WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED

Model trees with topic model pre-processing: An approach for data journalism illustrated with the Wikileaks Afghanistan war logs

Rusch, Thomas; Hofmarcher, Paul; Hatzinger, Reinhold; Hornik, Kurt

Published in:
Annals of Applied Statistics

DOI:
10.1214/12-AOAS618

Published: 01/06/2013

Document Version
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Rusch, T., Hofmarcher, P., Hatzinger, R., & Hornik, K. (2013). Model trees with topic model pre-processing: An approach for data journalism illustrated with the Wikileaks Afghanistan war logs. *Annals of Applied Statistics*, *7*(2), 613 - 639. https://doi.org/10.1214/12-AOAS618

Download date: 08. Jun 2023

Model validation supplement to "Model trees with topic model pre-processing: An approach for data journalism illustrated with the WikiLeaks Afghanistan War Logs "

Thomas Rusch, Paul Hofmarcher, Reinhold Hatzinger and Kurt Hornik

January 8, 2013

In this supplement to the paper "Fatalities in the WikiLeaks Afghanistan War Logs: An approach for data journalism using model trees with topic model pre-processing" we explain in detail how we validated the model trees and thus expand on Section 5 in the main paper. We start with using a bootstrap resampling approach to make in-bag and out-of-bag predictions of observations and assess stability of the trees structure, the segmentation and report concordance based on a segment-wise Jaccard index. We then establish stability and reproducibility of the segment-wise parameter estimates. Lastly, we assess fit of the local models and show for each segment that the observations meet the requirements for assuming a negative binomial model to hold.

1 Tree Validation

With the model tree approach suggested in the main paper we build an exploratory model of fatalities in the overall WikiLeaks war diary. The model tree approach allows to use split variables to identify segments of the overall data to which a series of local models is fitted. Hence instead of validating the results globally (e.g., in terms of prediction or fit) the validation needs to be concerned with three aspects: How stable/reproducible is the tree structure (and hence the segmentation), how stable/reproducible are the segment-wise parameter estimates, and how well can the local models describe the observations in the segments (next section).

In this section we address validation of model trees by assessing stability and reproducibility of the tree structure and the segmentation, and the parameter estimates. We find that we have ten stable segments in the original tree which are reproducible both in terms of the assigned reports and the estimated parameter values in the segments (R_1 through R_6, R_8 through R_{10} and R_{15}). Among them are five segments that we described in detail in Section 4 of the main paper, associated with the topics "Task Force Reports (Bushmaster)", "Hostile Contacts ACF

vs TF", "Suicide and IED Bombing" and "Attacks (incl. IED) on Afghan and ISAF patrols". We also have three unstable segments (R_{11}, R_{14} and R_{13}) and two low to moderately stable segments (R_{12} and R_7). The latter five segments all have in common that they appear further down the tree hierarchy (see Figure 3 in the main paper), where for tree models less stability is not particularly surprising. These are the segments that in the original tree arise from the split based on `region` in node 6 and subsequent splits which lead to a segmentation of the reports in this branch that is only sometimes reproduced in the resampled data sets. Note that the reports associated with Topic 85 "Civilian Casualties" (segments R_7 and R_{12}) are among those.

1.1 Stability of Tree Structure and Segmentation

First we look at the segmentation and the tree structure. Since building palpable segments with a local model is the objective of our tree, we must first establish that the segment structure is not completely volatile (i.e., that we did not find segments that are not reproducible). Additionally, the assignment of reports to segments should be stable (i.e., the same report should be part of two corresponding segments most of the time). This is a similar rationale as used in the work on cluster validation by Hennig [2007] or Tibshirani and Walther [2005].

To investigate these two points, we choose to use data sets of smaller size than the overall data set, which were resampled with replacement and fit the tree to the new bootstrap data sets. For the out-of-bag observations (which are not part of the new data set), their segment membership is then predicted from the tree model. The rationale behind this is that if we can more or less reproduce the segment structure on the resampled data sets our segmentation can be considered stable. Additionally, the corresponding segments of the original tree and the new tree built on the resampled data set should contain as many of the same reports as possible (high concordance of the report assignment).

We use two resampling strategies to build the new data sets, a regular random resampling scheme (i.e., all observations are equally likely to be resampled; we call these data sets regularly resampled sets or RRS) as well as stratified random resampling (we resample observations from each segment proportional to the segment size and from each segment all observations are equally likely to be resampled; we call them stratified resampled sets or SRS). Both times resampling was with replacement (a bootstrap sample). The new data sets are 5/6 of the size of the original data set. For a resampled data set, a tree with $\alpha = 1 \times 10^{-3}$ and minsplit=250 is grown which is then used to predict the segment membership of the observations from the training set as well as the out-of-bag observations. We therefore get a segmentation of all observations for each data set. We do this 200 times for RRS and 200 times for SRS.

Since we are interested in the segmentation and a local validation, we need to calculate a concordance index based on the number of reports in the segments from the resampled data that are also in the corresponding segment in the original data. Please note that we need a segment-wise local measure of concordance. We therefore first have to choose which segment from a resampled tree corresponds to the given segment from the original tree. We do this by assigning the segment from a tree on a bootstrap sample to a given segment from the original

tree for which there is the highest segment-specific measure of concordance between the two segments. Second we assess how strong the concordance is.

Segment-specific concordance of reports We need a concordance measure to automatically choose which segments correspond to the original tree segments. Following, Hennig [2007] we use a segment-wise Jaccard index that is defined as follows: Let T denote the original tree and $T^{(b)}$ the tree fitted on bootstrap sample b, $b = 1, \ldots, 200$ with T having the segments $R_k, k = 1, \ldots, r$ and $T^{(b)}$ the segments $R_l^{(b)}, l = 1, \ldots, r^{(b)}$. Let S be the set of observations $\{s_1, \ldots, s_n\}$. We have two segmentations $R = \{R_1, \ldots, R_r\}$ and $R^b = \{R_1^b, \ldots, R_{r^{(b)}}^b\}$, the first stemming from the original tree T and the latter from the tree $T^{(b)}$. Here, R is the partition with the segments from our original tree (so $r = 15$) and R^b is the partition obtained from the tree grown on a given, single resample b (with $r^{(b)}$ being not necessarily equal to r).

For our concordance measure we set up a contingency table of the partitions with the counts n_{kl} in the cells, which is the number of observations that fall into the combination or cell (R_k, R_l^b).

	R_1^b	$\ldots$	$R_{r^b}^b$	
R_1	n_{11}	$\ldots$	n_{1r^b}	n_{1+}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	n_{k+}
R_r	n_{r1}	$\ldots$	n_{rr^b}	n_{r+}
	n_{+1}	n_{+l}	n_{+r^b}	n

Based on this contingency table, different indices for concordance can be calculated (e.g., the Rand index, the adjusted Rand index, the prediction strength, the dice index or may others) for the whole table or for a contingency table derived by collapsing cells. We need a segment-specific index, so we use a segment-wise version of the Jaccard index [Jaccard, 1901] labeled by Jac_{kl}:

For each combination (R_k, R_l^b) we create a 2×2 contingency table. This 2×2 table looks like

		R_l^b: $+$	R_l^b: $-$	
R_k	$+$	$a := n_{kl}$	$c := \sum_{o \neq l} n_{ko}$	n_{k+}
	$-$	$b := \sum_{p \neq k} n_{pl}$	$d := \sum_{p \neq k, o \neq l} n_{po}$	n_{k-}
		n_{+l}	n_{-l}	n

We can now calculate the segment-specific Jaccard index as

$$Jac_{kl} = \frac{a}{a+b+c} = \frac{n_{kl}}{n_{+l} + n_{k+} - n_{kl}} \tag{1}$$

but in principle any other index employing a combination of the a, b, c, d and their marginals could be used as well.

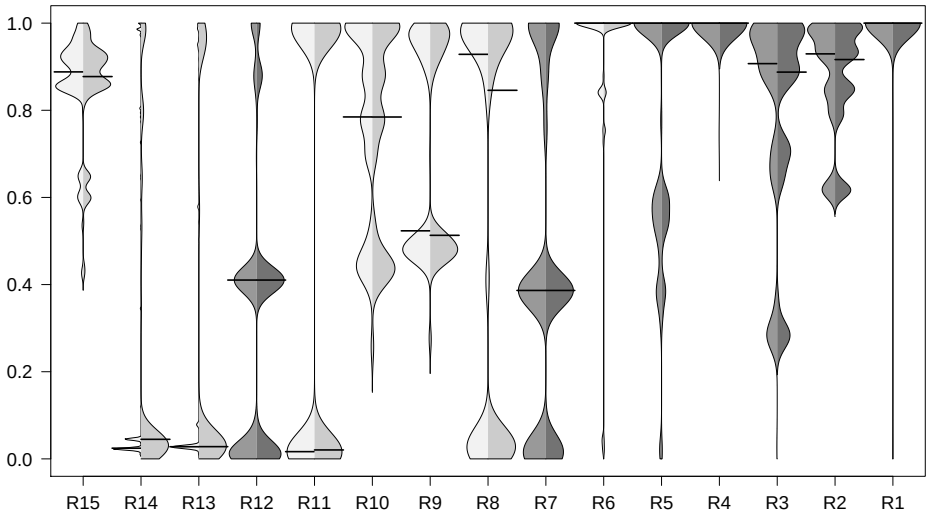

Figure 1: Bean plots of the segment-wise Jaccard indices, $Jac_k^{*(b)}$, between the original segmentation and the segmentations for predictions over the bootstrap samples b for all $R_k,\ k = 1, \ldots, 15$. Darker beans mark segments we described in detail in the paper. The left part of each bean is for RRS (slightly lighter shaded) and the right side for the SRS (slightly darker shaded).

We denote the segment-wise Jaccard index for each resample b by $Jac_{kl}^{(b)}$ with $k = 1, \ldots, r$ and $l = 1, \ldots, r^{(b)}$. For each resample b we calculate the segment-wise indices $Jac_{kl}^{(b)}$ and then, for a given R_k, assign the segment with the highest index over all l from the resample tree to be the corresponding segment of R_k, i.e., the segment $R_l^{(b)},\ l : \arg\max_l Jac_{kl}^{(b)}$ with concordance $Jac_k^{*(b)} = \max_l Jac_{kl}^{(b)}$

The segment-wise five point summary of the Jaccard indices over all corresponding segments from the 200 resamples can be found in Table 1. It also lists the mean and the relative frequencies of indices being 1 ("coincidence"), higher than 0.8 ("strong correspondence"), less than 0.5 and less than 0.25. A summary of the last line (the means over all segments) can be found in Table 2.

In Figure 1 we plotted for each segment a bean plot of the segment-wise Jaccard index over all corresponding segments from the resamples. Bean plots are similar to boxplots, but use kernel density estimators to draw the border of the beans. However, they allow asymmetric sides of the bean, which we use to display the density estimate for RRS and SRS next to each other. They are particularly useful if the distribution is not unimodal (as we often find here). The sides of the beans are therefore to be interpreted as the kernel density estimate for a given segment with the left side (slightly lighter shaded) being the distribution of the segment-wise index for RRS and with the right side (slightly darker shaded) being the distribution of the segment-wise index for SRS.

In terms of relative frequencies of corresponding tree segments exceeding a certain threshold

Table 1: Summary statistics for the segment-wise Jaccard indices, $Jac_k^{*(b)}$, between the original segmentation and the segmentations for predictions over all bootstrap samples b. The upper portion displays the regularly resampled bootstrap samples (RRS) and the lower portion stratified randomly resampled bootstrap samples (SRS). The rows list the original segments R_1 through R_{15} as in the tree plot from right to left.

RRS										
							Rel freq. $Jacc_k^{*(b)}$			
Segment	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	1	$> .8$	$< .5$	$< .25$
R_1	0.02	1.00	1.00	0.99	1.00	1.00	0.99	0.99	0.01	0.01
R_2	0.62	0.82	0.93	0.88	0.98	1.00	0.11	0.78	0.00	0.00
R_3	0.03	0.66	0.91	0.79	0.98	1.00	0.15	0.67	0.17	0.01
R_4	0.74	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.00	0.00
R_5	0.02	0.54	1.00	0.74	1.00	1.00	0.51	0.51	0.18	0.04
R_6	0.02	0.84	1.00	0.90	1.00	1.00	0.73	0.90	0.07	0.07
R_7	0.00	0.02	0.39	0.45	0.89	1.00	0.11	0.29	0.68	0.31
R_8	0.01	0.03	0.93	0.59	0.99	1.00	0.10	0.58	0.42	0.38
R_9	0.27	0.48	0.52	0.71	0.97	1.00	0.01	0.48	0.38	0.00
R_{10}	0.26	0.49	0.79	0.77	0.99	1.00	0.02	0.49	0.26	0.00
R_{11}	0.01	0.02	0.02	0.46	0.98	1.00	0.15	0.46	0.54	0.54
R_{12}	0.00	0.01	0.41	0.30	0.41	1.00	0.07	0.14	0.85	0.48
R_{13}	0.01	0.03	0.03	0.20	0.08	1.00	0.01	0.15	0.81	0.81
R_{14}	0.01	0.02	0.02	0.20	0.05	1.00	0.04	0.14	0.81	0.80
R_{15}	0.42	0.85	0.89	0.86	0.93	1.00	0.01	0.85	0.01	0.00
$r^{-1}\sum Jac_k^{*(b)}$	0.16	0.45	0.66	0.66	0.82	1.00	0.27	0.56	0.35	0.23
SRS										
							Rel freq. $Jacc_k^{*(b)}$			
Segment	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	1	$> .8$	$< .5$	$< .25$
R_1	0.02	1.00	1.00	0.99	1.00	1.00	0.99	0.99	0.01	0.01
R_2	0.62	0.83	0.92	0.87	0.97	1.00	0.14	0.77	0.00	0.00
R_3	0.28	0.70	0.89	0.79	0.98	1.00	0.17	0.62	0.15	0.00
R_4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00
R_5	0.02	0.54	1.00	0.78	1.00	1.00	0.56	0.56	0.13	0.01
R_6	0.02	1.00	1.00	0.94	1.00	1.00	0.82	0.91	0.03	0.03
R_7	0.00	0.02	0.39	0.44	0.88	1.00	0.14	0.26	0.72	0.28
R_8	0.01	0.03	0.85	0.52	0.98	1.00	0.10	0.52	0.48	0.47
R_9	0.27	0.47	0.51	0.71	0.97	1.00	0.05	0.47	0.36	0.00
R_{10}	0.26	0.44	0.79	0.74	0.99	1.00	0.05	0.47	0.31	0.00
R_{11}	0.01	0.02	0.02	0.50	0.99	1.00	0.13	0.49	0.51	0.51
R_{12}	0.00	0.01	0.41	0.33	0.41	1.00	0.04	0.14	0.84	0.41
R_{13}	0.01	0.03	0.03	0.27	0.58	1.00	0.04	0.24	0.73	0.73
R_{14}	0.01	0.02	0.04	0.24	0.49	1.00	0.03	0.15	0.75	0.73
R_{15}	0.42	0.85	0.88	0.85	0.92	1.00	0.01	0.83	0.02	0.00
$r^{-1}\sum Jac_k^{*(b)}$	0.20	0.46	0.65	0.66	0.88	1.00	0.29	0.56	0.34	0.21

Table 2: Summary of tree structure validation for the resampled data sets (RRS 1 through RRS 5 for regularly resampled sets, SRS 1 through 5 for stratified resampled sets). The rows list the relative frequencies of the segments with a report concordance over .8, relative frequencies of the segments with a report concordance of 1, the relative frequency of coinciding segments and the relative frequency of corresponding segments.

		RRS	SRS	Overall
All	$Jac_k^{*(b)} < .25$	.23	.21	.22
Segments	$Jac_k^{*(b)} < .5$	.35	.34	.34
	$Jac_k^{*(b)} > .8$	.56	.56	.56
	$Jac_k^{*(b)} = 1$	.27	.29	.28
Described	$Jac_k^{*(b)} < .25$	.12	.10	.11
Segments	$Jac_k^{*(b)} < .5$	.27	.27	.27
	$Jac_k^{*(b)} > .8$	.63	.62	.62
	$Jac_k^{*(b)} = 1$	.42	.44	.43

of the Jaccard Index (see Table 1 and Table 2), we find that for the regularly resampled data sets only there are 26.5% of the segments that coincide averaged over all 200 resamples. 56.2% of the segments were strongly corresponding. Coinciding means that $Jac_k^{*(b)} = 1$ and strong correspondence means $Jac_k^{*(b)} \geq 0.8$. The values are more or less the same for stratified resampling where averaged over all resampled data sets there are 28.6% of the segments coinciding and 56.3% strongly corresponding. Pooled this means there are 27.6% of the segments coinciding and 56.3% strongly corresponding to segments of the original tree.

This is even more pronounced for segments that we described in detail in the paper (i.e., segments R_1, R_2, R_3, R_4, R_5, R_7 and R_{12}). Here we have 41.8% segments coinciding over the simple random samples, and 43.7% of segments coinciding over the stratified samples. Overall this means 42.3% coinciding segments. Strong correspondence for this subset of segments is given in 62.6% of the segments for RRS, and 62.2% for the SRS. Overall 62.4% of the segment resampled segments show strong correspondence. Note that the first five described segments, R_1 through R_5 show even higher frequencies (56.2% coinciding and 79% strongly corresponding). Hence the described segments can be considered to be highly stable with perhaps the exception of the segments associated with Topic 85 (R_7 and R_{12}) which have for RRS and SRS a percentage of 9% and 9.3% coinciding segments and 21.8% and 20% strongly corresponding segments.

The distribution of the concordance measure over the bootstrap samples is summarized by quantiles and mean in Table 1 and with beanplots in Figure 1. While the median and mean values should be interpreted with caution as the distributions often have more than a single mode and have a number of outliers, we see that for most segments (9 of 15) the median segment-wise Jaccard index is 0.79 or higher. For the described segments this is even stronger, with a median 0.79 or higher in 5 of 7 segments. For the segments R_1 through R_5 as well as R_6,

R_8, R_{10} and R_{15} we have moderate to high median and mean segment-wise concordance with segments from resampled data sets but with considerable variability measured by interquartile range. The visualisation of the overall distributions as in Figure 1 allows to identify the more stable and less stable segments as well, probably better than the median and mean and alone as it also allows to gauge the multimodality and the outliers of the concordance measure distribution. However, the conclusions are very similar to using only the median with the caveat that there sometimes is substantial variability and outliers or multimodality in the Jaccard index for the segments over the bootstrap samples. Hence the uncertainty in the Jaccard index can be high, particularly for R_{12}, R_{10}, R_9 and R_5. Generally low stability we have for segments R_{14}, R_{13}, R_{11}. Also, R_{12} and R_7 are not particularly stable.

Segment-wise Variability of Fatality Rates Additional to the overall and segment-wise stability of the tree based on the concordance measure, we investigate the variability of the estimates of the model parameters for each segment. To achieve this we match a given segment R_k from T with a segment $R_l^{(b)}$ from $T^{(b)}$ based on the highest Jaccard index for each stratified or regular bootstrap sample as before. Figure 2 displays bean plots of the distributions of the log estimated death toll $\log(\hat{\mu}_l)$ over the matched segments and the shape parameter $\hat{\theta}_l$ over the bootstrap samples for each segment. The dotted horizontal lines indicate the parameter values estimated for the original tree. What we find is that on the one hand, for segments R_1, R_2, R_4 and R_5 as well as for segments R_6, R_8, R_9, R_{10} and R_{15}, the segment-wise estimates $\log(\hat{\mu}_k)$ of the original tree turn out to lie close to the medians of the estimates from the corresponding segments of the resampled data sets. Additionally, for $R_1, R_2, R_4, R_5, R_6, R_9$ and R_{15} the variability of the estimated parameters $\log(\hat{\mu}_l)$ over the bootstrap samples is rather small. These are the same segments that we could identify as being stable in the previous section. On the other hand, the low stability segments R_{14}, R_{13}, R_{11} as well as R_{12} often display a multimodal distribution of $\log(\hat{\mu}_l)$ and have a median that is considerably different from the estimated value of the original tree. Furthermore they display much higher variability over the bootstrap samples, especially segments R_{12} and R_7. For the estimated shape parameters, the picture is very similar.

This corroborates the results on segmentation stability based on the segment-wise Jaccard index presented before: We have ten stable segments of the original tree which are reproducible both in terms of the assigned reports and the estimated parameter values in the segments (R_1 through R_6, R_8 through R_{10} and R_{15}). Among them are five segments that we described in detail in Section 4.1. in the main paper, associated with the topics "Task Force Reports (Bushmaster)", "Hostile Contacts ACF vs TF", "Suicide and IED Bombing" and "Attacks (incl. IED) on Afghan and ISAF patrols". We also have three unstable segments (R_{11}, R_{14} and R_{13}) and two low to moderately stable segments (R_{12} and R_7).

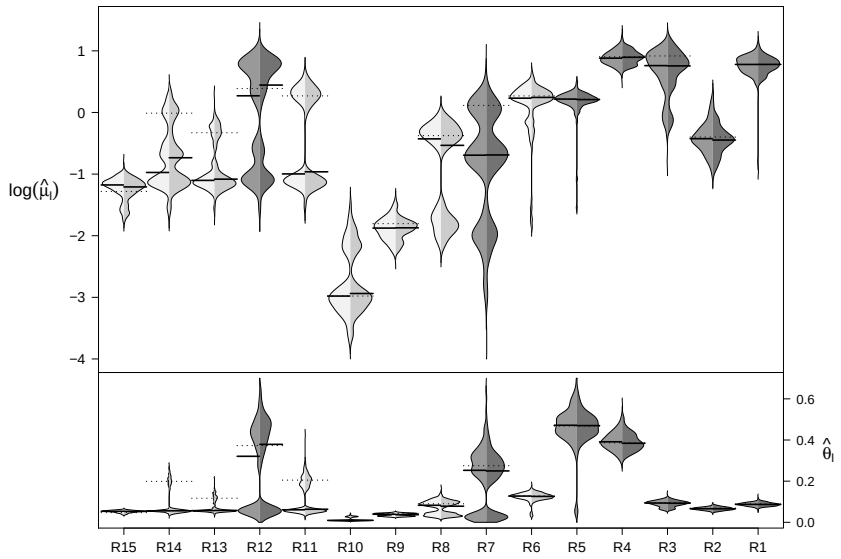

Figure 2: Bean plots of the segment-wise estimated death toll parameter $\log(\hat{\mu}_l)$ and shape parameter $\hat{\theta}_l$ over the bootstrap samples. The dotted horizontal lines indicate the values of the original tree (compare Table 2 in the paper). Again, darker beans mark segments described in detail in the paper. The left part of each bean is for RRS (slightly lighter shaded) and the right side for the SRS (slightly darker shaded).

2 Node Model Validation

In this section we investigate the appropriateness of the node model in more detail. This is important because our approach is designed to deliver segments with *local* models. We start with judging the goodness of fit of the local models in the nodes. Then—since we used a negative binomial model—we analyze the appropriateness by checking assumptions and properties of the underlying negative binomial distributions, like overdispersion of the count in a Poisson model, independence of the residuals, and no excess zeros.

2.1 Fit assessment

We report the deviance and the degrees of freedom in Table 2 of the main paper. The deviance, its ratio to the degrees of freedom and/or the segment-wise mean absolute prediction errors, and the form of the negative binomial distribution in each segment all point to a good fit.

In Figures 3 and 4 we show the observed frequencies of each count (grey bar) and the fitted negative binomial probability distribution (red dots) for all segments (the y-axis has been square root transformed for better readability of smaller probabilities). Note that here we work with the node numbers of the tree rather than the segment labeling from the paper, but the mapping between node numbers and our segment labeling is obvious since we labeled them in a decreasing order. So nodes 29, 28, 27, 25, 24, 23, 22, 21, 20, 19, 15, 14, 13, 12, 11 are R_1 through R_{15} (hence 29 is R_1 and 11 is R_{15}). We clearly see a good approximation of

the expected frequencies to the observed frequencies.

The ratio of deviance to degrees of freedom for each segment never exceeds 1, and can be substantially smaller than one. We therefore sometimes find underdispersion in the segments, or less variability as expected under the negative binomial model. This is effectively due to a usually large percentages of reports with zero fatalities and occasionally very large outliers. This is especially pronounced in segments R_{10} (where we find 98.2% zero fatalities but a maximum death toll of 67), and also R_9 and R_{15}. We believe this is no problem for our exposition though, as the only practical influence of underdisperison is on the standard errors where it will lead to them being too conservatively estimated and thus take effect if used for inference (which we do not). Additionally, the lower the variability, the better the predictions from our intercept-only model will be. For example for R_{10} the mean absolute prediction error (mape) is 0.1, for R_9 it is 0.308 and for R_{15} it is 0.507. As a third point, the interesting segments that we focused on in Section 4.1 in the main paper usually show relatively moderate underdispersion phenomena (again due to the high number of reports with zero deaths and some extreme fatality numbers throughout).

2.2 Overdispersion

We check whether there is overdispersion relative to a Poisson model in our count data for the overall data set by using the standard dispersion test of Cameron and Trivedi [1990] as implemented in the R package AER [Kleiber and Zeileis, 2008].

We test for overdispersion for all 15 segments yielded by the tree algorithm. The first entry in each list element is the estimated disperson parameter α (for overdispersion $\alpha > 0$), the second the test statistic used to test for $H_0 : \alpha = 0$ (which is asymptotically normal) and third the corresponding p-value. We see that relative to the Poisson distribution there is substantial overdispersion in every segment and hence a Poisson model would be inappropriate. A count data model that allows to have extra Poisson variation is necessary to model the counts in each segment appropriately. This leaves the negative binomial or its zero-inflated counterpart. However, the zero-inflated model does not fit the data better than the simple negative binomial model (see Section 2.3 and in terms of parsimony, the latter is to be preferred.

```
$`Node 29`
  dispersion            z
2.764138e+01 3.391617e+00 3.474074e-04

$`Node 28`
  dispersion            z
12.671942465  2.565556263  0.005150526

$`Node 27`
  dispersion            z
43.067475131  2.402623846  0.008138961
```

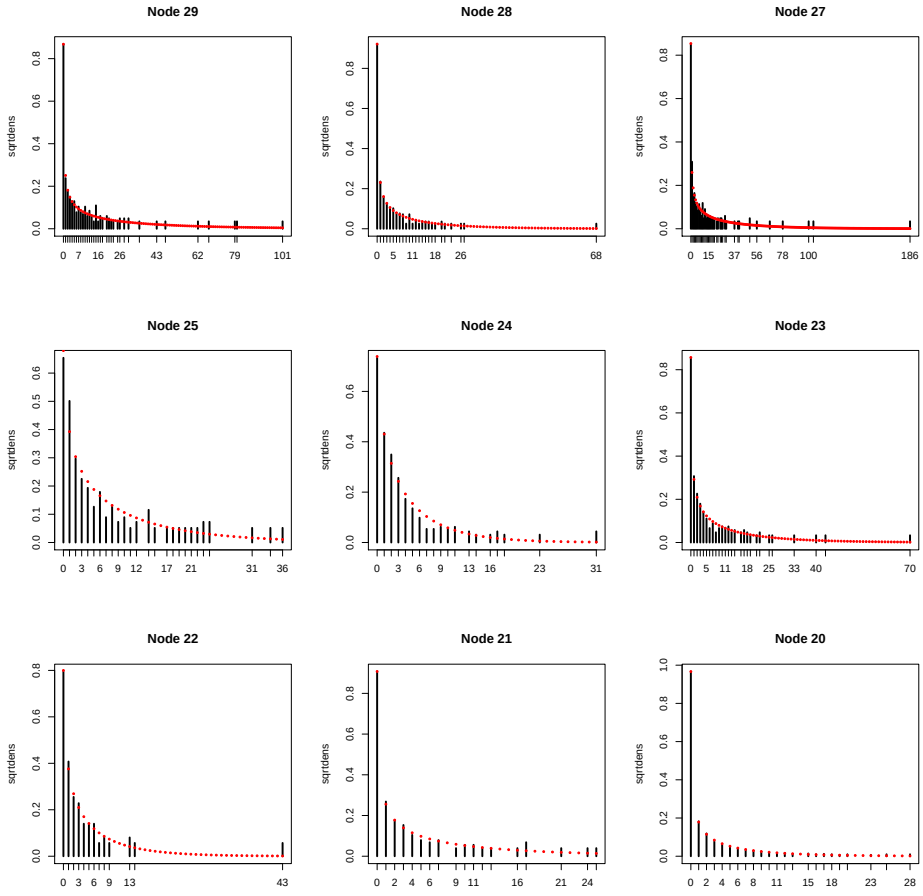

Figure 3: Observed relative frequency of counts (grey bar) vs. expected relative frequency (red dot) from the fitted negative binomial distribution for various segments (all frequencies are on the square root scale).

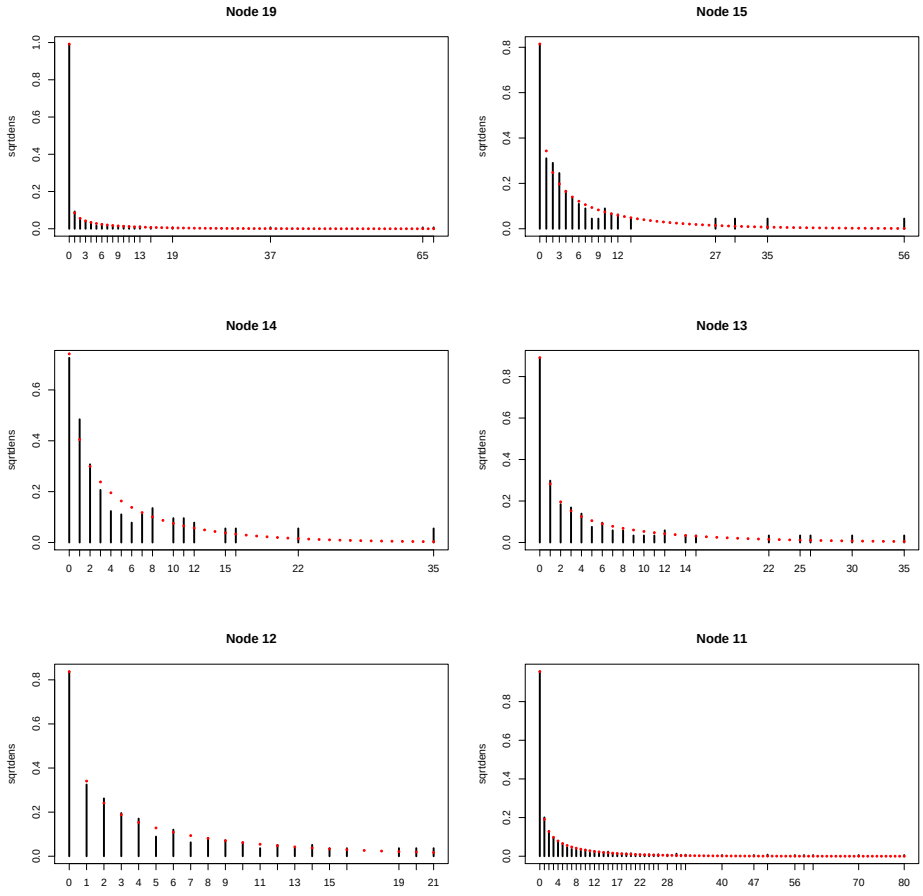

Figure 4: Observed relative frequency of counts (grey bar) and expected relative frequency (red dot) from the fitted negative binomial distribution for various segments (the frequencies are on the square root scale).

```
$`Node 25`
  dispersion            z
1.022638e+01 4.354504e+00 6.668417e-06

$`Node 24`
  dispersion            z
5.1977191180 3.8216796579 0.0000662729

$`Node 23`
  dispersion            z
14.988224863  3.057318629  0.001116634

$`Node 22`
dispersion          z
8.69924627 1.53436592 0.06246983

$`Node 21`
  dispersion            z
9.3089134138 3.5824837245 0.0001701714

$`Node 20`
  dispersion            z
5.711488e+00 8.030757e+00 4.843636e-16

$`Node 19`
 dispersion           z
15.27271657  2.10877372  0.01748206

$`Node 15`
dispersion          z
11.8932170  2.1553507  0.0155672

$`Node 14`
 dispersion           z
7.408687764 2.577936171 0.004969618

$`Node 13`
  dispersion            z
9.5839797007 3.1480875012 0.0008217125

$`Node 12`
  dispersion            z
5.941142e+00 4.968938e+00 3.366033e-07
```

```
$`Node 11`
  dispersion            z
1.062005e+01 7.069658e+00 7.765802e-13
```

2.3 Segment-wise Comparison of Negative Binomial Distribution vs. Alternatives

Here we fit competing intercept-only count data models to the segments resulting from the tree and compare them. This will allow to judge whether the negative binomial is appropriate to model the data in each segment or whether a Poisson distribution or models with excess zeros (zero-inflated Poisson distribution or a zero-inflated negative binomial distribution) are better. The fit will be judged by AIC, BIC and the log-likelihood. A higher log-likelihood and a lower AIC, BIC are considered to be better.

We extracted all observations from each terminal node and compare the fit of the four distributions with AIC, BIC and log-likelihood. We see that in all the segments the negative binomial has the lowest AIC and BIC and often highest log-likelihood on par with the zero-inflated negative binomial distribution and hence is the most appropriate node model distribution out of these four.

```
> mod_compare(segs[1],afg1, contr = zeroinfl.control(method="BFGS", EM=TRUE)) #AIC nb

$Node
[1] 29

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     8247.118      4100.414      2154.140      2152.654

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     8251.840      4109.857      2168.304      2162.097

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -4122.559     -2048.207     -1074.070     -1074.327

> mod_compare(segs[2],afg1) #AIC nb

$Node
[1] 28

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     5692.763      3115.713      2361.179      2359.179
```

```
$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     5698.097      3126.380      2377.180      2369.846

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -2845.382     -1555.856     -1177.590     -1177.589
```

```
> mod_compare(segs[3],afg1) #AIC nb
```

```
$Node
[1] 27

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    10099.942      5629.634      2344.601      2342.601

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    10104.686      5639.122      2358.833      2352.089

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -5048.971     -2812.817     -1169.301     -1169.300
```

```
> mod_compare(segs[4],afg1) #AIC nb
```

```
$Node
[1] 25

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2780.847      2255.875      1469.959      1467.957

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2784.771      2263.723      1481.732      1475.806

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -1389.4234    -1125.9373     -731.9794     -731.9786
```

```
> mod_compare(segs[5],afg1) #AIC nb
```

```
$Node
[1] 24
```

```
$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     4212.173      3588.835      3097.349      3095.348

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     4217.112      3598.713      3112.166      3105.227

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -2105.086     -1792.417     -1545.674     -1545.674
```

```
> mod_compare(segs[6],afg1) #AIC nb
```

```
$Node
[1] 23

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     5326.201      3277.758      2183.101      2181.100

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     5331.003      3287.362      2197.508      2190.705

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -2662.101     -1636.879     -1088.550     -1088.550
```

```
> mod_compare(segs[7],afg1) #AIC nb
```

```
$Node
[1] 22

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    1340.0392     1046.3569      830.2033      828.2028

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    1343.7660     1053.8106      841.3838      835.6565

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -669.0196     -521.1785     -412.1016     -412.1014
```

```
> mod_compare(segs[8],afg1) #AIC nb

$Node
[1] 21

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2299.249      1375.872      1077.720      1075.719

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2303.707      1384.789      1091.095      1084.636

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -1148.6246     -685.9362     -535.8600     -535.8597

> mod_compare(segs[9],afg1) #AIC

$Node
[1] 20

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     23199.61      14963.67      13568.39      13566.33

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     23207.48      14979.41      13592.01      13582.08

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -11598.805     -7479.833     -6781.194     -6781.167

> mod_compare(segs[10],afg1) #AIC nb

$Node
[1] 19

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     9605.126      5078.089      4122.032      4120.031

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     9612.931      5093.698      4145.446      4135.640
```

```
$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -4801.563     -2537.044     -2058.016     -2058.016
```

```
> mod_compare(segs[11],afg1) #AIC nb
```

```
$Node
[1] 15

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2618.085      1832.500      1368.118      1366.117

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     2622.296      1840.922      1380.750      1374.538

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -1308.0426     -914.2502     -681.0592     -681.0584
```

```
> mod_compare(segs[12],afg1) #AIC nb
```

```
$Node
[1] 14

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     1652.324      1367.303      1041.056      1039.055

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     1656.117      1374.889      1052.435      1046.641

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -825.1618     -681.6513     -517.5278     -517.5276
```

```
> mod_compare(segs[13],afg1) #AIC nb
```

```
$Node
[1] 13

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     3174.684      2071.572      1624.826      1622.823
```

```
$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     3179.462      2081.127      1639.159      1632.378

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -1586.3422    -1033.7858     -809.4129     -809.4116
```

```
> mod_compare(segs[14],afg1) #AIC nb
```

```
$Node
[1] 12

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     3016.267      2112.480      1834.652      1832.919

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     3020.896      2121.739      1848.540      1842.178

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
   -1507.1333    -1054.2402     -914.3260     -914.4596
```

```
> mod_compare(segs[15],afg1) #AIC nb
```

```
$Node
[1] 11

$AIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     57227.66      33942.34      27566.68      27564.67

$BIC
      poisson zeroinfl_pois   zeroinfl_nb        negbin
     57235.97      33958.97      27591.62      27581.30

$logLik
      poisson zeroinfl_pois   zeroinfl_nb        negbin
    -28612.83     -16969.17     -13780.34     -13780.34
```

2.4 Temporal Dependence

As the war log reports are actually a highly irregular time series, there might be some temporal dependence between reports filed after each other. Here we check whether there is a temporal dependence between subsequent observations or whether the assumption of independence of the fatality counts can be upheld. The tree algorithm and the usage of the explanatory variables automatically breaks up some of the possible autocorrelation of the observations, but we wanted to make sure that there is no substantial time series component that we overlooked or that might interfere with the negative binomial assumption. For this we order the observed counts in each segment from least to most early and calculate autocorrelograms for the deviance residuals and the Durbin Watson Statistic.

We find that usually (10 of 15 times) there is no autocorrelation to be found. Nodes 20 and 11 show signs of autocorrelation in the DW test, but the test should be interpreted with caution because of the high number of observations in the segments. For these segment, the autocorrelogram does not point to dependence as the first autocorrelation is virtually zero. In four nodes (28, 27, 25, 21) there might be a possible AR(1) process judging from the DW statistic and test as well as the autocorrelogram in Figure 2.4. However, if there were an AR(1) process in the residuals, the autocorrelation is quite small with less than 0.1. In node 29 —the Task Force Reports (Bushmaster) segment— there seems to be a substantial time component ($0.1 < r < 0.2$), although based on the DW statistic there is not enough evidence to conclude it is an AR(1) process), see Figure 2.4. This may be due to the nature of the reports in this segment, as a number of crucial NATO operations are collected within this segment. Additional modeling of the temporal structure might prove to be prudent for these reports. Overall though, we believe that the assumption of near temporal independence of the reports in each segment can be upheld or at least serves as a good enough approximation for all segments.

The following list displays the Durbin-Watson test statistic (first entry) and the p-value of the test for positive autocorrelation (second entry).

```
$`Segment 29`
        DW
1.91081211 0.09917113

$`Segment 28`
          DW
1.660048e+00 1.414811e-11

$`Segment 27`
        DW
1.84466641 0.01173509

$`Segment 25`
        DW
```

```
1.78618741 0.01908557

$`Segment 24`
      DW
1.960474 0.262551

$`Segment 23`
       DW
1.9680418 0.3156471

$`Segment 22`
       DW
1.8893644 0.1654225

$`Segment 21`
         DW
1.801925945 0.006114441

$`Segment 20`
         DW
1.962241017 0.004256304

$`Segment 19`
       DW
1.9986959 0.4650316

$`Segment 15`
       DW
1.9897148 0.4542256

$`Segment 14`
       DW
1.9270429 0.2537695

$`Segment 13`
      DW
1.919408 0.115971

$`Segment 12`
       DW
1.9600372 0.2909925

$`Segment 11`
```

```
         DW
1.972412782 0.008337933
```

References

Cameron, A. and Trivedi, P. (1990). Regression-based tests for overdispersion in the poisson model. *Journal of Econometrics*, 46:347–364.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis*, 52:258–271.

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines [distribution of alpine flora in the dranse basin and several neighboring regions]. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:241–272.

Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*. Springer-Verlag, New York. ISBN 978-0-387-77316-2.

Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14:511–528.

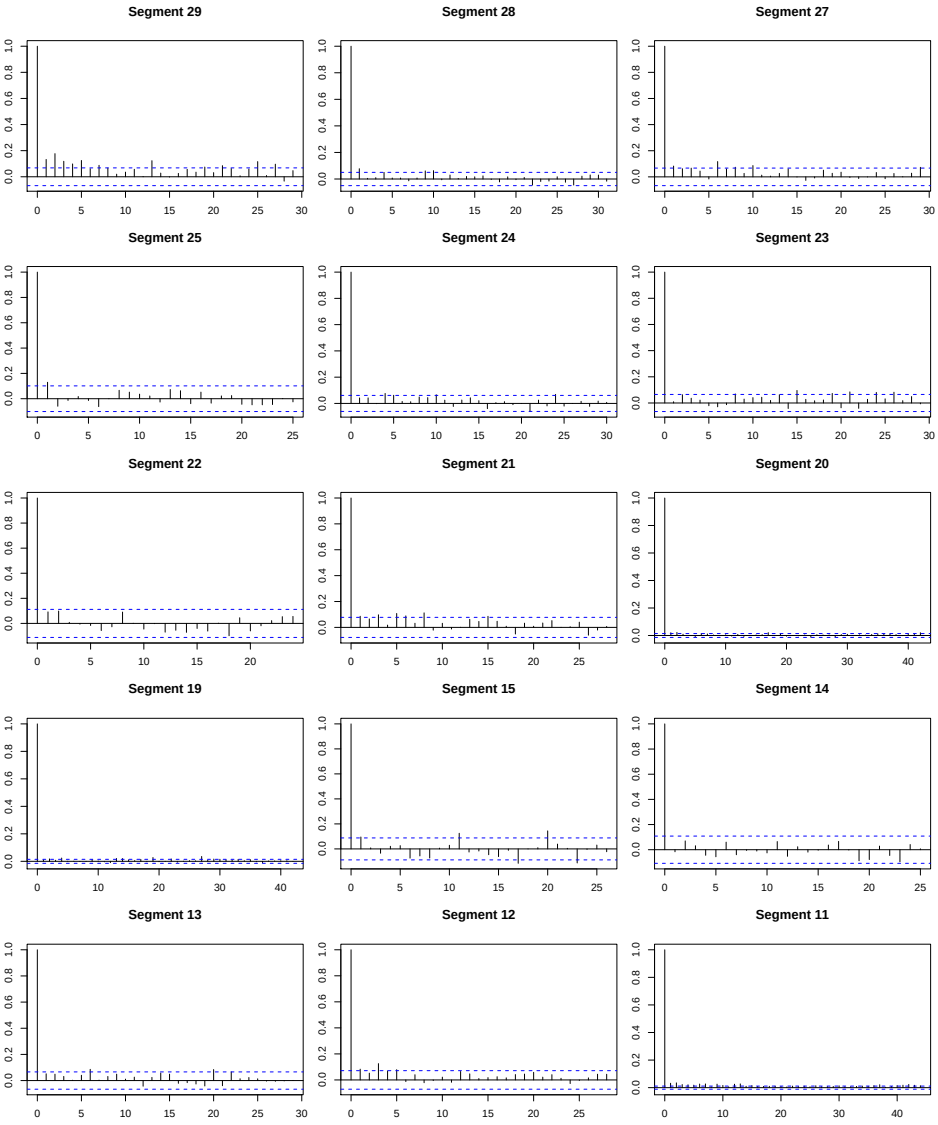

Figure 5: Segment-Wise autocorrelogram of the deviance residuals from a fitted negative binomial distribution.