High-dimensional data

Melvin Geubbelmans,[a] Axel-Jan Rousseau,[a] Dirk Valkenborg,[a] and Tomasz Burzykowski[a,b]
Hasselt, Belgium, and Bialystok, Poland

Big data is a relatively recent term that has emerged because of the rapid collection and generation of data, the increase in storage and computing capacity, and the rise of a new generation of machine learning algorithms. It is generally characterized by 3 "V's." The first "V," volume, refers to the size and scale of the data. This can be the number of subjects (observations) or variables (covariates, features) in the dataset. Velocity, the second "V," stands for the rate at which the data are generated and the speed of analysis. The final "V," variety, refers to the variation within a dataset and is associated with the noisiness, occurrence of missing data, or because of differences in the storage methods.

Specialized and distributed computer architectures that can provide adequate memory and processing power are often required to handle extremely large datasets. In machine learning (ML), the volume of the datasets and the increasing number of variables pose additional issues to the ML algorithms, especially when the number of variables (features) vastly exceeds the number of observations. This case is often called the ($n \leq p$)-problem or high-dimensional data; sometimes, it is called the *curse of dimensionality*. In what follows, we present 2 examples of problems regarding high-dimensional data.

In a previous article of this series,[1] the K-nearest-neighbors algorithm was discussed. One of the main assumptions of this method is that 2 observations have to be close to each other in every dimension (across each variable). Adding dimensions (variables) without adding data reduces the reliability of the algorithm because the points will lie further apart in the expanded variable space resulting in empty data regions. To compensate for this, the number of observations in the dataset should be increased accordingly.

We will use a linear regression model to illustrate another problem with high dimensionality. Let us denote by n and p the number of observations and the total number of variables ($p - 1$ explanatory variables and the dependent variable) in a dataset used to fit a linear regression model. The *least-squares method* is often applied to ensure the best fit of the model. A linear combination of the $p - 1$ explanatory variables is found such that the *residual sum of squares* is the smallest. When the number of observations is less than or equal to the number of variables ($n \leq p$), it is possible to find combinations of the features with the residual sum of squares equal to zero (see Fig 1), leading to a model that fits the data perfectly. However, the results of validating such a model on an independent testing dataset will likely be poor because of overfitting.[2]

The curse of dimensionality also holds for classification tasks. When adding more explanatory variables to a dataset, the dimensionality increases to a point in which the classification problem can be solved perfectly (without misclassification), even considering the most straightforward and inflexible models. This principle is exploited in *support vector machines*, a technique that will be discussed in a subsequent article in this series on ML.

Two approaches can be considered to address the curse of dimensionality when fitting a model: selection of only important variables (features) when building the model or dimension reduction based on data transformations.

FEATURE SELECTION

Apart from feature subset selection methods, a general approach to performing feature selection is model *regularization*, that is, adding, in the search for the best-fitting model, a penalty for the number and magnitude of the coefficients included in the model.[2] For instance, in the *Least Absolute Shrinkage and Selection Operator* (LASSO), the penalty takes the form of the sum of the absolute values of all coefficients multiplied by a constant λ from the [0, 1] interval. This penalty term is also known as the *shrinkage penalty*. LASSO performs

[a]Data Science Institute and Center for Statistics, Hasselt University, Hasselt, Belgium.
[b]Department of Biostatistics and Medical Informatics, Medical University of Bialystok, Bialystok, Poland.
This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.
Address correspondence to: Tomasz Burzykowski, Hasselt University - Data Science Institute, Agoralaan 1, Building D, B-3590 Diepenbeek, Belgium; e-mail, tomasz.burzykowski@uhasselt.be.
Am J Orthod Dentofacial Orthop 2023;164:453-6
0889-5406/$36.00

https://doi.org/10.1016/j.ajodo.2023.06.012

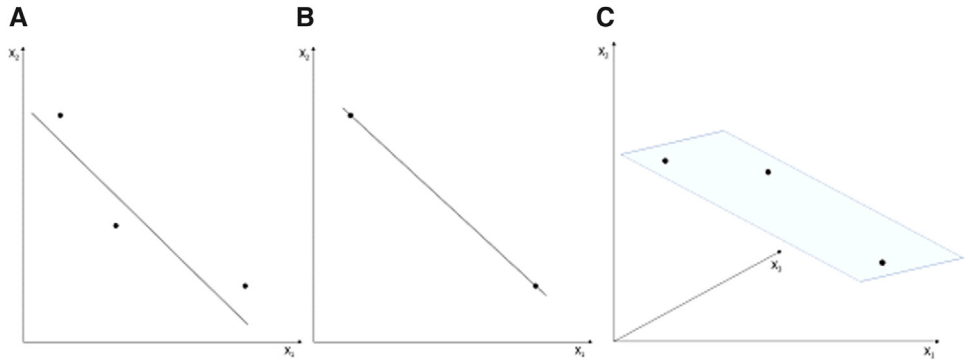

Fig 1. A, Least-squares fit of a linear regression model when $p = 2$ (explanatory variable/feature X_1 and dependent/target variable X_2) and the number of observations $n = 3$. The observed values of target X_2 deviate from the regression line (residuals are not equal to zero); hence, the residual sum of squares is not equal to zero; **B,** Least-squares fit of a linear regression model when $p = 2$ and $n = 2$. The observed values of X_2 lie on the regression line (residuals are equal to zero); hence, the residual sum of squares equals zero; **C,** Least-squares fit of a linear regression model when $p = 3$ (features X_1 and X_3 and target X_2) and the number of observations $n = 3$. The observed values of X_2 lie on the (*blue*) plane representing the linear combination of X_1 and X_3 (residuals are equal to zero); hence, the residual sum of squares equals zero.

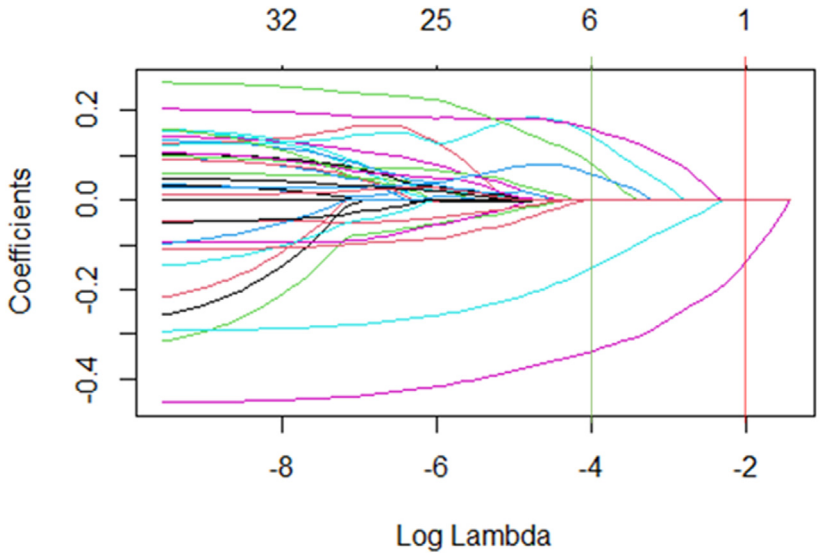

Figure 2. LASSO regression model for data in Konstantonis et al.[3] The y-axis represents the values of the coefficients for features included in the model. The x-axis represents the values of $\ln(\lambda)$, with increasing values indicating higher penalties. For larger penalties, fewer nonzero coefficients (ie, features) are included in the model. For instance, for $\ln(\lambda) = -2$ ($\lambda = 0.14$), only 1 coefficient is not equal to 0 (ie, only 1 feature is included in the model). For $\ln(\lambda) = -4$ ($\lambda = 0.02$), 6 coefficients are not equal to 0 (ie, 6 features are included in the model).

feature selection by considering various values of λ. The penalty term is small at low values, and more coefficients (and features) are allowed to enter the model. However, when λ gets large, the penalty gets larger, and only the most important features (ie, with nonzero coefficients) will remain in the model. Through the penalization process, LASSO aims to select the most optimal covariate combination that will produce a final model with high predictive performance or, equivalently, yield the smallest test error by tuning the bias-variance trade-off.

To illustrate the LASSO, we use the Konstantonis et al[3] dataset to support the extraction or nonextraction treatment plan. For this binary classification task, 26 cephalometric variables, 6 model measurements, and 2 demographic variables (gender, age) are available. The result of the LASSO procedure is displayed in Figure 2. In this example, we combine the LASSO penalty with a logistic regression model suitable for classification tasks. The logistic regression model is explained in a subsequent article in this series. The y-axis represents the estimated coefficients for the covariates (or features) included in the regression model. The x-axis represents the values of $\ln(\lambda)$. The lines with different colors show how the estimated coefficients change with λ. In particular, each line indicates that, as λ increases, the estimated coefficient tends toward zero (it is often said that the coefficient is *shrunk to zero*; hence the name of the penalty). This implies that, with increasing λ, the model will include fewer nonzero coefficients and, consequently, fewer features. In the extreme scenario when λ is very large (right side of Fig 2), the so-called null model is obtained (ie, a model with no covariates or features included). On the extreme left side, the penalty is very small or equal to zero, which reduces the LASSO procedure in this example to simple least square logistic regression. For instance, Figure 2 indicates that a model with only 1 nonzero coefficient is obtained for $\ln(\lambda) = -2$ (ie, $\lambda = 0.14$). This can be seen from the fact that the red vertical line, drawn at $\ln(\lambda) = -2$, crosses only 1 colored curve. It appears that the coefficient corresponds to the mandibular crowding feature. In contrast, when $\ln(\lambda) = -4$ (ie, $\lambda = 0.02$), a model with 6 nonzero coefficients is obtained because the *green* vertical line at $\ln(\lambda) = -4$ crosses 6 colored curves. The coefficients correspond to the following explanatory variables: mandibular crowding, maxillary crowding, lower lip protrusion, mandibular incisor position and inclination, overbite, and overjet. The order in which the coefficients of the covariates are set to zero provides some information about their importance and predictive value. Nonetheless, the final model and optimal value of λ must be decided using cross-validation, minimizing the prediction error on the test set.[4] As a result, the LASSO procedure arrives at a selection of covariates in the model that provides the best predictive performance along with their corresponding estimated coefficients.

DIMENSION REDUCTION

Instead of selecting explanatory variables (features) to be entered into the model on the basis of their importance, we can consider reducing the number of dimensions (features) of the dataset. The 2 most well-known methods used for this purpose are *principal component analysis* (PCA) and *partial least squares* (PLS).

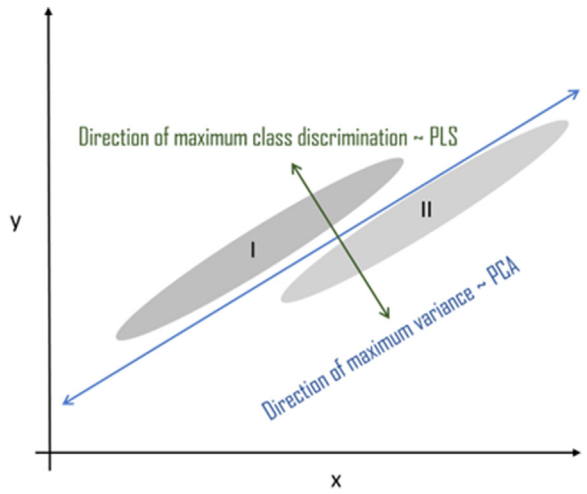

Figure 3. Dimension reduction is based on PCA (unsupervised construction of components) and PLS (supervised construction of components). The figure illustrates 2 strongly correlated covariates, x and y, and a class label (I vs II) as a response variable. The classification task is to discriminate between subjects from class I and II on the basis of their covariates. For constructing the components for PCA, the class labels are disregarded, and the direction of maximum variance is depicted in *blue*. Although this linear combination of x and y describes most of the variability observed in the data, it is not very good in discriminating label I from II. PLS considers information about the class label when constructing its components, such that a maximum separation between the class labels is possible with only a limited number of components (linear combinations of x and y). *Green arrow* indicates the linear combination with the best discriminatory power between the class labels.

The main goal of PCA is to create a lower-dimensional projection of a dataset with a smaller number of newly created variables that reflect the variability present in the original dataset.[1] Toward this aim, various linear combinations of all the covariates in the original dataset are considered, such that the combinations (called the *principal components*) "capture" as much as possible of the total variability of observations in the dataset, as explained in.[1] Next, the regression or classification problem is solved by relating a small number of highly-informative principal components to the response variable. This is called *principal components regression* (PCR). Note that this technique is applicable mainly in the regression context with continuous explanatory variables (covariates). PCR does not select any original covariates but considers a linear

combination of the original covariates (ie, principal component variables). Therefore, it is difficult to interpret the model when it is based on PCs because they have no physical meaning. This is also why comparing with classical variable subselection methods or LASSO regression is hard.

PCA constructs the components in an unsupervised manner. In particular, the dependent variable (target or response) is not used in generating the principal components. Therefore, PCR only works well when the dependent variable is associated with the principal components (which capture only the most variability among the covariates). Figure 3 presents a counterexample of such a situation in which the direction of maximum variability does not coincide with the direction of the response variable. In other words, the direction to optimally discriminate between the class labels is perpendicular to the direction of most variability; hence PCA regression will perform badly in this case. PLS also provides a dimension reduction via linear combinations of the features, but it uses the information on the dependent variable to obtain maximum class discrimination rather than capturing maximum variability. Thus, PLS will construct the components in a supervised manner.[5]

After the first PLS component is computed, the residuals for each feature are calculated. Those residuals can be interpreted as the variation the model has not yet explained. The residuals from the first stage can be used to find the second linear combination of the PLS component. This process can be performed multiple times, after which a linear regression model similar to the PCR is obtained. Figure 3 explains the difference between PCA and PLS, in which the *blue arrow* would constitute the first principal component capturing most variation in the data disregarding the class labels. The *green arrow* would constitute the first PLS component that would optimally correlate with the class label to obtain maximum class discrimination.

When PLS is applied for a single dependent variable (as in the explanation above), it is often called PLS1, and when applied to several targets, it is referred to as PLS2. There exist various extensions of PLS. For instance, orthogonal PLS aims to reduce model complexity by removing features unrelated to the response.

REFERENCES

1. Valkenborg D, Rousseau AJ, Geubbelmans M, Burzykowski T. Unsupervised learning. Am J Orthod Dentofacial Orthop 2023;163:877-82.
2. Burzykowski T, Geubbelmans M, Rousseau AJ, Valkenborg D. Validation of machine-learning algorithms. Am J Orthod Dentofacial Orthop 2023;164:295-7.
3. Burzykowski T, Rousseau AJ, Geubbelmans M, Valkenborg D. Introduction to machine learning. Am J Orthod Dentofacial Orthop 2023;163:732-4.
4. Konstantonis D, Anthopoulou C, Makou M. Extraction decision and identification of treatment predictors in Class I malocclusions. Prog Orthod 2013;14:47.
5. Valkenborg D, Geubbelmans M, Rousseau AJ, Burzykowski T. Supervised learning. Am J Orthod Dentofacial Orthop 2023;164:146-9.