Comparative analysis of linguistic features and machine learning methods in the task of assessing the complexity of texts

Artem Zaikin[1*], Valery Solovyev[1], Marina Solnyshkina[1]

[1] Kazan Federal University, Kazan, Russian Federation
[*] Corresponding author's email: kaskrin@gmail.com

DOI: https://doi.org/10.53482/2026_60_432

ABSTRACT

Text complexity assessment is an important applied problem that lacks a comprehensive solution. Studies vary in the text corpora used, the features analyzed, the analysis algorithms applied, and the assessment methods employed. We present a new methodology for complexity assessment, developed based on a representative collection of Russian-language school textbooks compiled by the authors. Textual complexity is represented numerically by a textbook's target class grade. First, we evaluate the average number of words per sentence and the average number of syllables per word for their ability to predict text complexity and implement machine learning methods, including linear regression and neural networks. We confirm the linear Flesch-Kincaid complexity formula in the sense that it cannot be improved upon. The influence of text segmentation on the assessment results is also examined, and several approaches to utilizing such information are presented in the paper. This approach gave us approximately 10% improvement in the R^2 metric compared to the baseline model. We also examine a total of 47 linguistic parameters as predictors of complexity and evaluate their significance.

Keywords: text complexity, linguistic features, regression, machine learning, school textbooks

1 Introduction

The need for an objective assessment of text complexity to ensure adequate reader understanding was recognized over 80 years ago. The first mathematical readability formula, known as the Flesch Readability Formula, was proposed for English in Flesch (1948). It was later refined into the Flesch–Kincaid Grade Level formula (Kincaid et al., 1975). The formula is expressed as $FKG(ASL, ASW) = 0.39ASL + 11.8ASW - 15.59$, where ASL denotes the average sentence length (in words) and ASW represents the average word length (in syllables). The resulting numerical score approximately corresponds to the school grade level required to understand the text. The parameters ASL and ASW were not chosen arbitrarily; they are significant predictors of complexity and have since been utilized in various formulas and approaches. Subsequently, similar formulas were proposed for other languages, incorporating additional linguistic parameters. For example, Mikk (1974) analyzed Estonian texts and introduced a new parameter: the degree of word abstractness.

Naturally, each language requires its own readability formulas. In this article, we study the complexity of texts in Russian. For Russian, the first formula was proposed by Matskovskiy (1976): $X_1 = 0.62X_2 + 0.123X_3 + 0.051$, where X_2 is the average sentence length (in words) and X_3 is the percentage of words with more than three syllables. Specialized formulas for different genres and subject areas were also proposed. Shpakovskiy et al. (2007) proposed a formula for chemical texts that accounts for specialized terms and symbols. Oborneva (2006) proposed the formula $FKG_{Rus}(ASL, ASW) = 0.5ASL + 8.4ASW - 15.59$ using the same variables. This formula is well-suited for fiction, as it was derived from the English Flesch–Kincaid formula by adapting the coefficients to account for linguistic differences, based on a corpus of fiction texts. Furthermore, Solovyev et al. (2018) demonstrated that this formula yields unreasonably high scores for school textbooks and proposed a genre-specific formula: $FKG_{sis}(ASL, ASW) = 0.36ASL + 5.76ASW - 11.97$. The mean squared error is approximately 1.02, corresponding to one grade level. A limitation of this work was that the formula was derived from a small collection of only 14 textbooks for grades 5–11, all within a single subject area: social studies. Later attempts were made to improve this formula Solovyev et al. (2022) while maintaining a linear model. Solnyshkina et al. (2018) made the only known attempt to derive a quadratic formula. These works achieved a slight increase in accuracy compared to the simple FKG_{sis} formula, though the improvement was insignificant. Morozov et al. (2022) considered a coarser classification into five grade groups: 1–2, 3–4, 5–7, 8–9, and 10–11. An F-measure of 80% was achieved for fiction texts.

One independent area of research focuses on assessing the complexity of individual words. This field is known as Complex Word Identification or lexical complexity prediction (Yimam et al., 2018). The complexity of word combinations has also been studied (Feng and Yu, 2024). For Russian, several studies have employed neural network approaches in this domain (Abramov and Ivanov, 2022; Abramov et al., 2023). The results are comparable to those for English, though slightly lower.

A number of studies have focused on texts for teaching Russian as a foreign language. Using the generally accepted division of instructional texts into six difficulty levels, researchers have achieved 60% accuracy in complexity assessment (Aleksandrovich, 2022). Corlatescu et al. (2022) propose limiting the classification to only two difficulty levels; with this binary approach, accuracy reaches 92%. Almost all studies employ either linear models with multiple features or neural architectures such as BERT. Significant variations across studies in terms of text collections, feature sets, and analysis methods make it difficult to directly compare results. A comprehensive recent review of research on text complexity can be found in Solnyshkina et al. (2022).

Thus, questions regarding how complexity should be represented, and whether it depends linearly on average sentence length and average word length, remain relevant.

This study aims to analyze a new corpus of school textbook texts for grades 2–11. We evaluate both established methods and explore the application of novel approaches. The dataset is described in Section 2.

The following questions were considered within the framework of the work:

1. How well does linear regression based on two "classical" features – average sentence length and average word length – assess textbook complexity?
2. Is it possible to improve the assessment of textbook complexity based on the "classical" characteristics using more advanced models?
3. Is it possible to improve the assessment of the complexity of the entire textbook based on knowledge of its parts?
4. To what extent can the assessment of textbook complexity be improved knowing all the other characteristics?
5. Which characteristics are the most significant, and how does complexity depend on them?

2 Data

We compiled a corpus of texts from Russian school textbooks for grades 2 through 11. The Russian school system comprises 11 grades; first grade was excluded because the texts are too simple for meaningful complexity analysis. Third-grade textbooks were also excluded due to technical constraints.

The corpus covers four subject areas: humanities (history, social studies), natural sciences (physics, biology, "The world around us", ecology, geography), mathematical sciences (mathematics, computer science), and language arts (Russian language, literature). Chemistry textbooks were excluded because chemical formulas present significant preprocessing challenges.

All textbooks underwent a preprocessing stage prior to analysis: illustrations and ancillary content (e.g., page headers, publisher information, exercise labels) were removed, and obvious typographical errors were corrected.

We identified 47 linguistic features, listed in Appendix 1. The feature set encompasses descriptive, morphological, syntactic, lexical, and discursive categories. Most features previously shown to be informative for readability assessment are included in this list.

To compute feature values, we developed the RuLingva software package (https://rulingva.kpfu.ru/), which is described in Solnyshkina et al. (2024). RuLingva employs the Natasha morphological analyzer (https://github.com/natasha/natasha) for lemma extraction, part-of-speech tagging, and other preprocessing tasks.

We note the following regarding data structure. Textbooks are segmented into chunks of approximately equal length, and features are computed for each chunk. The feature representation of a textbook is defined as the average of its chunk-level values. In this hierarchical structure, the complexity label is assigned to the entire textbook rather than to individual chunks. However, in some subsequent analyses, complexity labels will also be associated with chunks for methodological purposes.

The initial dataset comprised 16,596 text chunks extracted from 206 textbooks, each described by 47 features (excluding token count). During preprocessing, chunks containing fewer than 500 tokens were removed, as typical chunks contain 700–800 tokens; such short chunks were treated as statistical outliers. After filtering, 16,467 chunks remained for analysis.

3 Applied methods for comparing models

Here we present the model evaluation metrics. Let y_n denote the observed complexity class for the n-th observation, and $\widehat{y}_n$ the predicted complexity class. Let N be the total number of observations. For each prediction method, we calculate the following metrics:

- **Exact-match accuracy**

$$R_0 = \frac{1}{N} \sum_n I(y_n = \widehat{y}_n), \tag{1}$$

  where $I(\cdot)$ is the indicator function.

- **Normalized absolute deviation score**

$$R_1 = 1 - \frac{\sum_n |y_n - \widehat{y}_n|}{\sum_n |y_n - y_{\text{med}}|}, \tag{2}$$

  where y_{med} is the sample median of $y_1, \ldots, y_N$.

- **Coefficient of determination (R^2)**

$$R_2 = 1 - \frac{\sum_n (y_n - \widehat{y}_n)^2}{\sum_n (y_n - \overline{y})^2}, \tag{3}$$

  where $\overline{y}$ is the sample mean.

The closer each score is to one, the better the model performance. To reduce variance and mitigate overfitting, all metrics are computed using the same cross-validation splits.

Although we report all metrics for each model, we consider R_1 to be the primary evaluation criterion, as it is best suited to our task. The R_0 metric is overly sensitive to small estimation errors (treating a prediction

of grade 5 as equally wrong whether the true label is 4 or 10), while R_2 relies heavily on the assumption that the numerical encoding of complexity classes reflects meaningful ordinal distances between them.

Note that R_1 (analogous to the Nash–Sutcliffe efficiency coefficient) is scale-invariant and interpretable as the proportion of variance explained relative to a median baseline, making it particularly suitable for ordinal regression tasks where absolute error magnitude matters more than exact class matching.

There is no fundamental difference between the R_2 score and the mean squared error, $\mathrm{MSE} = N^{-1} \sum_n (y_n - \widehat{y}_n)^2$, nor between R_1 and the mean absolute error, $\mathrm{MAE} = N^{-1} \sum_n |y_n - \widehat{y}_n|$. Naturally, the model with the highest R_1 will exhibit the lowest MAE among a given set of models. The R_1 score expresses prediction error as a proportion of the total variation in the data, whereas MAE reports the absolute magnitude of prediction error. Since both interpretations are informative, we include MSE and MAE values for selected models.

4 Classical linear model

Here we investigate the linear relationship between complexity class and two numerical features: average sentence length (in words) and average word length (in syllables). The distributions of these features across complexity classes are shown in Figure 1. The relationship between average sentence length and complexity class appears more clearly linear than that for average word length. Moreover, average word length does not exhibit a monotonic relationship with class. Consequently, the regression coefficient for word length is smaller than those reported in previous works (e.g., Matskovskiy, 1976; Oborneva, 2006; Solovyev et al., 2018).

We model this relationship using both simple linear regression and ordinal regression approaches. Simple linear regression is fitted by least squares with complexity class as the target variable. Both regression coefficients are significantly positive, as summarized in Table 1.[1]

During prediction, we round the continuous regression output to the nearest valid class value. To assess model performance robustly, we apply 8-fold cross-validation. Results are presented as a contingency table in Table 2. From this table, we compute the following metrics: $R_0 \approx 0.276$, $R_1 \approx 0.4$, $R_2 \approx 0.554$, corresponding to $\mathrm{MSE} \approx 2.3$ and $\mathrm{MAE} \approx 1.15$.

[1]Thus, the final dependence could be represented via formula:

$$\begin{aligned}\text{Complexity class} = {} & -5.299 + 0.512 \cdot \text{Average number of words per sentence} \\ & + 2.466 \cdot \text{Average number of syllables per word.}\end{aligned}$$

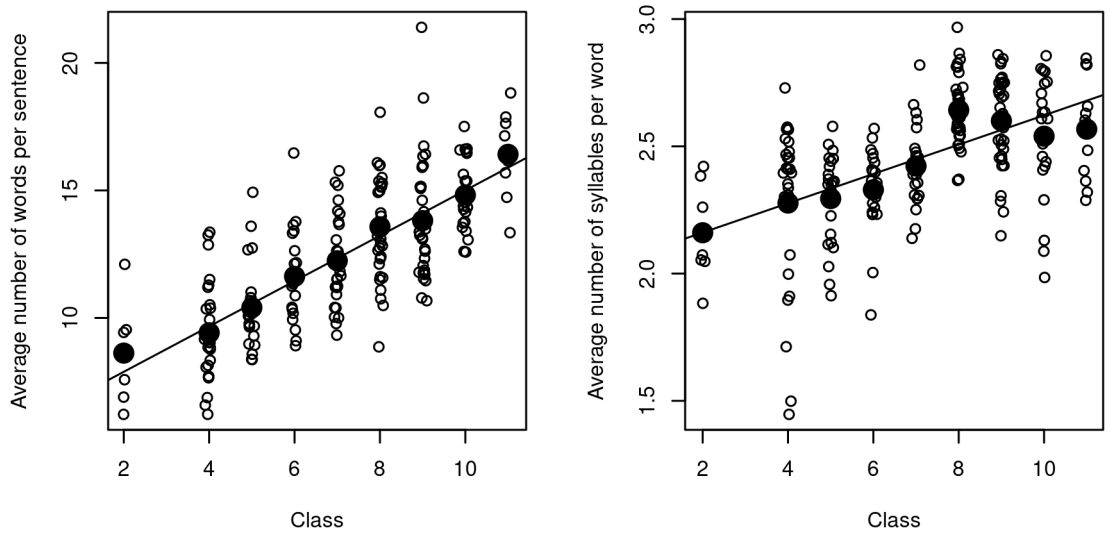

Figure 1: Average sentence length and average word length by complexity class. Each point represents a single textbook; class labels have been jittered slightly to reduce overplotting. Solid points indicate class-wise means. Lines represent least-squares fits with complexity class as the explanatory variable. Note that our analysis models the inverse relationship (complexity class as a function of linguistic features), so these lines should not be interpreted as the final predictive model.

Table 1: Simple linear regression coefficients fit by least squares.

Variable	Estimate	Std. err.	t value	P(> |t|)
(Intercept)	-5.299	0.996	-5.318	0
Average number of words per sentence	0.512	0.04	12.843	0
Average number of syllables per word	2.466	0.439	5.624	0

Table 2: Contingency table for simple linear regression. Rows represent true values, and columns represent estimates.

	2	4	5	6	7	8	9	10	11
2	1	3	2	1	0	0	0	0	0
4	3	2	12	5	3	1	0	0	0
5	0	0	12	9	1	2	0	0	0
6	0	0	2	9	6	4	0	0	0
7	0	0	1	8	7	8	2	0	0
8	0	0	1	1	5	13	13	1	1
9	0	0	0	4	7	11	7	3	2
10	0	0	0	2	1	9	5	5	0
11	0	0	0	0	1	2	2	5	1

Since complexity class is an ordered variable, ordinal regression is a natural modeling choice (Winship and Mare, 1984). Ordinal regression accounts for the ordered nature of complexity classes by modeling cumulative probabilities, rather than treating classes as nominal categories or continuous values.

We apply ordinal logistic regression (proportional odds model). Let X denote a vector of explanatory variables, θ the corresponding vector of regression coefficients, and K the maximum number of classes.

The model is defined as

$$\mathbf{P}(y \le k \mid X) = \sigma\left(\eta_k - X^\top \theta\right), \quad \sigma(z) = \frac{1}{1+\exp(-z)}, \quad k = 1, \ldots, K-1,$$

where $\eta_1, \ldots, \eta_{K-1}$ are cut-point parameters that, together with θ, are estimated via maximum likelihood. The predicted class is the one with the highest predicted probability.

Results for ordinal regression are presented in Table 3. The metric values are $R_0 \approx 0.247$, $R_1 \approx 0.382$, $R_2 \approx 0.546$. In terms of predictive accuracy, ordinal regression performs nearly identically to simple linear regression; however, it additionally provides class-probability estimates for each textbook. An example of such probabilistic predictions is shown in Figure 2. This figure also suggests that the complexity class estimation task is inherently uncertain, with substantial probability mass often spread across adjacent grades.

Table 3: Contingency table for ordinal linear regression. Rows represent true values, and columns represent estimates.

	2	4	5	6	7	8	9	10	11
2	0	6	1	0	0	0	0	0	0
4	3	12	7	0	2	2	0	0	0
5	0	12	4	2	4	1	1	0	0
6	0	2	7	0	7	5	0	0	0
7	0	1	7	0	6	8	4	0	0
8	0	1	0	0	5	12	14	2	1
9	0	0	1	1	5	10	9	6	2
10	0	0	0	2	1	6	7	5	1
11	0	0	0	0	0	2	2	4	3

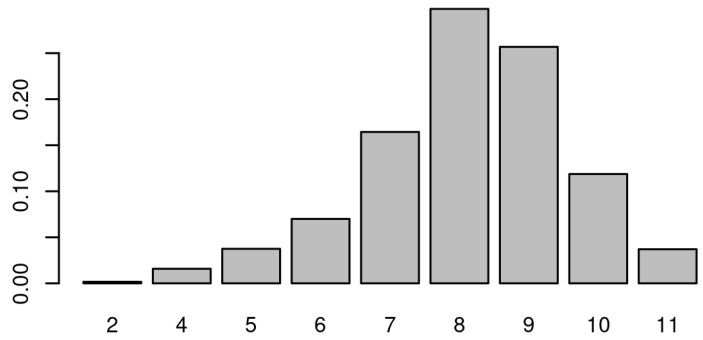

Figure 2: Predicted class probabilities for a randomly selected textbook, as estimated by ordinal regression. The true complexity class is 10 in this example.

5 Model refinement

We now evaluate additional regression methods on the same data and compare them to the baseline models. Let x_1 and x_2 denote the explanatory variables (average sentence length and average word length, respectively). All models are assessed using the R_0, R_1, and R_2 metrics computed via cross-validation. Additionally, each model is trained in two variants: on the textbook-level dataset and on the chunk-level dataset. The models considered are:

1. **Simple linear regression.**
2. **Ordinal logistic regression.**
3. **Additive regression** (Wood, 2011). This model extends simple linear regression by replacing the linear predictor $\theta_1 x_1 + \theta_2 x_2$ with an additive nonlinear specification $f_1(x_1) + f_2(x_2)$, where f_1 and f_2 are smooth functions represented by cubic splines. The model is fitted via penalized least squares, with smoothing parameters selected by generalized cross-validation (GCV).
4. **Ordinal additive regression.** This model combines the ordinal regression framework with the additive spline specification described above.
5. **Tensor-product spline regression.** This model extends additive regression by including a bivariate interaction term $f_{1,2}(x_1, x_2)$, parameterized as a cubic spline surface. The function space for such surfaces is constructed via the tensor product of univariate cubic spline bases.
6. **Ordinal tensor-product spline regression.**
7. **Gradient boosting for regression.** We use the XGBoost implementation (Chen and Guestrin, 2016).
8. **Gradient boosting for classification.** We use the XGBoost implementation (Chen and Guestrin, 2016).
9. **Feedforward neural network for regression.** Hyperparameters were selected via cross-validation. The final architecture comprises two hidden layers with 32 units each, ReLU activation, no dropout, and the Adam optimizer with a learning rate of 0.003. Training was terminated via early stopping on a validation set comprising approximately 20% of the data.
10. **Feedforward neural network for classification.** This model is nearly identical to the regression variant, but employs a softmax output layer for class-probability estimation.

To ensure consistent evaluation across models, all continuous predictions are rounded to the nearest valid complexity class.

Model comparison results are presented in Table 4. We conclude that none of the tested methods substantially outperform simple linear regression. The best-performing model was additive regression. Figure 3 shows the estimated component functions f_1 and f_2 from this model.

A further observation is that models trained on chunk-level data consistently yield poorer performance than those trained on textbook-level aggregates. More complex models, such as gradient boosting and neural networks, performed considerably worse than the baseline.

Table 4: Comparison of modeling approaches using classical features (average sentence length and average word length). All metrics are computed using the same book-level cross-validation splits.

Method	R_0	R_1	R_2
Simple linear regression	0.272	0.408	0.569
Additive regression	0.277	**0.418**	**0.574**
Ordinal linear regression	0.272	0.405	0.566
Ordinal additive regression	**0.296**	0.405	0.538
Simple linear regression, chunks	0.189	0.086	0.069
Additive regression, chunks	0.209	0.106	0.074
Ordinal linear regression, chunks	0.136	-0.086	-0.366
Ordinal additive regression, chunks	0.131	-0.104	-0.414
Tensor-product spline regression	0.282	0.413	0.572
Ordinal tensor-product spline regression	0.248	0.375	0.526
Tensor-product spline regression, chunks	0.204	0.094	0.049
Ordinal tensor-product spline regression, chunks	0.136	-0.124	-0.461
Neural network	0.034	-1.686	-5.125
Classification neural network	0.262	0.347	0.452
Neural network, chunks	0.034	-1.686	-5.125
Classification neural network, chunks	0.117	-0.197	-0.584
Gradient boosting	0.262	0.344	0.466
Classification gradient boosting	0.286	0.316	0.375
Gradient boosting, chunks	0.286	0.316	0.375
Classification gradient boosting, chunks	0.286	0.316	0.375

6 Leveraging the hierarchical data structure

Here we explore methods that exploit the hierarchical structure of our data. In this setting, predictions for a textbook can incorporate information from all its constituent chunks. We continue to restrict our analysis to the classical feature set: average sentence length and average word length.

We evaluate the following approaches:

1. **Cascade modeling.** This approach requires a base regression method (such as linear or additive regression) and operates as follows. First, the base model is trained on textbook-level data. This model is then used to generate predictions for each chunk within a textbook. Finally, chunk-level

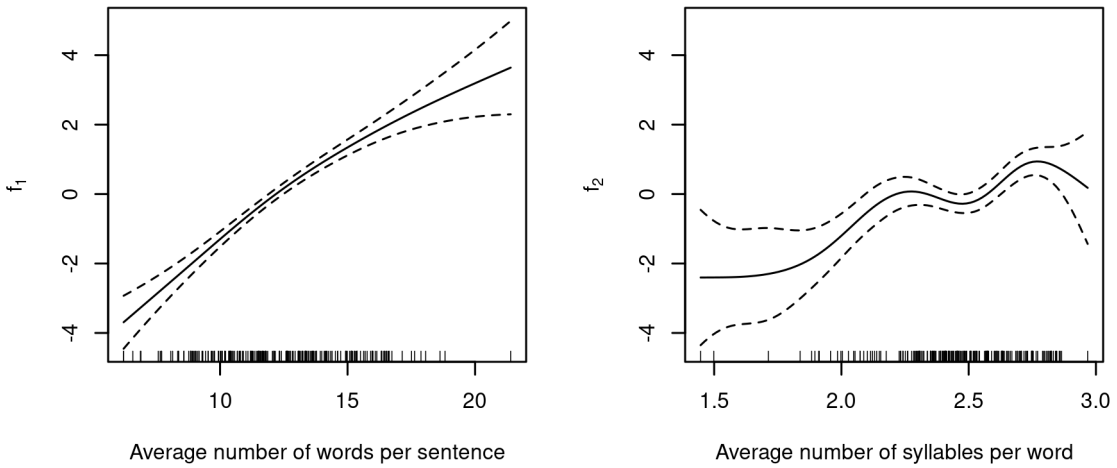

Figure 3: Estimated component functions f_1 (average sentence length) and f_2 (average word length) from the additive regression model predicting complexity class. Dashed lines indicate approximate 95% confidence intervals.

predictions are aggregated to produce a textbook-level prediction. Aggregation is implemented in two variants:

- *Simple averaging:* the arithmetic mean of chunk-level point predictions.
- *Probabilistic averaging:* applicable when the base model outputs class probabilities; probabilities are averaged across chunks before selecting the final class.

Base methods are some of those listed in the previous section.

2. **Covariance augmentation.** For each textbook, we compute the sample covariance matrix from its chunk-level features. The variance and covariance terms are then appended to the feature vector. With two classical features, this adds only three variables. Regression models from the previous section are then applied to this augmented feature set.

3. **Generative hierarchical model.** Let X denote the two-dimensional feature vector for a single chunk. We assume that, conditional on the complexity class y, X follows a bivariate normal distribution with mean θ and covariance matrix Λ. The mean parameter θ is itself a random vector, shared across all chunks from the same textbook, with distribution $\theta \mid y \sim \mathcal{N}(\mu(y), \kappa(y)^{-1}\Lambda)$. The covariance matrix Λ follows an inverse Wishart distribution with parameters $\nu(y)$ and $\Sigma(y)$. Class priors are specified as $\mathbf{P}(y = k) = \pi_k$. Prediction for a textbook is based on the posterior probabilities $\mathbf{P}(y = k \mid X_1, \ldots, X_n)$, where n is the number of chunks. We consider several model specifications: a uniform prior over classes and a homoscedastic variant where $\nu(y)$ and $\Sigma(y)$ are constant across classes. Parameter estimation, predictive inference, and derivations are provided in the Appendix 2.

The normality assumption underlying the generative model is motivated by visual inspection of chunk-level scatter plots. Figure 4 suggests this assumption is plausible. A further consequence of normality is that the mean vector and covariance matrix are sufficient statistics for the chunk distribution, rendering additional summary statistics redundant. However, formal multivariate normality tests (specifically, Royston's test; Royston, 1983) indicate that chunk-level data deviate from normality for approximately 70% of textbooks at the 0.1 significance level. Nevertheless, we retain the normality assumption to obtain a simple, interpretable model with closed-form solutions for parameter estimation.

Results are presented in Table 5. We find that cascade methods substantially improve the predictive performance of base models. Incorporating covariance information yields further gains. The generative model performs comparably to cascade approaches. Figure 5 displays the component functions from an additive regression model fitted on textbook-level data augmented with variance and covariance features. Although the generative model also leverages covariance information, it appears to do so in a less flexible manner, resulting in comparatively lower metric scores.

Table 5: Comparison of modeling approaches leveraging the hierarchical data structure and classical features. All metrics are computed using the same textbook-level cross-validation splits.

Method	R_0	R_1	R_2
Cascade, linear regression	0.272	0.403	0.559
Cascade, additive regression	0.248	0.403	0.574
Cascade, ordinal regression	0.228	0.357	0.516
Cascade, ordinal additive regression	0.267	0.377	0.499
Cascade, neural network	0.296	0.42	0.56
Cascade, classification neural network	0.272	0.314	0.373
Cascade, gradient boosting	0.267	0.347	0.475
Cascade, classification gradient boosting	0.262	0.311	0.387
Generative model, homoscedasticity, flat prior	0.282	0.311	0.407
Generative model, heteroscedasticity, flat prior	0.296	0.309	0.35
Generative model, homoscedasticity, variate prior	0.252	0.365	0.508
Generative model, heteroscedasticity, variate prior	0.282	0.39	0.53
Covariance augmentation, linear regression	0.286	0.433	0.604
Covariance augmentation, additive regression	**0.325**	**0.461**	**0.622**
Covariance augmentation, ordinal regression	0.272	0.441	0.618
Covariance augmentation, ordinal additive regression	0.301	0.446	0.613
Covariance augmentation, neural network	0.034	-1.686	-5.125
Covariance augmentation, gradient boosting	0.301	0.451	0.581
Covariance augmentation, classification gradient boosting	0.316	0.334	0.376

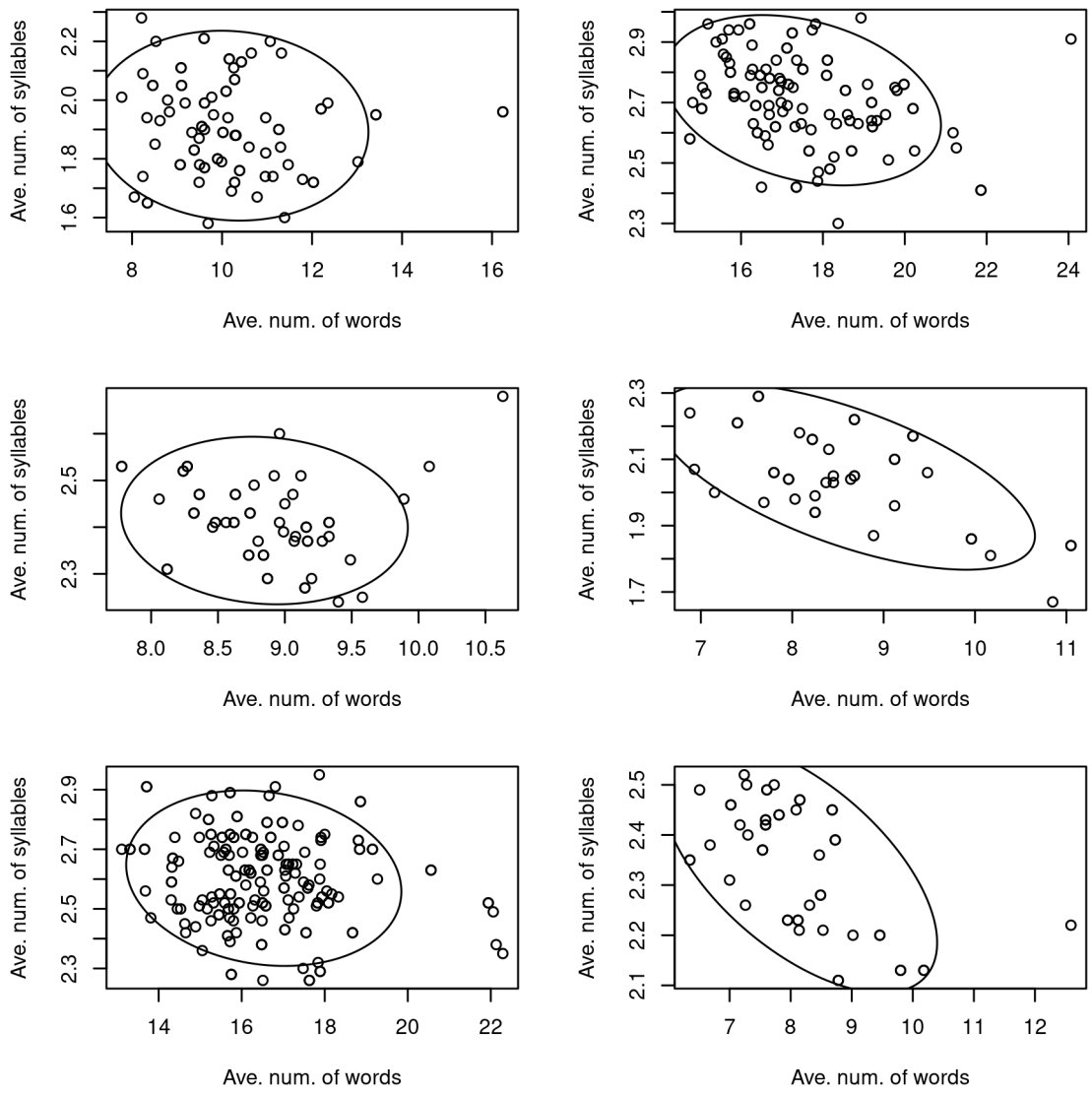

Figure 4: Chunk-level scatter plots with dispersion ellipses for several randomly selected textbooks.

7 Using the full feature set

As noted earlier, the dataset comprises 47 numerical features. In principle, all features could be leveraged to improve complexity class prediction. We apply the same modeling approaches as in previous sections, with some modifications to accommodate the higher dimensionality. Specifically, we consider:

1. The models from Section 5, excluding additive regression and tensor-product spline models.

2. Stepwise variable selection applied to linear regression. We evaluate three variants: forward selection, backward elimination, and bidirectional stepwise selection (Hastie et al., 2009). Model selection is based on leave-one-out cross-validation (LOOCV), which can be computed efficiently using the PRESS statistic from Allen (1974) under squared-error loss.

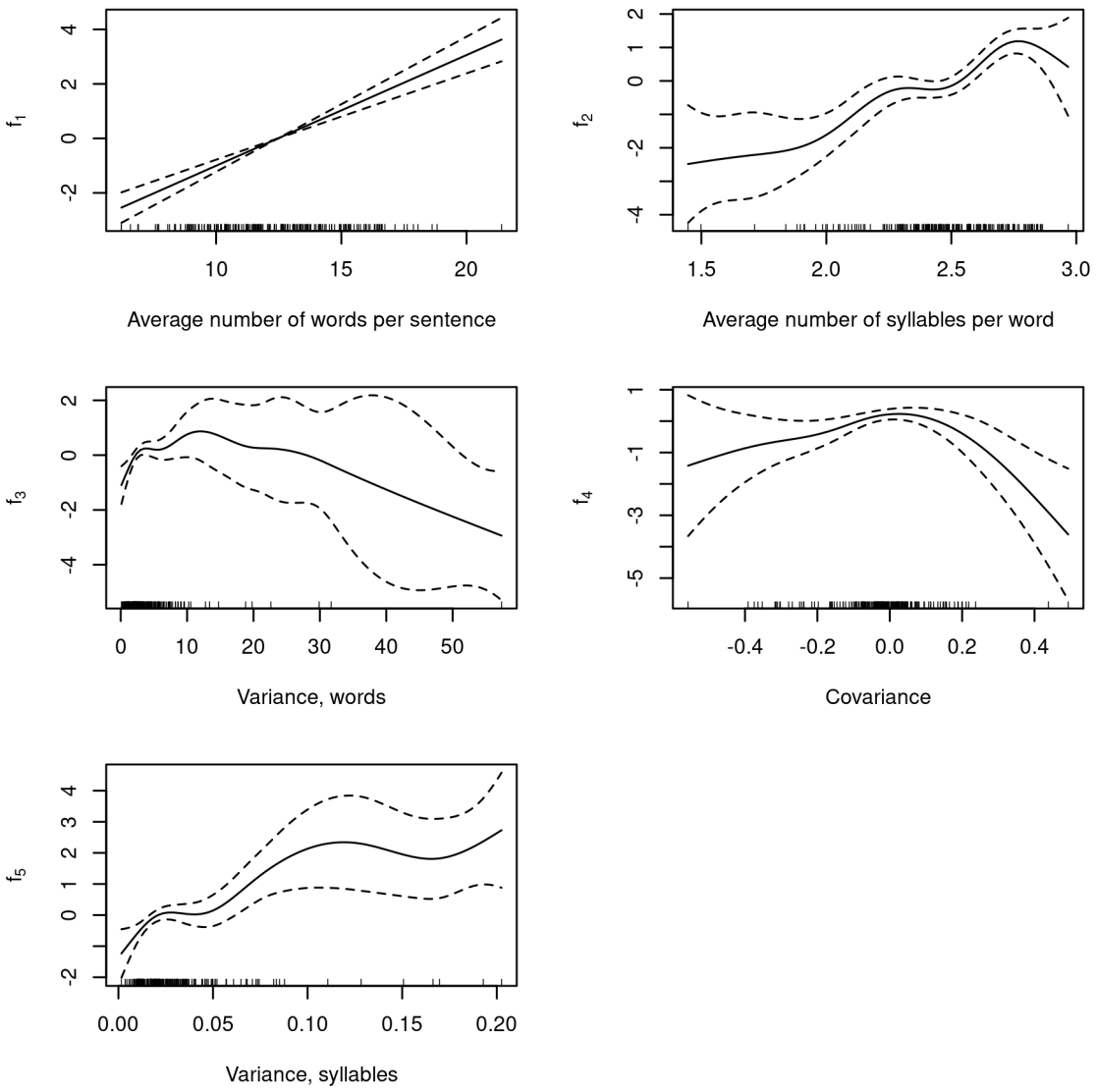

Figure 5: Component functions from the additive regression model predicting complexity class, fitted on textbook-level data augmented with variance and covariance features.

3. Selected models from Section 5 applied to the reduced feature set identified by stepwise selection.

Results are presented in Table 6. The best-performing models achieved $\mathrm{MSE} = 1.04$, $\mathrm{MAE} = 0.69$ for full set and $\mathrm{MSE} = 0.84$, $\mathrm{MAE} = 0.61$ for reduced feature set. Notably, stepwise selection degraded predictive performance, likely due to its instability when optimizing cross-validation metrics. Applying other models to the reduced feature set yielded better results; however, because feature selection was performed on the full dataset prior to cross-validation, these metrics are likely optimistically biased due to data leakage. To obtain unbiased estimates, feature selection should be nested within each cross-validation fold; we acknowledge this limitation and report the current results as upper-bound estimates. Additive models were not attempted with the full feature set, as the feature-to-sample ratio is too high for such methods to perform reliably.

Table 6: Comparison of modeling approaches using the full feature set. All metrics are computed using the same textbook-level cross-validation splits. Best results are highlighted separately for the full feature set and the reduced feature set.

Method	R_0	R_1	R_2
Linear regression	0.471	**0.641**	**0.799**
Ordinal regression	0.461	0.633	0.795
Gradient boosting	0.442	0.63	0.796
Classification gradient boosting	0.476	0.635	0.781
Neural network	0.476	0.635	0.781
Classification neural network	0.466	0.613	0.742
Backward stepwise regression	**0.515**	0.557	0.543
Forward stepwise regression	0.034	-1.686	-5.125
Two-way stepwise regression	0.495	0.597	0.676
Linear regression, reduced columns	0.485	**0.678**	**0.838**
Ordinal regression, reduced columns	0.49	0.661	0.816
Gradient boosting, reduced columns	0.447	0.557	0.66
Gradient boosting, classification, reduced columns	**0.495**	0.542	0.543
Neural network, reduced columns	0.034	-1.686	-5.125
Classification neural network, reduced columns	0.495	0.625	0.717

Leveraging the hierarchical data structure with the full feature set may yield further improvements; we leave this exploration to future work.

We also examine the relationship between complexity class and the selected features. Coefficients from the linear regression model fitted on the reduced feature set are shown in Table 7. This reduced set comprises 25 features. As is standard, the t-statistic for each coefficient indicates both the direction and relative strength of the predictor's association with the outcome. Notably, the two classical features (ASL and AWL) were not retained in the reduced model. This is likely due to their high correlation with other selected features; thus, information about sentence and word length is captured indirectly through correlated predictors.

8 Conclusions

The classical readability model – which expresses text complexity as a linear combination of average sentence length (ASL, in words) and average word length (ASW, in syllables) – was validated on our corpus. Researchers may substitute an ordinal regression framework if the ordered nature of complexity classes warrants it, though predictive performance remains comparable. While the classical model exhibits substantial error when complexity is defined by textbook target audience (i.e., grade level), more flexible models using the same two features do not outperform it, supporting the adequacy of the linear specification.

Parameter estimates for the classical model on our data align with prior work. The ASL coefficient is close to values reported previously (e.g., Matskovskiy, 1976; Oborneva, 2006; Solovyev et al., 2018), while the ASW coefficient is smaller in magnitude but remains positive. Predictive performance, compared to

Table 7: Linear regression coefficients estimated via least squares on the reduced feature set.

Variable	Estimate	Std. Error	t value	P($>\|t\|$)
(Intercept)	-15.75562	6.98833	-2.25	0.02537
Average number of characters per word	2.67601	0.66429	4.03	8e-05
Nouns	0.21809	0.03297	6.62	0
Verbs	0.12258	0.02952	4.15	5e-05
Average number of adjectives per sentence	-2.51834	0.49561	-5.08	0
Numericals	-0.02619	0.00813	-3.22	0.00152
Average rank by Sharov dictionary	0.00029	0.00011	2.66	0.00849
Frequency by Sharov dictionary	0.00228	0.00063	3.61	0.00039
Abstractness score	3.37084	1.24693	2.7	0.00752
Local noun overlap	4.61296	2.64968	1.74	0.0834
Global noun overlap	14.71276	5.50109	2.67	0.00817
Local argument overlap	-5.70477	1.57202	-3.63	0.00037
Type/Token Ratio absolute	-474.42970	481.91975	-0.98	0.32621
Type/Token Ratio average	478.34127	482.14596	0.99	0.32248
Nominative case Noun	-0.02837	0.00918	-3.09	0.00233
Dative case Noun	-0.05885	0.03385	-1.74	0.08385
Present tense Verb	0.06762	0.01380	4.9	0
Past tense Verb	0.06839	0.01374	4.98	0
Adjective/Noun ratio	54.26626	9.66085	5.62	0
Monosyllabic words	-0.02601	0.00820	-3.17	0.00177
Two-syllable words	-0.02409	0.00660	-3.65	0.00034
Three-syllable words	-0.02989	0.00763	-3.92	0.00013
Four-syllable words	-0.03105	0.00941	-3.3	0.00117
Average number of adverbs per sentence	13.06065	1.46875	8.89	0
Content words	-0.12834	0.02310	-5.56	0
Lexical density	-28.94135	7.55847	-3.83	0.00018

earlier results on a textbook corpus (Solovyev et al., 2018), is somewhat lower; this difference is likely attributable to the greater subject-area diversity in our dataset.

Exploiting the hierarchical structure of the data – by segmenting textbooks into chunks and aggregating chunk-level information – improves textbook-level predictions. The most effective approach augments classical features with covariance statistics computed across chunks within each textbook. This method yielded approximately 10% improvement across the R_0, R_1, and R_2 metrics. A generative model based on a multivariate normality assumption achieved performance comparable to linear regression, suggesting that the added complexity did not translate into predictive gains.

Incorporating all 47 available features further improved predictive performance, though prediction errors were not eliminated. Linear regression emerged as the best-performing model among those evaluated. Relative to the classical two-feature linear model, the full-feature model achieved a 50% increase in prediction score: $R_1 = 0.641$ versus $R_1 = 0.418$. The strong performance of linear regression, relative to more flexible methods, likely reflects the limited sample size (206 textbooks) relative to feature dimensionality (47 predictors), which predisposes complex models to overfitting.

References

Abramov, A., Ivanov, V., Solovyev, V. (2023). Estimating Lexical Complexity in Multi-Domain Settings for the Russian Language. *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pp. 516–526.

Abramov, A., Ivanov, V. V. (2022). Collection and evaluation of lexical complexity data for Russian language using crowdsourcing. *Russian Journal of Linguistics*, 26(2), pp. 409–425. https://doi.org/10.22363/2687-0088-30118

Aleksandrovich, S. S. (2022). What neural networks know about linguistic complexity. *Russian Journal of Linguistics*, 26(2), pp. 371–390. https://doi.org/10.22363/2687-0088-30178

Allen, D. M. (1974). The Relationship Between Variable Selection and Data Agumentation and a Method for Prediction. *Technometrics*, 16(1), pp. 125–127. https://doi.org/10.1080/00401706.1974.10489157

Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. https://doi.org/10.1145/2939672.2939785

Corlatescu, D., Ruseti, S., Dascalu, M. (2022). ReaderBench: Multilevel analysis of Russian text characteristics. *Russian Journal of Linguistics*, 26(2), pp. 342–370. https://doi.org/10.22363/2687-0088-30145

Feng, J., Yu, S. (2024). A Method for Measuring Word Sequence Complexity of Text. *Journal of Quantitative Linguistics*, pp. 1–21. https://doi.org/10.1080/09296174.2024.2417448

Flesch, R. (1948). A new readability yardstick. *Journal of applied psychology*, 32(3), pp. 221–233. https://doi.org/10.1037/h0057532

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (1995). *Bayesian data analysis*. Chapman; Hall/CRC.

Hastie, T., Tibshirani, R., Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Technical report, Naval Technical Training Command Millington TN Research Branch*, pp. 1–51.

Matskovskiy, M. (1976). Problemy chitabelnosti pechatnogo teksta [The Problems of Typed Text Readability]. *Smyslovoe vospriyatie rechevogo soobshcheniya (v usloviyakh massovoy kommunikatsii)*, pp. 126–142.

McLachlan, G. J., Krishnan, T. (2007). *The EM algorithm and extensions*. John Wiley & Sons. https://doi.org/10.1002/9780470191613

Mikk, Y. (1974). Methodology for developing readability formulas. *Sovetskaya pedagogika i shkola*, (9), pp. 78–163.

Morozov, D. A., Glazkova, A. V., Iomdin, B. L. (2022). Text complexity and linguistic features: Their correlation in English and Russian. *Russian Journal of Linguistics*, 26(2), pp. 426–448. https://doi.org/10.22363/2687-0088-30132

Oborneva, I. V. (2006). Avtomatizirovannaja ocenka složnosti ucebnyx tekstov na osnove statisticeskix parametrov [Automatic evaluation of the complexity of educational texts on the basis of statistical parameters] [Doctoral dissertation, Ph. D. thesis].

Royston, J. P. (1983). Some techniques for assessing multivarate normality based on the Shapiro-Wilk W. *Journal of the Royal Statistical Society (C)*, 32(2), pp. 121–133. https://doi.org/10.2307/2347291

Shpakovskiy, Y., Et al. (2007). Otsenka trudnosti vospriyatiya i optimizatsiya slozhnosti uchebnogo teksta [Evaluation of the difficulty of perception and optimization of the text complexity] [Doctoral dissertation, PhD thesis].

Solnyshkina, M., Ivanov, V., Solovyev, V. (2018). Readability formula for Russian texts: A modified version. *Advances in Computational Intelligence: 17th Mexican International Conference on Artificial Intelligence, MICAI 2018, Guadalajara, Mexico, October 22–27, 2018, Proceedings, Part II 17*, pp. 132–145.

Solnyshkina, M., Solovyev, V., Danilov, A., Zamaletdinov, R., Akhtyamova, S. (2024). Multilevel Analyses of Russian Texts with RuLingva: A Case Study. *Mexican International Conference on Artificial Intelligence*, pp. 234–246.

Solnyshkina, M., Solovyev, V., Gafiyatova, E., Martynova, E. (2022). Text complexity as interdisciplinary problem. *Voprosy Kognitivnoy Lingvistiki*, pp. 18–39.

Solovyev, V., Ivanov, V., Solnyshkina, M. (2018). Assessment of reading difficulty levels in Russian academic texts: Approaches and metrics. *Journal of intelligent & fuzzy systems*, 34(5), pp. 3049–3058. https://doi.org/10.21236/ADA006655

Solovyev, V., Solnyshkina, M. I., McNamara, D. S., Et al. (2022). Computational linguistics and discourse complexology: Paradigms and research methods. *Russian Journal of Linguistics*, 26(2), pp. 275–316. https://doi.org/10.22363/2687-0088-31326

Winship, C., Mare, R. D. (1984). Regression Models with Ordinal Variables. *American Sociological Review*, 49(4), pp. 512–525. https://doi.org/10.2307/2095465

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1), pp. 3–36. https://doi.org/10.1111/j.1467-9868.2010.00749.x

Yimam, S. M., Biemann, C., Malmasi, S., Paetzold, G. H., Specia, L., Štajner, S., Tack, A., Zampieri, M. (2018). A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*, pp. 1–13. https://doi.org/10.48550/arXiv.1804.09132

Appendix 1. List of RuLingva parameters

- Tokens (words only)
- Types (words only)
- Number of syllables
- Number of sentences
- Content words per sentence
- Average number of tokens in a sentence
- Average number of syllables in a word
- Average number of characters in a word
- Nouns
- Average number of nouns in a sentence
- Verbs
- Average number of verbs in a sentence
- Adjectives
- Average number of adjectives in a sentence
- Adverbs
- Pronouns
- Numerals
- Average frequency rank (by Sharoff)
- Frequency (by Sharoff)
- FKGL (SISmod)
- FLGL (Onorneva)
- Abstractness
- Local noun overlap

- Global noun overlap
- Local argument overlap
- Global argument overlap
- Type/Token ratio (absolute)
- Type/Token ratio (average)
- Nominative case (noun)
- Genitive case (noun)
- Accusative case (noun)
- Instrumental case (noun)
- Prepositional case (noun)
- Present tense (verb)
- Future tense (verb)
- Past tense (verb)
- Verb/Noun ratio
- Adjective/Noun ratio
- Social science terms
- One syllable words
- Two-syllable words
- Three-syllable words
- Four-syllable words
- Average number of adverbs in a sentence
- Hapax legomena
- Content words
- Lexical density

Appendix 2. Generative model parameters estimation

Let us define the generative model from Section 6 more formally. Suppose N books with complexity classes $y_1, \ldots, y_N$ are observed. Each class y_n takes values from the set $1, \ldots, K$. Each book consists of m_n chunks, $n = 1, \ldots, N$, and each chunk is represented as a vector of M real-valued characteristics. All chunks of a single book indexed by n are denoted by $\mathbf{X}_n = (X_{n,1}, \ldots, X_{n,m_n})$ and $X_{n,i} \in \mathbb{R}^M, i = 1, \ldots, m_n$.

The model assumes the existence of M-dimensional unobserved random vectors $\theta_1, \ldots, \theta_N$ and $M \times M$ positive definite random matrices $\Lambda_n, n = 1, \ldots, N$. The random variables $y_n, \mathbf{X}_n, \theta_n, \Lambda_n$ are assumed independent across n. Another assumption is that chunks $X_{n,i}$ of a single book n are conditionally independent given y_n, θ_n, Λ_n. Now, for any $n = 1, \ldots, N$ let

$$\mathbf{P}(y_n = k) = \pi_k, k = 1, \ldots, K;$$

$$\Lambda_n \mid y_n \sim \mathcal{W}^{-1}\big(\nu(y_n), \Sigma(y_n)\big);$$

$$\theta_n \mid \Lambda_n, y_n \sim \mathcal{N}\left(\mu(y_n), \frac{1}{\kappa(y_n)}\Lambda_n\right);$$

$$X_{n,i} \mid \theta_n, \Lambda_n, y_n \sim \mathcal{N}\big(\theta_n, \Lambda_n\big), \quad i = 1, \ldots, m_n.$$

In the last three expressions, a conditional distribution is implied. Parameters $\mu(y), y = 1, \ldots, K$ are vectors in $\mathbb{R}^M$, and $\Sigma(y)$ is an $M \times M$ positive definite matrix. The notation $\mathcal{W}^{-1}$ denotes the inverse Wishart distribution. The above probabilistic formulation of the model is a variation of the standard conjugate model of normally distributed observations, see Gelman et al. (1995, paragraph 3.6).

Let $\varphi(X \mid \mu, \Lambda)$ denote the density of the normal distribution with mean vector μ and covariance matrix Λ evaluated at X, and $p_{\mathcal{W}^{-1}}(\Lambda \mid \nu, \Sigma)$ denote the density of the inverse Wishart distribution with parameters ν, Σ evaluated at Λ. Their respective formulas for M-dimensional random variables are given by

$$\varphi(X \mid \mu, \Lambda) = (2\pi)^{-\frac{M}{2}} |\Lambda|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(X-\mu)^T \Lambda^{-1} (X-\mu)\right),$$

$$p_{\mathcal{W}^{-1}}(\Lambda \mid \nu, \Sigma) = |\Sigma|^{\frac{\nu}{2}} 2^{-\frac{\nu M}{2}} \left(\Gamma_M\left(\frac{\nu}{2}\right)\right)^{-1} |\Lambda|^{-\frac{\nu+M+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Sigma\Lambda^{-1}\right)\right),$$

where Γ_M is the multivariate gamma function

$$\Gamma_M(x) = \pi^{\frac{M(M-1)}{4}} \prod_{i=1}^{M} \Gamma\left(x + \frac{1-i}{2}\right).$$

We also define

$$N_y = \sum_{n=1}^{N} I(y_n = y),$$

$$\overline{\mathbf{X}}_n = \frac{1}{m_n} \sum_{i=1}^{m_n} X_{n,i}.$$

A standard result (see Gelman et al., 1995, paragraph 3.6) states that the posterior distribution is also Normal-Inverse-Wishart:

$$\Lambda_n \mid y_n, \mathbf{X}_n \sim \mathcal{W}^{-1}\left(\nu(y_n) + m_n, \Sigma(y_n) + S_n + \frac{m_n \kappa(y_n)}{m_n + \kappa(y_n)} \left(\overline{\mathbf{X}}_n - \mu(y_n)\right) \left(\overline{\mathbf{X}}_n - \mu(y_n)\right)^T\right), \tag{4}$$

$$\theta_n \mid \Lambda_n, y_n, \mathbf{X}_n \sim \mathcal{N}\left(\frac{\kappa(y_n)\mu(y_n) + m_n \overline{\mathbf{X}}_n}{m_n + \kappa(y_n)}, \frac{1}{\kappa(y_n) + m_n} \Lambda_n\right), \tag{5}$$

where

$$S_n = \sum_{i=1}^{m_n} \left(X_{n,i} - \overline{\mathbf{X}}_n\right) \left(X_{n,i} - \overline{\mathbf{X}}_n\right)^T.$$

If the parameters $\pi, \mu, \kappa, \nu, \Sigma$ are known, then it is possible to calculate the posterior probabilities of any complexity class for a new observation $\mathbf{X}$ consisting of m parts:

$$\mathbf{P}(y = k|\mathbf{X}) \propto p(\mathbf{X} \mid y = k)\mathbf{P}(y = k) = \pi_k \frac{p(\mathbf{X}, \theta, \Lambda \mid y = k)}{p(\theta, \Lambda \mid \mathbf{X}, y = k)}.$$

Here $p(\mathbf{X} \mid y = k)$ is the density of $\mathbf{X}$ for known y. The right-hand side of the expression above suggests a straightforward way to compute this quantity using (4) and (5). We state the standard result of this calculation:

$$p(\mathbf{X} \mid y = k) = \pi^{-\frac{mM}{2}} |\Sigma(k)|^{\frac{\nu(k)}{2}} \left|\Sigma(k) + S + \frac{m\kappa(k)}{m + \kappa(k)} \left(\overline{\mathbf{X}} - \mu(k)\right) \left(\overline{\mathbf{X}} - \mu(k)\right)^T\right|^{-\frac{\nu(k)+m}{2}} \times$$
$$\Gamma_M\left(\frac{\nu(k) + m}{2}\right) \left[\Gamma_M\left(\frac{\nu(k)}{2}\right)\right]^{-1} \sqrt{\frac{\kappa(k)}{\kappa(k) + m}}$$

with analogous definitions for $\overline{\mathbf{X}}$ and S.

Next, we present an estimation algorithm for the parameters $\pi, \mu, \kappa, \nu, \Sigma$. Since there are unobserved variables θ_n and Λ_n, we employ the EM algorithm McLachlan and Krishnan (2007). To initialize it, we need starting parameter values. A natural initial approximation is given by

$$\mu(y) = \frac{1}{N_y} \sum_{n:y_n=y} \overline{\mathbf{X}}_n,$$

$$\Sigma(y) = \frac{1}{N_y} \sum_{n:y_n=y} \frac{1}{m_n} \sum_{i=1}^{m_n} \left(X_{n,i} - \overline{\mathbf{X}}_n\right) \left(X_{n,i} - \overline{\mathbf{X}}_n\right)^T,$$

$$\pi_k = \frac{N_k}{N},$$

and the parameters $\kappa(y)$ and $\nu(y)$ are initialized to 1.

Then the E-step and M-step are repeated until convergence.

E-step. The conditional distributions of θ_n, Λ_n given $y_n, \mathbf{X}_n$ are calculated for known parameter values. These are given by (4), (5). We define

$$\xi_n = \frac{\kappa(y_n)\mu(y_n) + m_n\overline{\mathbf{X}}_n}{m_n + \kappa(y_n)},$$

$$\kappa_n = m_n + \kappa(y_n),$$

$$\nu_n = \nu(y_n) + m_n,$$

$$\Sigma_n = \Sigma(y_n) + S_n + \frac{m_n\kappa(y_n)}{m_n + \kappa(y_n)} \left(\overline{\mathbf{X}}_n - \mu(y_n)\right) \left(\overline{\mathbf{X}}_n - \mu(y_n)\right)^T.$$

We also define

$$\lambda_n = \mathbf{E}\left(\ln|\Lambda_n| \mid y_n, \mathbf{X}_n\right) = -\sum_{i=1}^{M} \psi\left(\frac{\nu_n}{2} + \frac{1-i}{2}\right) - M\ln 2 + \ln|\Sigma_n|,$$

where ψ is the digamma function; this result is a known property of the Wishart distribution. We also use another property of the Wishart distribution:

$$L_n = \mathbf{E}\left(\Lambda_n^{-1} \mid y_n, \mathbf{X}_n\right) = \nu_n\Sigma_n^{-1}.$$

These values are calculated in the E-step and treated as known in the M-step. Let the corresponding joint density of θ_n, Λ_n (evaluated at point θ, Λ) be denoted by $q_n(\theta, \Lambda)$.

M-step. We need to maximize the function

$$(6) \quad \sum_{n=1}^{N} \left[\int\int q_n(\theta,\Lambda) \left\{ \sum_{i=1}^{m_n} \ln\varphi\left(X_{n,i} \mid \theta, \Lambda\right) + \ln\varphi\left(\theta \mid \mu(y_n), \frac{1}{\kappa(y_n)}\Lambda\right) + \ln p_{\mathcal{W}^{-1}}\left(\Lambda \mid \nu(y_n), \Sigma(y_n)\right) \right\} d\theta d\Lambda + \ln\pi_{y_n} \right]$$

with respect to parameters $\mu, \kappa, \nu, \Sigma, \pi$ subject to the constraint $\sum \pi_k = 1$.

The estimates of π_k are straightforward and given by

$$\pi_k = \frac{N_k}{N},$$

which matches the initial values.

In expression (6), only the terms depending on μ, κ, ν, Σ are the prior density terms. Thus, we need to calculate expectations of these terms. Expectation of twice the log prior density of θ equals

$$\int q(\theta \mid \Lambda) 2 \ln \varphi \left(\theta \mid \mu(y_n), \frac{1}{\kappa(y_n)}\Lambda\right) d\theta = \int q(\theta \mid \Lambda)\Big[-M \ln(2\pi) + \ln \kappa(y_n) - \ln|\Lambda| - \kappa(y_n)(\theta - \mu(y_n))^T \Lambda^{-1}(\theta - \mu(y_n))\Big] d\theta. \tag{7}$$

Now we use the fact that $q_n(\theta|\Lambda)$ is a normal density with mean ξ_n and covariance matrix $(\kappa(y_n)+m_n)^{-1}\Lambda$. Thus the expression (7) equals to

$$-M \ln(2\pi) + \ln \kappa(y_n) - \ln|\Lambda| - \kappa(y_n)\left[(\xi_n - \mu(y_n))^T \Lambda^{-1}(\xi_n - \mu(y_n)) + \frac{1}{\kappa_n}\mathrm{tr}(\Lambda^{-1}\Lambda)\right] \tag{8}$$

In what follows, we use the fact that $\mathrm{tr}(\Lambda^{-1}\Lambda) = M$.

Continuing the calculation of (6) by integrating with respect to Λ, the density $q_n(\Lambda)$ is given by the expression

$$q_n(\Lambda) = p_{\mathcal{W}^{-1}}(\Lambda \mid \nu_n, \Sigma_n).$$

Combining the previous integration result (8) and $\ln p_{\mathcal{W}^{-1}}\big(\Lambda \mid \nu(y_n), \Sigma(y_n)\big)$:

$$-M \ln(2\pi) + \ln \kappa(y_n) - \ln|\Lambda| - \kappa(y_n)\left[(\xi_n - \mu(y_n))^T \Lambda^{-1}(\xi_n - \mu(y_n)) + \frac{1}{\kappa_n}\mathrm{tr}(\Lambda^{-1}\Lambda)\right] + \nu(y_n) \ln|\Sigma(y_n)| - \nu(y_n) M \ln 2 - 2 \ln \Gamma_M\left(\frac{\nu(y_n)}{2}\right) - (\nu(y_n) + M + 1) \ln|\Lambda| - \mathrm{tr}\left(\Sigma(y_n)\Lambda^{-1}\right). \tag{9}$$

Next, we calculate the expectation of (9) with respect to $q_n(\Lambda)$. Here we need $\mathbf{E} \ln|\Lambda|$ and $\mathbf{E}\Lambda^{-1}$, which are given by λ_n and L_n, respectively. Taking this step and summing over n, we obtain

$$\sum_{n=1}^{N}\Big[-M \ln(2\pi) + \ln \kappa(y_n) - \lambda_n - \kappa(y_n)(\xi_n - \mu(y_n))^T L_n(\xi_n - \mu(y_n)) - \frac{M\kappa(y_n)}{\kappa_n} + \nu(y_n) \ln|\Sigma(y_n)| - \nu(y_n) M \ln 2 - 2 \ln \Gamma_M\left(\frac{\nu(y_n)}{2}\right) - (\nu(y_n) + M + 1)\lambda_n - \mathrm{tr}\left(\Sigma(y_n) L_n\right)\Big] \tag{10}$$

up to a constant.

We now optimize this function.

The derivative of (10) with respect to $\mu(k)$ equals

$$2\kappa(k) \sum_{n:y_n=k} (L_n\xi_n - L_n\mu(k)).$$

Equating this to zero and solving gives

$$\mu(k) = \left(\sum_{n:y_n=k} L_n \right)^{-1} \left(\sum_{n:y_n=k} L_n\xi_n \right).$$

The derivative of (10) with respect to $\kappa(k)$ equals

$$\frac{N_k}{\kappa(k)} - \sum_{n:y_n=k} \left[(\xi_n - \mu(y_n))^T L_n (\xi_n - \mu(y_n)) + \frac{M}{\kappa_n} \right].$$

Therefore,

$$\kappa(k) = N_k \left(\sum_{n:y_n=k} \left[(\xi_n - \mu(y_n))^T L_n (\xi_n - \mu(y_n)) + \frac{M}{\kappa_n} \right] \right)^{-1}.$$

The derivative of (10) with respect to $\Sigma(k)$ equals

$$N_k \nu(k) \Sigma(k)^{-1} - \sum_{n:y_n=k} L_n$$

Equating this to zero and solving gives

$$\Sigma(k) = \left(\frac{1}{N_k \nu(k)} \sum_{n:y_n=k} L_n \right)^{-1}. \tag{11}$$

The derivative of (10) with respect to $\nu(k)$ equals

$$N_k \ln |\Sigma(k)| - N_k M \ln 2 - N_k \sum_{i=1}^{M} \psi \left(\frac{\nu(k)}{2} - \frac{1-i}{2} \right) - \sum_{n:y_n=k} \lambda_n.$$

Substituting (11) into this expression:

$$\ln |\Sigma(k)| = \ln \left| \frac{1}{N_k \nu(k)} \sum_{n:y_n=k} L_n \right|^{-1} = -\ln \nu(k)^{-M} \left| \frac{1}{N_k} \sum_{n:y_n=k} L_n \right| = M \ln \nu(k) - \ln \left| \frac{1}{N_k} \sum_{n:y_n=k} L_n \right|.$$

Thus, to find the M-step estimate of $\nu(k)$, we need to solve the equation

$$N_k M \ln \nu(k) - N_k \ln \left| \frac{1}{N_k} \sum_{n:y_n=k} L_n \right| - N_k M \ln 2 - N_k \sum_{i=1}^{M} \psi\left(\frac{\nu(k)}{2} - \frac{1-i}{2}\right) - \sum_{n:y_n=k} \lambda_n = 0. \quad (12)$$

This equation is solved numerically, and the resulting root is then used in (11).

In the case of a flat prior distribution for y, the probabilities π_k are fixed: $\pi_k = K^{-1}, k = 1, \ldots, K$. This assumption may be natural since the researcher does not expect any of the complexity classes to be more frequent than others.

Homoscedasticity means that the distribution of Λ_n is independent of y_n. In this case, the algorithm requires only minor modifications. The expressions for ν (12) and Σ (11) are modified so that they depend on all observations, not only those from their corresponding complexity classes. This modification is used when the number of observations is too small to properly estimate the Σ parameters. The results in Section 6 (see Table 5) show that in our case we had sufficient observations to use the heteroscedastic model.