Google Play Application Install Approximations

Predicting Installs based on Rating and Number of Reviews

Joseph Padilla and Ashish Tiwari

Department of Mathematical Sciences
Binghamton University

December 3, 2018

Outline

- Purpose and Benefit
- Data Set and Data Frame Cleaning
- Model Creation and Analysis
  - Initial Model
  - Testing Linearity of Model
  - Improvements based on Tests
- Future Applications and Benefits of Model
- Further Model Refinement

Purpose of Analysis

- In the current age, developers can make a living creating applications which, with enough installs, can generate substantial revenue.
  - Advertisements
  - Paid Application
  - In App Purchases
- Android is the most popular and most widely used cell phone operating system in the world.
  - Creating applications which can run on older phones could substantially increase installs.
- Understanding the general trend of installations could be beneficial to a developer.

Data set and Data frame Cleaning

- Our data set was found on Kaggle, a website which allows people to upload data sets for testing, training, and modeling.
- Our Google Play data set contains 9660 unique applications each with a plethora of data. Only several of these are of interest to this model: `Installs`, `Reviews`, `Rating`, `Price`, and `Category`.
    - Items to consider for future models include `Size`, `Last Updated`, `Android Ver`, and `Current Ver`.
- The entire data frame when exported from Excel to R contained `string`-class objects. This is very unfortunate because they cannot be modified until they have a numeric class.
- Column-wide character deletions were used to rid non-numeric characters so that each `string` could be changed to a `double`. Columns then changed to `double`.

Data set and Data frame Cleaning

- We now had to determine what values are to be considered harmful to our model. What does this mean?
  - Repetitions of the same data point are important to delete since this weighs those points more than other points.
  - Allowing Installs below a certain amount is something that can skew all data due to the nature of how ratings work.
  - Considering apps with do not have a rating is entirely pointless since they have next to next to zero installs and reviews.
- Using the above criterion, we delete all repetitions of applications, all applications which have no rating, and all applications which have below 101 installs.
- We next split the data into 80% training and 20% testing.

Initial Model

- We begin by considering the simplest model that seems intuitively meaningful.

$$\log_{10} \text{Installs} \;=\; \beta_0 \;+\; \beta_1 \text{Rating} \;+\; \beta_2 \log_{10}(\text{Reviews} + 1)$$

.

- For intuitive reasons, $\log_{10}$ is used over natural $\log$. In addition, a $+1$ is added to the reviews term to avoid taking $\log_{10} 0$.
- This model yields very promising initial results with $R^2_{Adj} = .09192$ and all parameters being significant.

Initial Model

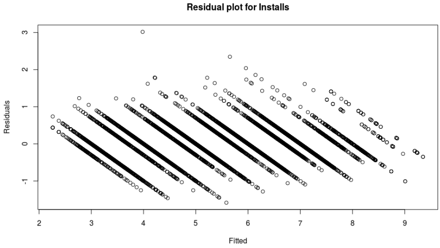
```
Call:
lm(formula = log10(Installs) ~ Rating + log10(Reviews + 1), data = trainingdata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.58533 -0.26965  0.00162  0.26576  3.01747

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         2.994087   0.043845   68.29   <2e-16 ***
Rating             -0.234904   0.010882  -21.59   <2e-16 ***
log10(Reviews + 1)  0.927209   0.003358  276.08   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.4247 on 7028 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.9192,Adjusted R-squared:  0.9192
F-statistic: 3.999e+04 on 2 and 7028 DF,  p-value: < 2.2e-16
```

Figure: Summary for Initial Model

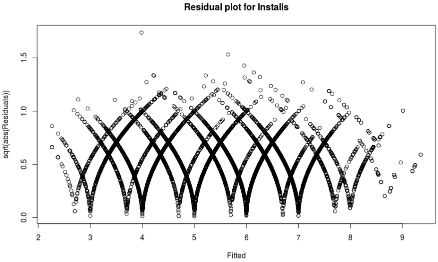
- We now run our usual gamut of tests.
  - Variance Inflation Factor
  - Constant Variance Assumption
  - Normality Assumption
  - Leverages
  - Influential Points
  - Outliers

- The VIF $= 1.092264$ indicates that our parameters are nearly perfectly uncorrelated.

Analysis of Initial Model



- The residuals here appear as lines
- This is alright because they are appear to be normally distributed along those lines
- These do look any better, but its due to the discrete nature of the data
- Adding noise can make these appear to look a little better, but it is unwise so plots were left out.



Family of models

- As dimensionality increases, the number of parameters increases quadratically.
  - e.g. If 5 groups are being estimated for data with dimensionality 10, 275 parameters have to be estimated.

- Constraints can be imposed such that $\mathbf{T}_g$ and $\mathbf{D}_g$ are equal or different across groups, and $\mathbf{D}_g$ is anisotropic or isotropic, resulting in a parsimonious family of 8 models.

Model	$\mathbf{T}_g$	$\mathbf{D}_g$	$\mathbf{D}_g$	Parameters
EEA	Equal	Equal	Anisotropic	$p(p-1)/2+p$
VVA	Variable	Variable	Anisotropic	$G[p(p-1)/2]+Gp$
VEA	Variable	Equal	Anisotropic	$G[p(p-1)/2]+p$
EVA	Equal	Variable	Anisotropic	$p(p-1)/2+Gp$
VVI	Variable	Variable	Isotropic	$G[p(p-1)/2]+G$
VEI	Variable	Equal	Isotropic	$G[p(p-1)/2]+1$
EVI	Equal	Variable	Isotropic	$p(p-1)/2+G$
EEI	Equal	Equal	Isotropic	$p(p-1)/2+1$

Yeast data

- Chu *et al*. (1998) measured expression levels of 6118 genes during sporulation over seven time points.
- A $G = 5$ component model is selected.
    - Closer to the findings of Mitchell (1994) and Chu *et al*. (1998).
    - Contrary to Wakefield *et al*. (2003) and McNicholas and Murphy (2010) who found around 11–14 components.
- A *t*-distribution allows less “tightness” around the mean of the time course and is therefore more accommodating heavier tails.
- Details in
    - McNicholas, P. D., & Subedi, S. (2012). Clustering gene expression time course data using mixtures of multivariate t-distributions. Journal of Statistical Planning and Inference, 142(5), 1114-1127.

Let's take a look at the $G = 5$ component solution

yeastplot2.pdf

Figure: The temporal patterns for the five clusters fitted by our best model on the yeast data.

Key References

- Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, **28**, 781–793.
- Chen, J. and Li, H (2013). Variable selection for sparse dirichlet-multinomial regression with an application to microbiome data analysis. *The annals of applied statistics*, **7**(1), 418–442.
- Chu, S., DeRissi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., and Herskowitz, I. (1998). The Transcriptional Program of Sporulation in Budding Yeast. *Science* **282 (5389)**, 699–705.
- McGrory, C.A. and Titterington, D.M. (2007). Variational approximations in Bayesian model selection for finite mixture distributions. *Computational Statistics and Data Analysis* **51**, 5352–5367.
- McNicholas, P.D. and Murphy, T.B. (2010) Model-based clustering of longitudinal data. *Canadian Journal of Statistics* **38(1)**, 153–168.
- Pourahmadi, M. (2000). Maximum likelihood estimation of generalized linear models for multivariate normal covariance matrix.*Biometrika***87**, 425–435.
- A. Rau, G. Celeux, M.-L. Martin-Magniette, C. Maugis-Rabusseau (2011). Clustering high-throughput sequencing data with Poisson mixture models. Technical report RR-7786, Inria Saclay – Ile-de-France.
- Subedi, S. and P. D. McNicholas (2013). A variational approximations-dic rubric for parameter estimation and mixture model selection within a family setting. arXiv:1306.5368.
- Subedi, S., Punzo, A., Ingrassia, S. and McNicholas, P. D. (2013). Clustering and classification via cluster-weighted factor analyzers. *Advances in Data Analysis and Classification* **7**(1), 5–40.
- Subedi, S. and McNicholas, P. D. (2014). Variational Bayes Approximations for Clustering via Mixtures of Normal Inverse Gaussian Distributions.*Advances in Data Analysis and Classification* **8**(2), 167–193.
- Subedi, S., Punzo, A., Ingrassia, S. and McNicholas, P.D. (2015). Cluster-Weighted t-Factor Analyzers for Robust Model-based Clustering and Dimension Reduction. *Statistical Methods and Applications*. To appear.

Thank you!