Final Project Write-Up

Alex Xiaotong Gui

5/4/2018

Unsupervised Learning

In the previous write-ups, our analysis concerns predictions: given a set of explanatory variables $X_1, ...X_{p-1}$ and n observations, how can we predict the response variable Y? We have adopted various regression techniques and yielded satisfying results. However, in this part of analysis, we want to shift our focus away from the Y variable and ask ourselves: what are something interesting things we can say about the Xs? What are some underlying structures and how can we present them efficiently? We introduce the concept of unsupervised learning, "a set of statistical tools intended for the setting in which we have only a set of features X1, X2, . . . , Xp measured on n observations."

Principal Components Analysis(PCA)

Principal components analysis is widely used as an unsupervised learning method for feature extraction and data compression. In our analysis, we will apply principal components in our regression model as a dimensionality reduction technique. The intuition behind PCA is: given a set of highly correlated predictors, PCA will transform it into a smaller set of linearly independent variables called principal components. The transformation is defined such that the first principal component direction captures the greatest possible variability in the data, in others words, explain the most variability of the data. The succeeding principal components are linear combinations of the variables that is un- correlated with the preceding component and has largest variance subject to this constraint. The set of components constitute a basis for our data space.

Principal Components Regression

The principal components regression approach will first construct M principal components and then regress on the components instead of individual predictors. The underlying assumption of the model is "the directions in which $X_1, ...X_p$ shows the greatest variance are those associated with Y". Although this assumption is not guaranteed, it regardless provides a decent approximation that often yields good results. M, the number of principal components, is our tuning parameter that will be chosen by cross-validation.

We believe PCA works well with our pokemon data given the existence of strong correlation among our predictors. Our analysis shows that PCR greatly reduced variance and contributed to our model's predictive power.

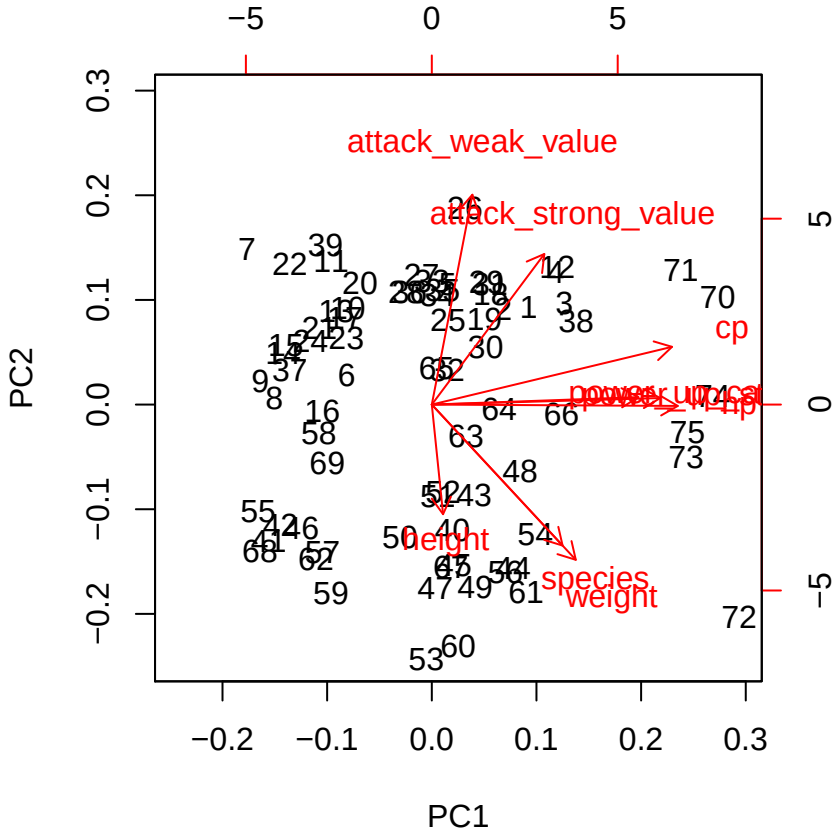
Principal Components

Our model first constructs 9 principal components(this makes sense since $p = 9$ and $M \leq p$).

```
## Standard deviations:
## [1] 2.000 1.379 1.164 1.023 0.581 0.417 0.304 0.243 0.191
##
## Rotation:
##                        PC1     PC2    PC3      PC4     PC5     PC6     PC7
## species             0.2539 -0.3970  0.238 -0.52512  0.3707 -0.2008 -0.4838
## cp                  0.4664  0.1625  0.119 -0.00223 -0.1473  0.3512  0.1679
## hp                  0.4776 -0.0038 -0.106  0.03351 -0.0244  0.4642 -0.2838
```

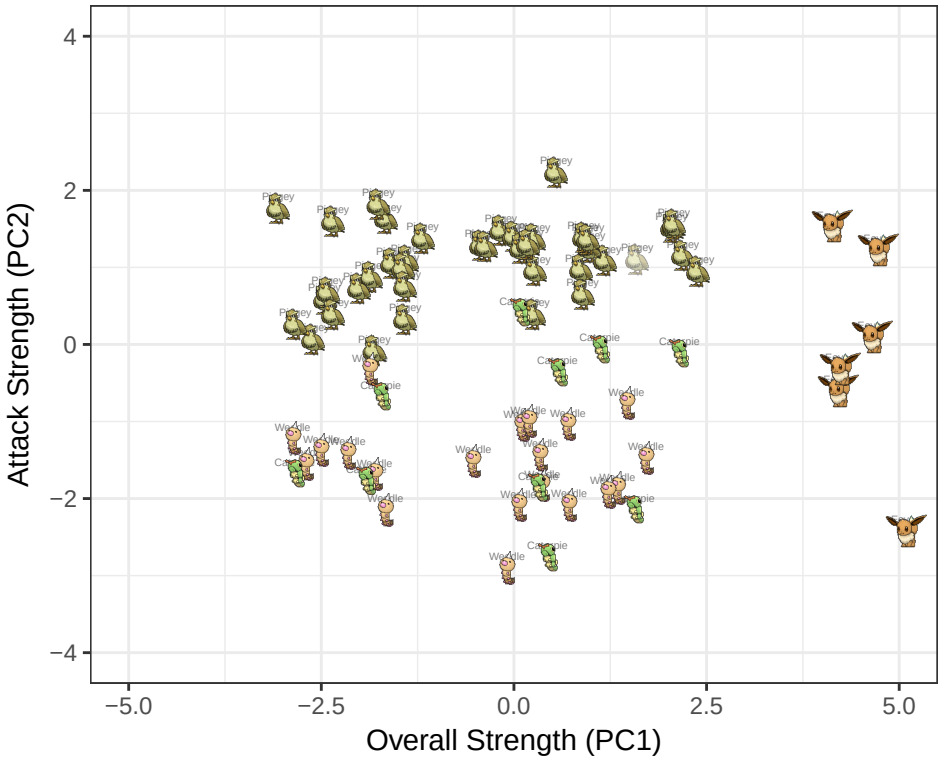
```
## weight              0.2795 -0.4370  0.453  0.01056  0.0396 -0.0293  0.6677
## height              0.0219 -0.3086  0.287  0.81195  0.0283 -0.0865 -0.3673
## power_up_stardust   0.4434  0.0184 -0.354  0.12697  0.0397  0.0742 -0.0423
## power_up_candy      0.4011  0.0209 -0.426  0.12977  0.0511 -0.6973  0.1514
## attack_weak_value   0.0787  0.5910  0.286  0.14366  0.7268 -0.0244  0.0783
## attack_strong_value 0.2185  0.4239  0.491 -0.10070 -0.5527 -0.3466 -0.2137
##                         PC8     PC9
## species             -0.1821  0.0209
## cp                  -0.6273  0.4185
## hp                   0.6523  0.1931
## weight               0.2265 -0.1601
## height              -0.1271  0.0580
## power_up_stardust   -0.2290 -0.7748
## power_up_candy       0.0918  0.3465
## attack_weak_value    0.0754 -0.0396
## attack_strong_value  0.1171 -0.1902
```

To visualize it:



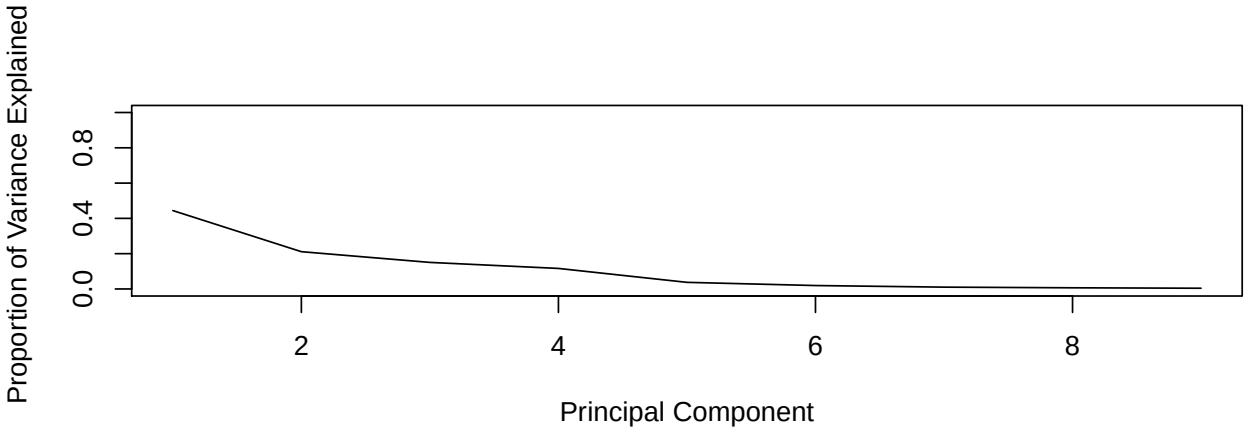
Look at PC1. We observe that the higher the performance metrics, the higher the PC1 value. Therefore we can interpret PC1 as a measurement of overall strength. As for PC2, we notice that higher PC2 is associated with higher attack value. Therefore we interpret PC2 as a measurement of attack strength.

We then plot our pokemon on our new principal components space:



Interesting Insights: Using principal components, we get to create two new powerful metrics to evaluate our pokemons. From the plot, you can observe that Eeeve in general has high overall strength and high attack strength. Pidgey has good attack stats but is weaker in general due to the creature's intrinsic nature. Caterpie and Weedle are weak on both metrics. Overall, this PCA gives you a high-level overview of our pokemon's strength. You should feel excited!

Regression:

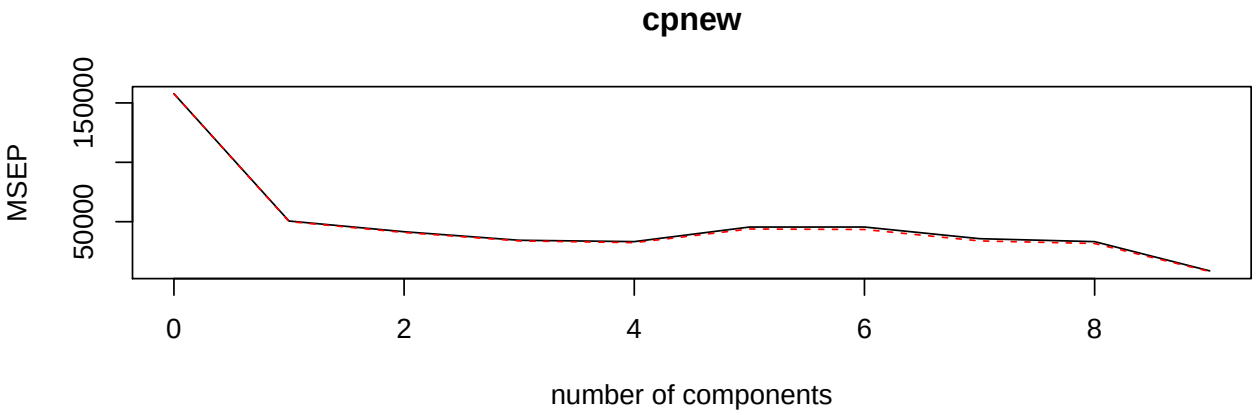


Observe that the first two principal components explain nearly 60% of the variability in the data. As $M \to 10$, the marginal contribution to variability explained decreases. Our regression model will use cross validation to tune M, the number of components as predictors.

```
## Data:    X dimension: 50 9
##  Y dimension: 50 1
```

```
## Fit method: svdpc
## Number of components considered: 9
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           397.1    224.9    203.8    185.4    182.3    213.3    213.4
## adjCV        397.1    223.8    202.3    184.0    179.9    209.4    208.5
##        7 comps  8 comps  9 comps
## CV         189    182.3    92.81
## adjCV      184    177.9    89.94
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X        45.77    65.32    80.62    91.87    96.18    98.04    98.98
## cpnew    70.69    79.25    84.37    87.68    88.05    90.30    93.22
##        8 comps  9 comps
## X        99.56   100.00
## cpnew    94.48    98.72
```

The metric of cross-validation is *root_mean_squared_error*. Ideally, we want to pick M where the CV score is minimized.



We compute the test MSE as follows using $M = 9$ components:

```
pcr.pred = predict(pcr.fit, pokemondata[-train, ], ncomp = 9)
mean((pcr.pred - pokemondata[-train, ]$cpnew)^2)
```

```
## [1] 2827
```

Interpretation

One disadvantage of PCR is that the model is not intepretable. Given the linear relationship of the components, we can't say much on the individual predictors themselves. PCR is also not available for feature selection. Therefore although we acknowledge the great predictive power of PCR, we don't think it is the most helpful model for pokemon users to understand pokemon's performance metrics.

Random Forest?Regression Tree?

```
## rf variable importance
##
```

```
##                      Overall
## cp                     100.0
## hp                      74.0
## species                 72.1
## attack_strong_value     64.7
## power_up_stardust       63.3
## weight                  48.7
## power_up_candy          46.2
## attack_weak_value       35.5
## height                   0.0
```

```
## [1] 2685
```

```
rf_default$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 15525
##                     % Var explained: 89.8
```