باشند و این بسته فقط توانایی کار با اعداد درون این رده را دارد. از همین رو، قبل از معرفی یک عدد فازی در این بسته، نخست باید توابع شکل آن را معرفی کرد. برای نمونه به مثال زیر توجه کنید.

مثال ۵.۴. برای معرفی و رسم عدد فازی $A = LR(,,)$ با توابع شکل $left.fun(x) = -x, \ x \ge$ و $right.fun(x) =$ $e^{-x}, \ x \ge$، می‌نویسیم:

```
Left.fun  = function(x)  { (1-x^2)*(x>=0)}
Right.fun = function(x)  { (exp(-x))*(x>=0)}
A = LR(20, 12, 10)
LRFN.plot(A, xlim=c(0,60), col=1)
```



شکل۸. نمودار تابع عضویت عدد فازی A در مثال ۵.۴

مثال ۶.۴. (معرفی و رسم سه نوع عدد فازی *LR*، *RL* و *L*)

معرفی اولین عدد فازی که از نوع LR است:

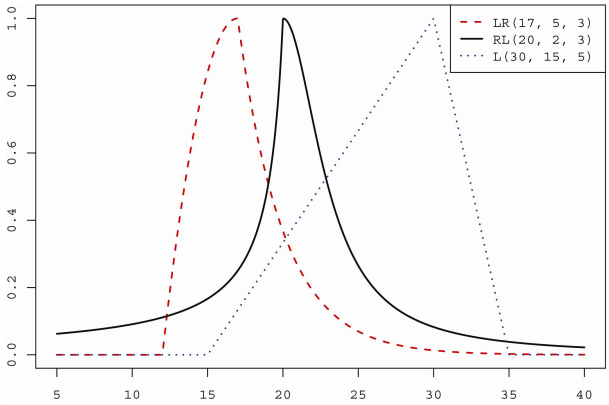
```
Left.fun  = function(x)  { (1-x^2)*(x>=0)}
Right.fun = function(x)  { (exp(-x))*(x>=0)}
LRFN.plot( LR(17,5,3), xlim=c(5,40),
lwd=2, lty=2, col=2)
```

معرفی دومین عدد فازی که از نوع RL است:

```
Left.fun  = function(x) {(1/(1+x^2))*(x>=0)}
Right.fun = function(x) {(1/(1+(2*abs(x))))*
(x>=0)}
LRFN.plot( RL(20,2,3), lwd=2, add=TRUE)
```

معرفی سومین عدد فازی که از نوع L است:

```
Left.fun  = function(x)  { (1-x)*(x>=0)}
LRFN.plot( L(30,15,5), lwd=2, lty=3, col=4,
add=TRUE)
legend("topright", c("LR(17,5,3)",
"RL(20,2,3)", "L(30,15,5)"), col = c(2,1,4),
text.col=1, lwd = c(2,2,2), lty = c(2,1,3))
```



شکل۹. توابع عضویت سه عدد فازی در مثال ۶.۴

تذکر ۷.۴. رسم اعداد فازی در بستهی *Calculator.LR.FNs* به‌وسیلهٔ تابع *LRFN.plot* انجام می‌شود. در استفاده از این دستور بهتر است که حدود محور xها برای رسم اولین عدد فازی به‌وسیلهٔ مؤلفهٔ *xlim* مشخص گردد. هم‌چنین اگر همانند مثال قبل، قصد چندین بار استفاده از دستور *LRFN.plot* را برای رسم چندین نمودار بر روی یک شکل داشته باشید، باید از مؤلفهٔ $add = TRUE$ استفاده کنید.

۲.۴ عملگرهای حسابی برای اعداد فازی

در بستهٔ Calculator.LR.FNs، عملگرهای دوبعدی جمع، تفریق، ضرب و تقسیم به ترتیب به‌وسیلهٔ توابع a، s، m و d (که مخفف addition، subtraction، multiplication و division می‌باشند) تعریف می‌شوند. همچنین عملگر ضرب اسکالر نیز با تابع $s.m$ در این بسته قابل استفاده است که میان یک عدد فازی LR و یک عدد حقیقی قرار می‌گیرد. گرچه مبنای این عملگرهای حسابی، اصل گسترش پ. عسگرزاده [۱] بوده، اما برای بسته بودن حاصل و نتیجه‌ی این عملگرها بر روی مجموعهٔ اعداد فازی LR، برای عملگرهای $\times$ و $\div$ از تقریب اعداد فازی LR نیز استفاده شده است.

از طرفی برای شفاف‌سازی و تأکید بیشتر، فرض کنید بعد از معرفی توابع شکل چپ و راست، مجموعه و ردهٔ همهٔ اعداد فازی LR، RL و L را به‌اختصار، ردهٔ اعداد فازی LR برای توابع شکل معرفی‌شده بنامیم. حال اگر حاصل عملیات حسابی اعداد فازیی که داخل ردهٔ اعداد فازی LR هستند، داخل همین رده قرار بگیرد، آنگاه بستهٔ Calculator.LR.FNs در پنجرهٔ Console نتیجه را در قالب یک عدد فازی LR، RL یا L به کاربر ارائه و گزارش می‌کند، و در غیر این‌صورت قابلیت انجام محاسبات مربوط را ندارد.

مثال ۸.۴. (میانگین n عدد فازی *LR*)

```
n = 10
Left.fun  = function(x)  { (1-x)*(x>=0)}
Right.fun = function(x)  { (exp(-x))*(x>=0)}
xlim=c(2, 18)
sum_x = c(0,0,0,0)
for (i in 1:n) {
 x = rnorm(1,10,3)
  x_l = runif(1,0,3)
  x_r = runif(1,0,2)
  X = c()
  X = LR(x, x_l, x_r)
  LRFN.plot( X, xlim=xlim, ylim=c(0,1), lwd=1,
  lty=1, col=1, add = (i != 1) )
  sum_x = a( sum_x , X )
}
sum_x
X_bar = s.m( (1/n) , sum_x )
LRFN.plot(X_bar, lwd=2, lty=2, col=2, add=TRUE)
```

```
legend("topright",c("LR FNs","mean of LR FNs"),
col=c(1,2),text.col=1,lwd=c(1,2),lty=c(1,2))
```

مثال ۹.۴. چهار عدد فازی $A = LR(,,)$ ، $B = LR(,,)$ ، $C = RL(,,)$ و $D = LR(,,)$ را مبتنی بر توابع شکل زیر در نظر می‌گیریم

$$\begin{aligned} left.fun(x) &= \quad -x, \ x \geq \\ right.fun(x) &= \quad -x, \ x \geq . \end{aligned}$$

علاوه بر رسم این چهار عدد فازی، قصد داریم مقادیر $\frac{A+B}{}$، $\frac{A+B}{} - C$ و $\left(\frac{A+B}{} - C\right) \times D$ را در ردهٔ اعداد فازی LR مشخص و سپس توابع عضویت آن‌ها را رسم کنیم.

```
Left.fun = function(x)  {(1-x^3)*(x>=0)}
Right.fun = function(x)  {(1-x)*(x>=0)}
A = LR(5, 0.5, 1)
B = LR(2, 0.3, 0.6)
C = RL(1, 0.7, 1.5)
D = LR(.5, 0.5, 1)
E = s.m(a(A,B), 1/2) # The mean of A and B
F = s(s.m(a(A,B), 1/2), C)
G = m(F,D)
LRFN.plot(A, xlim=c(-1,6), ylim=c(0,1.6),
lwd=3, lty=1, col=1)
LRFN.plot( B, lwd=3, lty=1, col=2, add=TRUE)
LRFN.plot( C, lwd=3, lty=1, col=3, add=TRUE)
LRFN.plot( D, lwd=3, lty=1, col=4, add=TRUE)
LRFN.plot( E, lwd=3, lty=2, col=5, add=TRUE)
LRFN.plot( F, lwd=3, lty=3, col=6, add=TRUE)
LRFN.plot( G, lwd=3, lty=4, col=7, add=TRUE)
legend( "topright", c("A = LR(5, 0.5, 1)",
"B = LR(2, 0.3, 0.6)", "C = RL(1, 0.7, 1.5)",
"D=LR(.5,0.5,1)","(A + B)/2=LR(3.5,0.4,0.8)",
"[(A + B)/2] - C = LR(2.5, 1.9, 1.5)",
"{[(A + B)/2] - C} * D = LR(1.25, 2.2, 3.2)"),
col = c(1:7), text.col = 1,
lwd=c(2,2,2,2,2,2,2), lty=c(1,1,1,1,2,3,4))
```

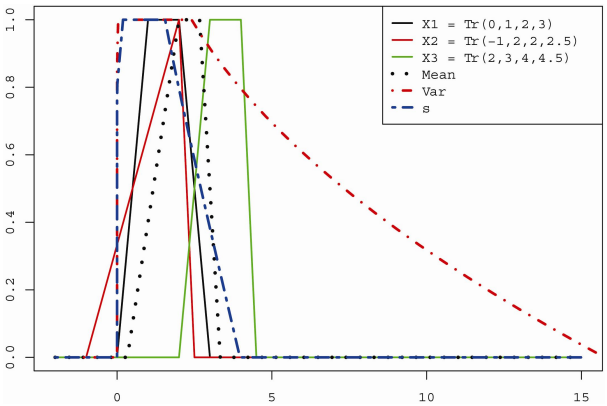
پس از اجرای برنامهٔ فوق به‌وسیلهٔ بستهٔ Calculator.LR.FNs، می‌توان نتایج محاسبات را در قالب اعداد فازی LR در پنجرهٔ Console به‌صورت زیر مشاهده کرد.

$$\frac{A+B}{} = LR(,,)$$
$$\frac{A+B}{} - C = LR(,,)$$
$$\left(\frac{A+B}{} - C\right) \times D = LR(,,)$$

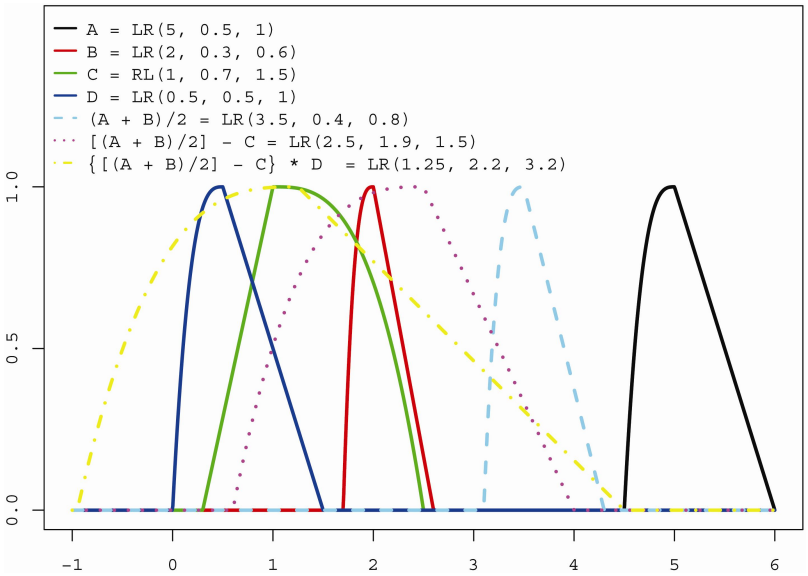
شکل ۱۰ حاوی نتایج این مثال است که برای دقت و وضوح بیشتر در انتهای مقاله آورده شده است.

۵ جمع‌بندی: مزیت‌ها و معایب بسته‌ها

مزیت بستهٔ Calculator.LR.FNs نسبت به بستهٔ FuzzyNumbers آن است که در صورت امکان، نتیجهٔ محاسبات را در قالب یک عدد فازی LR، RL یا L (و نه در قالب برش‌های عدد فازی) به کاربر ارائه می‌کند. از طرفی در بسیاری از اعمال حسابی پیچیده برای اعداد فازی LR ممکن است محاسبات و نتیجهٔ نهایی در ردهٔ اعداد فازی LR نگنجد. در این موارد می‌توان از برش‌های اعداد فازی برای پیش‌برد محاسبات به‌عنوان یک روش جایگزین استفاده نمود و از این موضوع می‌توان به‌عنوان یک نقطه‌ضعف برای بستهٔ Calculator.LR.FNs نسبت به بستهٔ FuzzyNumbers یاد کرد. علاقه‌مندان برای آشنایی بیشتر با بسته‌های FuzzyNumbers و Calculator.LR.FNs می‌توانند به [۴، ۵] مراجعه کنند، که حاوی فایل‌های راهنما با دستورالعمل‌ها و مثال‌های متعددی هستند.



شکل۵. نمودار توابع عضویت سه عدد فازی و همچنین میانگین، واریانس و انحراف معیار آن‌ها در مثال ۱۶.۳



شکل۱۰. نمودار توابع عضویت چندین عدد فازی و توابعی از آن‌ها در مثال ۹.۴

مراجع

[۱] طاهری، سید محمود؛ ماشین‌چی، ماشاالله (۱۳۸۷). مقدمه‌ای بر احتمال و آمار فازی، دانشگاه شهید باهنر کرمان.

[2] Coroianu, L., Gagolewski, M., and Grzegorzewski, P. (2013). Nearest piecewise linear approximation of fuzzy numbers, *Fuzzy Sets and Systems* **233**, 26-51.

[3] Coroianu, L., Gagolewski, M., Grzegorzewski, P. Adabitabar Firozja, M., and Houlari, T. (2014). Piecewise linear approximation of fuzzy numbers preserving the support and core. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Part II (CCIS 443), A. Laurent et al. (eds.), Springer, pp. 244-254.

[4] Gagolewski, M., and Caha, J. (2015). FuzzyNumbers Package: *Tools to Deal with Fuzzy Numbers in R*. R package version 0.4-1. http://FuzzyNumbers.rexamine.com/

[5] Parchami, A. (2016). Calculator.LR.FNs: *Calculator for LR Fuzzy Numbers*. R package version 1.1. https://CRAN.R-project.org/package=Calculator.LR.FNs