Applied Scaling & Classification Techniques in Political Science

Lecture 8

Supervised aggregated classification methods

Reference

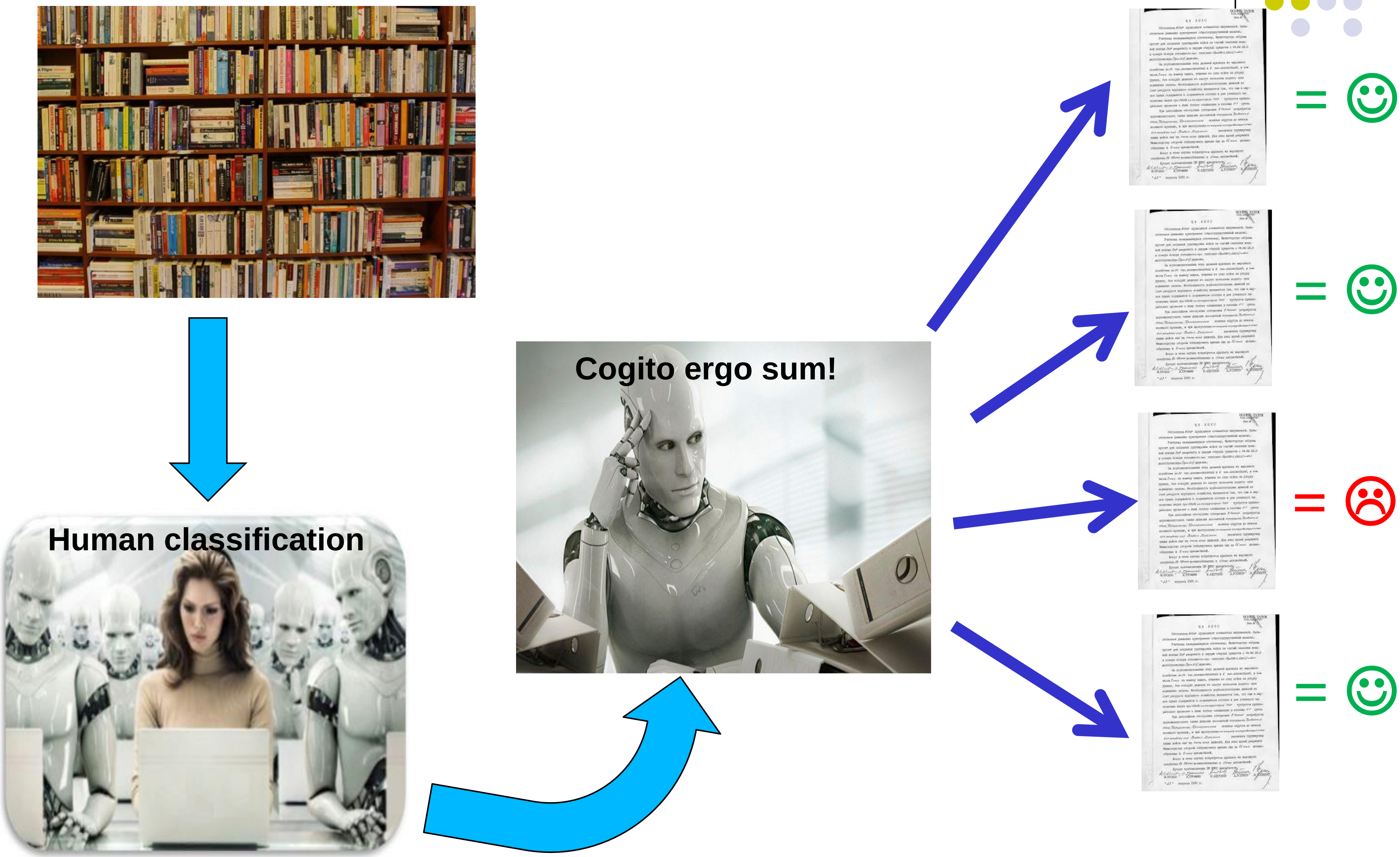
- ✓ Grimmer, Justin, and Stewart, Brandon M. (2013). “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts”. *Political Analysis*, 21(3): 267-297
- ✓ Ceron, Andrea, Curini, Luigi, Stefano M. Iacus (2016). “ iSA: a fast, scalable and accurate algorithm for sentiment analysis of social media content”, *Information Sciences*, 367–368 (1), 2016, 105–124

Measuring proportions

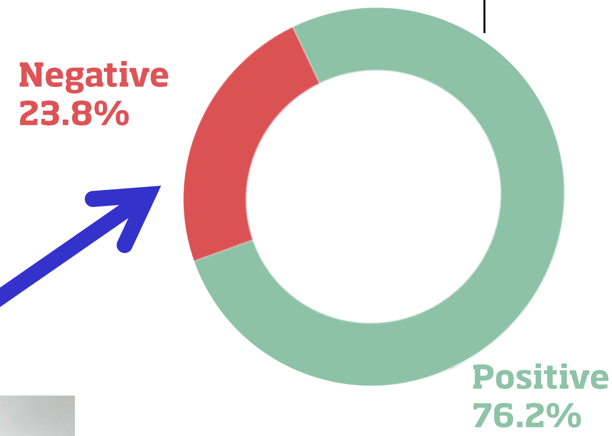
For many social science applications, only the proportion of documents in a category is needed, not the categories of each individual document

That is…

From here:
individual classification



To here: proportional classification



Measuring proportions

Shifting focus to **estimating proportions**, that is on p(**C**), can lead to substantial improvements in accuracy - even if the documents are not randomly sampled from the corpus (more on this later)

Measuring proportions

To understand how this approach actually works, we have to introduce a change in the TDM of a corpus as we discussed up to now

Now we include in the TDM an **indicator** (0/1) of whether a word occurred in a document, rather than **counts** of the words

Post	Cat	Word: nuclear	Word: fear	Word: radiation	Word: pollution	Word: waste	Word: economic
post#1	like	1	0	0	0	0	1

Using this representation, let's define a multinomial probability distribution (p(**W**)) with respect to words over all possible documents in the corpus, where $p(W_1)$ in the example above is (1,0,0,0,0,1) and is called a "word stem profile"

p(**W**) is therefore simply the proportion of documents in the corpus observed with each pattern of word profiles

Measuring proportions

The data-generating process for the documents can be written as:

$$p(\boldsymbol{W}) = p(\boldsymbol{W}|\boldsymbol{C}) * p(\boldsymbol{C}),$$

where:

$p(\boldsymbol{W}|\boldsymbol{C})$ is the proportion of words in the corpus conditional on categories and

$p(\boldsymbol{C})$ is the proportion of documents in each class in the corpus - the quantity of our interest

Measuring proportions

$$p(\boldsymbol{W}) = p(\boldsymbol{W}|\boldsymbol{C}) * p(\boldsymbol{C})$$

$p(\boldsymbol{W})$ is the distribution of the stems in the whole set (train + test). We have an **accurate estimation** here!

And what about $p(\boldsymbol{W}|\boldsymbol{C})$? It requires labeled documents - which are unavailable for the test set!

But if we assume that the conditional distributions **are identical in the training and test sets**, then we can estimate $p(\boldsymbol{W}|\boldsymbol{C})$ directly from the training-set

We have therefore also here an **accurate estimation** (as long as the coders did a good job!)

Estimating $p(\boldsymbol{C})$ is now therefore easy by solving the equation via standard regression algebra!

Measuring proportions

$$p(\boldsymbol{W}) = p(\boldsymbol{W}|\boldsymbol{C}) * p(\boldsymbol{C})$$

If we think of $p(\boldsymbol{C})$ as the unknown "regression coefficients" (the $\boldsymbol{\beta}$), $p(\boldsymbol{W}|\boldsymbol{C})$ as the "explanatory variables" matrix X, and $p(\boldsymbol{W})$ as the "dependent variable" Y, then this equation becomes the usual: $\mathrm{Y} = X\beta$ (with no error term)

From here, we can move to estimate $p(\boldsymbol{C})$ (via standard constrained least squares to ensure that elements of $p(\boldsymbol{C})$ are each in [0,1] and collectively sum to 1):

$$p(\boldsymbol{C}) = p(\boldsymbol{W}) * p(\boldsymbol{W}|\boldsymbol{C})^{-1}$$

Measuring proportions

In other words: instead of modeling the relation between features (i.e., words) and classes for **each single training document**, this approach uses a regression model that associates feature distribution ($p(\boldsymbol{W})$) with class distributions $p(\boldsymbol{W}|\boldsymbol{C})$ in the **entire training collection**

A key point is that this calculation does not require classifying individual documents into categories and then aggregating; it estimates the **aggregate proportions** $p(\boldsymbol{C})$ for target collections of unlabeled documents **directly**!

Measuring proportions

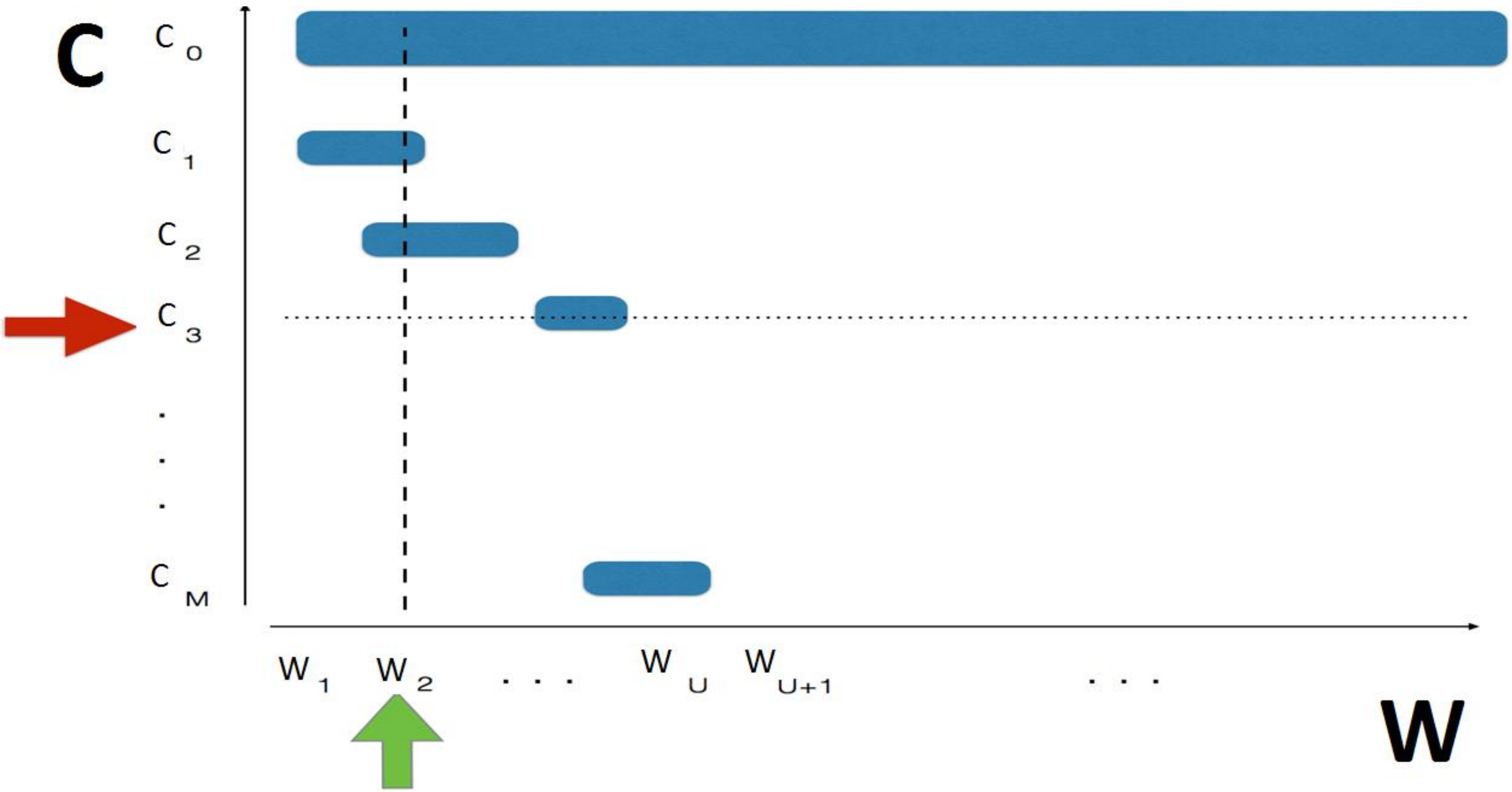
Focusing on $p(\boldsymbol{W}|\boldsymbol{C})$ rather than $p(\boldsymbol{C}|\boldsymbol{W})$ as done in the machine learning approach (remember!), has two main advantages

Theoretically: $p(\boldsymbol{W}|\boldsymbol{C})$ means: «given a post that is associated to a given content, which are the sequence of stems effectively employed to express that specific content»?

This makes a lot of sense: you do not start writing and only **afterwards** discover your sentiment toward for example a party. You start with a view, with a “category” in your mind (good, bad, support or not), and then set it out in words

Measuring proportions

Empirically: the existence of a category C_k extremely frequent in a training-set can negatively affect $p(\boldsymbol{C}|\boldsymbol{W})$ but not $p(\boldsymbol{W}|\boldsymbol{C})$



The intuition

It is easier to look at the shape of the haystack rather than trying to find a needle in it!

The intuition

Moreover…choosing a classifier by maximizing the percent correctly classified at the individual level can sometimes drastically increase the **bias of aggregate quantities**

For example, the decision rule "**war never occurs**" accurately classifies **each country-year dyad** into war/no war categories with over 99% *accuracy*, but is obviously misleading for social science research purposes!

Saying differently: a method that classifies 60% of documents correctly into one of 8 categories might be judged successful and useful for classification

However, because the individual category percentages still might be off by as much as 40 percentage points, the same classifier may be useless for some social science purposes (if individual-level errors do not cancel each other)

Measuring proportions

No statistical property must be satisfied by the training set for this approach to work properly: the training set is **not a representative sample** of the distribution of opinions in the population of texts to be analyzed!

However, the language used in the training-set to express some given concept is **assumed** to be the same as in the whole population of posts, i.e. social media users use the same language

✓ Is it a **reasonable assumption**?

Measuring proportions

After all, in the **Oxford Dictionary** (English) you have **650k terms**

In reality, for any given topic, in the everyday language there is a tendency to use at the maximum between **200 and 500 *stems***

This is what **makes possible** the statistical analysis

Measuring proportions

Of course, there are still challenges out there…

The first is the ***semantic change***, which is the difference in the meaning of language between the labeled and unlabeled sets

For example, we can have **emergent discourse**, where new words and phrases, or the meanings of existing words and phrases, appear in the unlabeled set but not the labeled set, and **vanishing discourse**, where the words, phrases, and their meanings exist in the labeled set but not the unlabeled set

Measuring proportions

Russian election hacking is an example of emergent discourse, language which did not exist a few years ago, whereas Russian Communism is an example of vanishing discourse, with language that has largely vanished from ongoing conversations over time

Measuring proportions

The second challenge is the ***lack of textual discrimination***, where the language used in documents falling in different categories is not clearly distinguishable

This problem may arise because the conceptual ideas underlying the chosen categories are not distinct

Lack of textual discrimination among categories can also occur because of heterogeneity in how authors express category-related information or a divergence between how authors of the documents express this information and how the analyst conceptualizes the categories

Measuring proportions

Validation when measuring proportions: how to do that given that you do not make any individual classification?

Well, you can still run a **cross-validation** procedure on your training-set (but ONLY at the aggregate level)!

For example, what you can do is estimating for example the **MAE** (mean average error) across categories

Supervised Aggregated approaches tend to be always better than ML at the aggregate level!

Measuring proportions

Two algorithms available for this type of analysis:

ReadMe and **iSA** (different implementations of the same idea explained above)

$$p(\boldsymbol{C}) = p(\boldsymbol{W}) * p(\boldsymbol{W}|\boldsymbol{C})^{-1}$$

- If $p(\boldsymbol{W})$ is the distribution of the stems only in the test-set, $p(\boldsymbol{C})$ is going to be estimate for the test-set only (**ReadMe** approach)
- If $p(\boldsymbol{W})$ is the distribution of the stems in the whole set (train + test), $p(\boldsymbol{C})$ is going to be estimate for both the training-set and the test-set (**iSA** approach)

Measuring proportions

ReadMe: to deal with sparsity, ReadMe solves the inverse problem saw above by subsetting of stems and averaging the results (so called: bagging procedure)

Possible problems: slow, large variability of the estimates, unstable for large dimension of D, requires further bootstrap to compute standard errors around each estimate

Measuring proportions

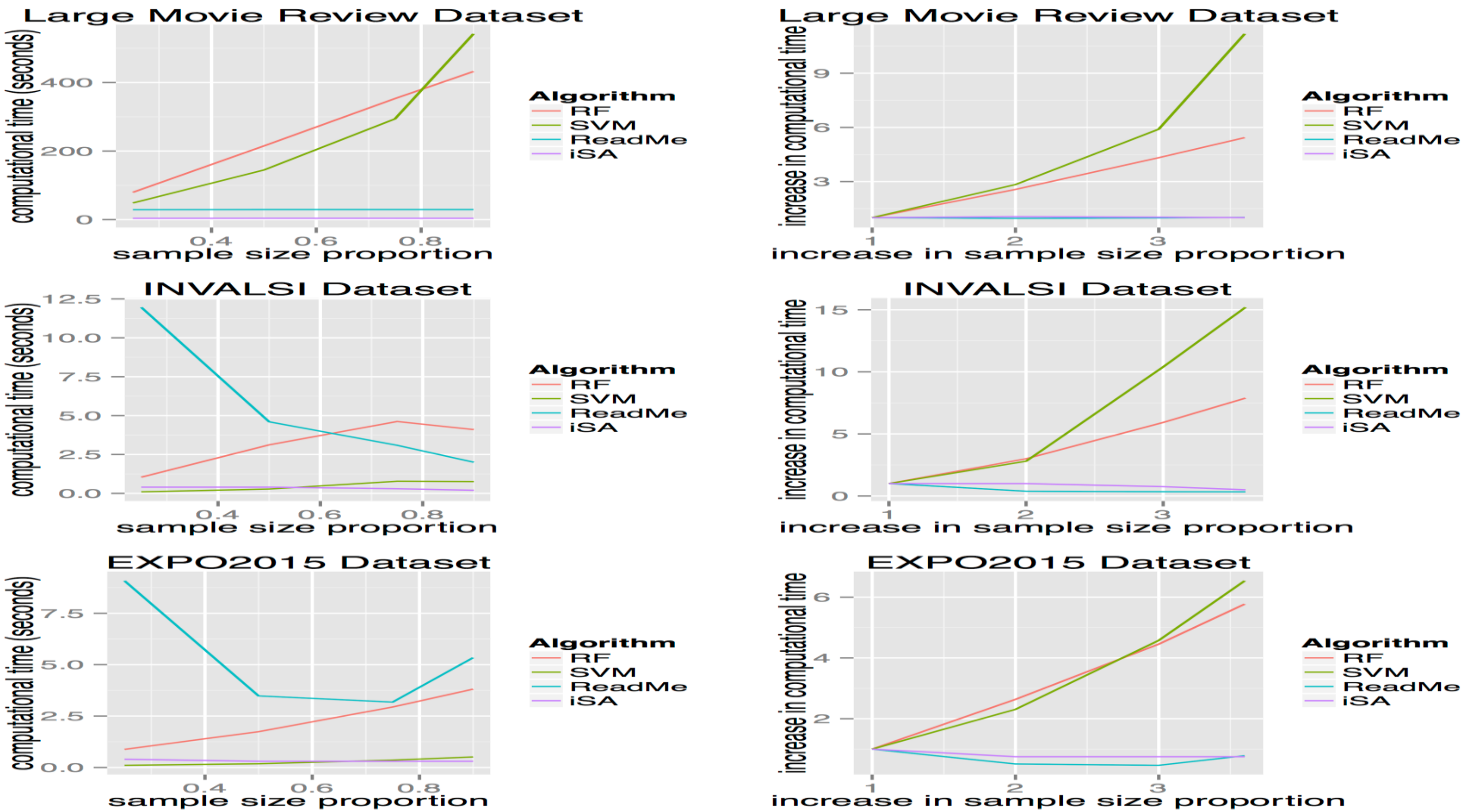
iSA: collapses the vector of stems into one-dimensional entity and solve the inverse problem in fraction of seconds

More in details (but read the Ceron, Curin and Iacus paper!):

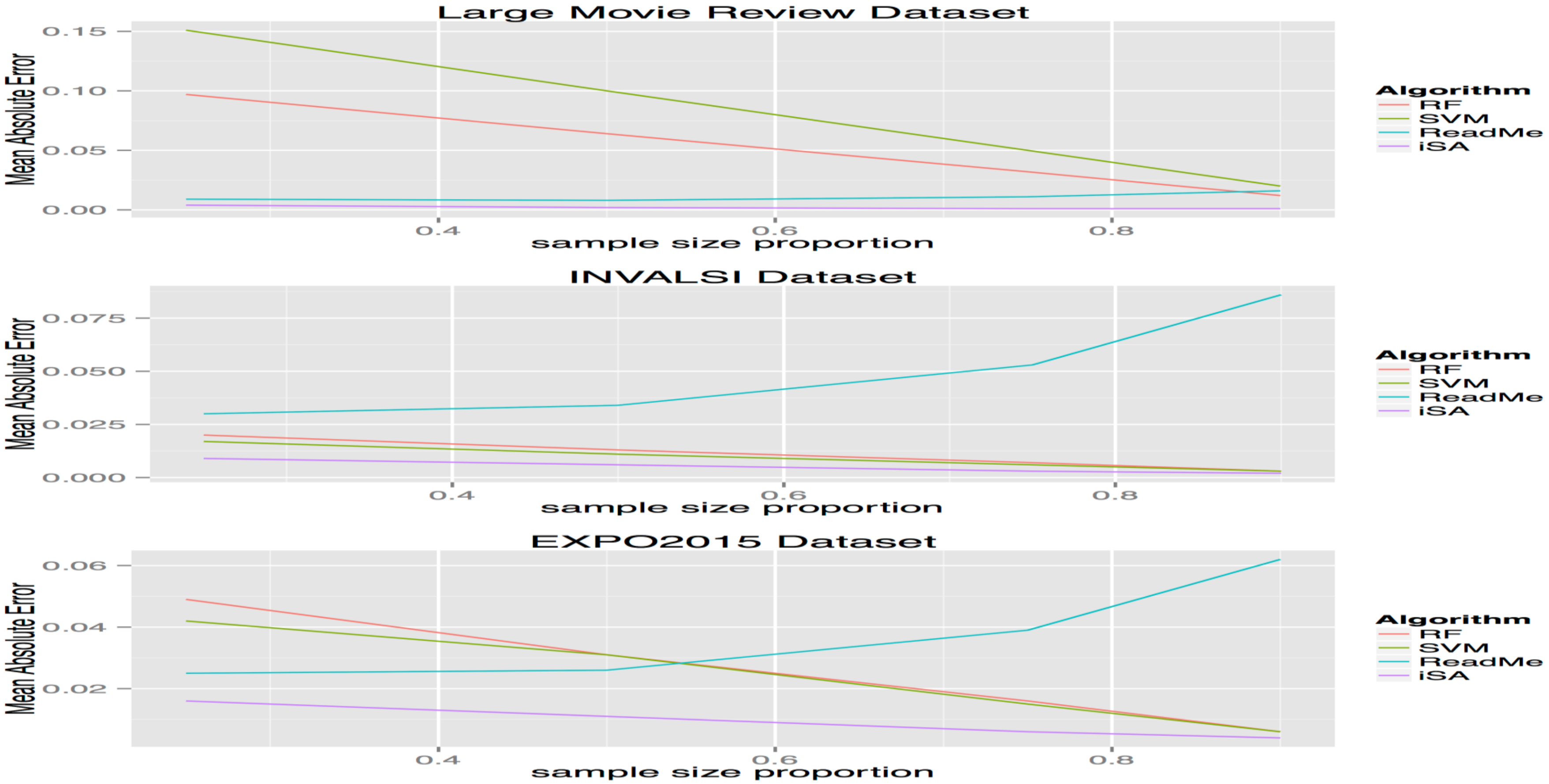
- ✓ Each vector of stems, e.g. $sj = (0, 1, 1, 0, \ldots, 0, 1)$ is transformed into a string-sequence that we denote by Cj = "0110 · · · 01"; this is the first level of dimensionality reduction of the problem: from a N x K matrix to a one-dimensional vector N x 1
- ✓ This sequence of 0's and 1's is further translated into hexadecimal notation such that the sequence '11110010' is recorded as λ = 'F2' or '11100101101' as λ = 'F2D', and so forth. So each text is represented by a label λ of shorter length

Implications: fast, memory saving (dimension reduction), reduced variability of the estimates, stable and scalable, exact standard errors are possible

Computational efficiency

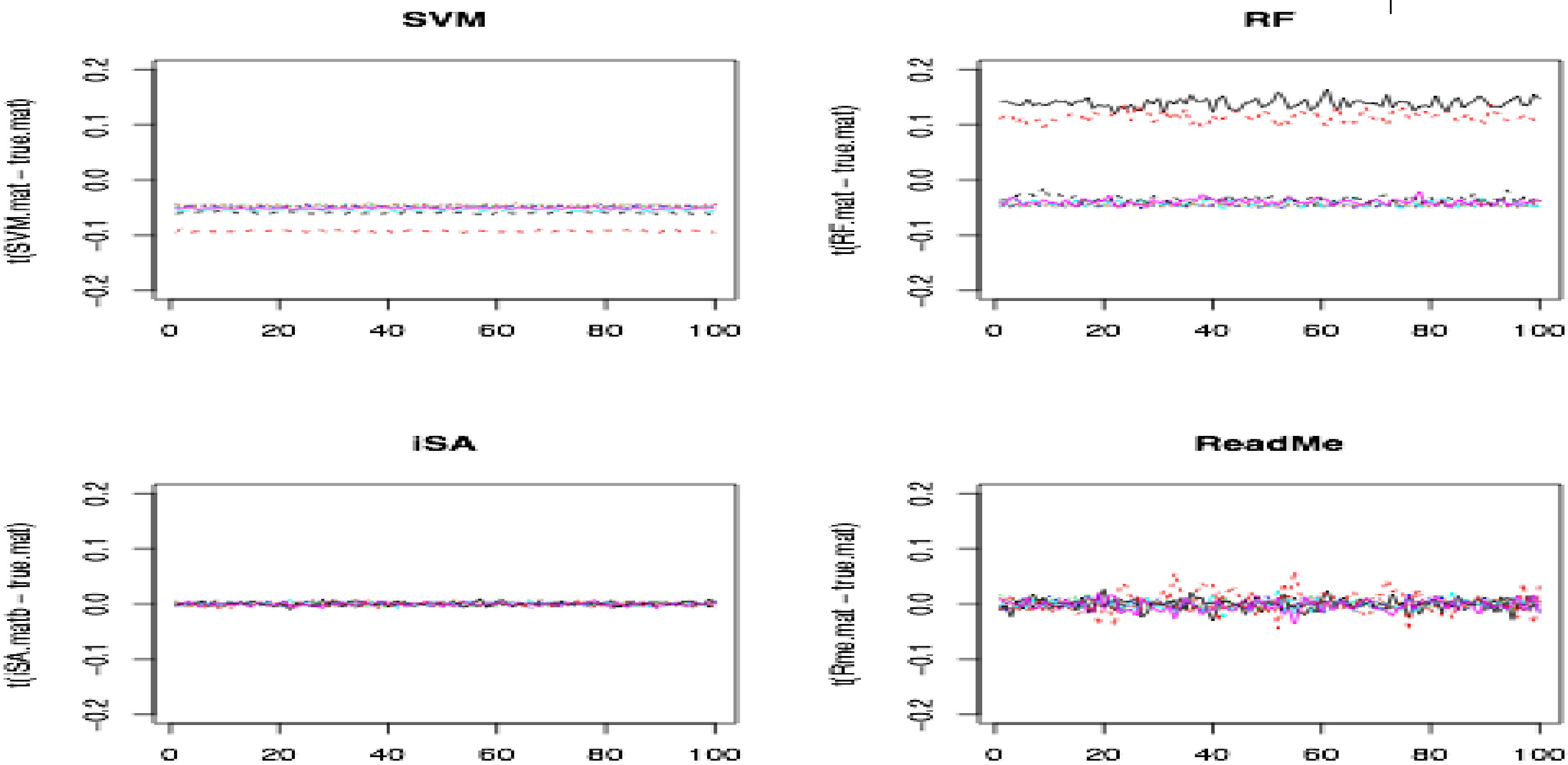


Bias



Large Movie Review Dataset
Mean Absolute Error
0.15
0.10
0.05
0.00
0.4
0.6
0.8
sample size proportion
Algorithm
RF
SVM
ReadMe
iSA
INVALSI Dataset
Mean Absolute Error
0.075
0.050
0.025
0.000
0.4
0.6
0.8
sample size proportion
Algorithm
RF
SVM
ReadMe
iSA
EXPO2015 Dataset
Mean Absolute Error
0.06
0.04
0.02
0.4
0.6
0.8
sample size proportion
Algorithm
RF
SVM
ReadMe
iSA

Variability



R pakcages to install

install.packages("VA", repos= "http://r.iq.harvard.edu", type="source")

install.packages("ReadMe", repos= "http://r.iq.harvard.edu", type="source")

library(devtools)

install_github(“blogsvoices/iSAX”)