Gmail **Andrew Ho <kironide@gmail.com>**

Week 2 Day 1 (Morning)

3 messages

Jonah Sinick <jsinick@gmail.com> Mon, Feb 22, 2016 at 9:59 AM
To: Ali Bagherpour <ali.bagherp@gmail.com>, Andrew Ho <Kironide@gmail.com>, Chad Groft <clgroft@gmail.com>, David Bolin <david@bolin.at>, Jacob Pekarek <jpekarek@trinity.edu>, Jaiwithani <jaiwithani@gmail.com>, James Cook <cookjw@gmail.com>, Linchuan Zhang <email.linch@gmail.com>, Matthew Gentzel <magw6270@terpmail.umd.edu>, Olivia Schaefer <taygetea@gmail.com>, Sam Eisenstat <sam.eisenst@gmail.com>, Tom Guo <tomguo4@gmail.com>, Trevor Murphy <trevor.m.murphy@gmail.com>

Pairings:

Jacob <---> Andrew
Jai <---> Trevor
Matt <---> Ali
James <---> Tom
Olivia <---> Chad
David <---> Linch

- If you haven't yet taken the survey: https://docs.google.com/forms/d/1AfPWP3_WAumxfz-HVaf1i43laQSsoE6KBb2pHL2yOgc/viewform please do.

  Respond to this message indicating how far you've gotten in *Advanced R*.

- Take a look at the Kaggle African Soil competition.
  https://www.kaggle.com/c/afsis-soil-properties
  Read the instructions, and download the train and test sets.

  (1) Construct a data frame giving the correlations between the targets
  "Ca" "P" "pH" "SOC" "Sand"
  and use order() to inspect the strongest positive or negative correlations between the spectral bands and the targets.

  (2) For each target, graph the spectral wave numbers against the target.

  (3) Use regularized linear regression (cv.glmnet) to predict each target in terms of the wave bands and the other features in the dataset (other than the targets). Experiment with alpha = 0 (ridge regression), alpha = 1 and alpha =0.25, 0.5 and 0.75, generating cross validated predictions with

```
tarDF = as.data.frame(target)
tarDF$predictions = 0
set.seed(1); folds = sample(1:nrow(df)) %% 10
for(i in unique(folds)){
  trainFeatures = features[folds != i,]
  testFeatures = features[folds == i,]
  trainTarget = target[folds != i]
  # train model on trainFeatures and trainTarget using cv.glmnet
  # Generate predictions preds
  # tarDF[folds == i,"predictions"] = preds
}
```

  and computing the Root Mean Squared Error, which is closely related to R^2 (see difference between R square and rmse in linear regression).

For each feature, pick the best alpha.

Generate predictions for the test set, and submit them to Kaggle, tosee how you score.

Andrew <kironide@gmail.com> Mon, Feb 22, 2016 at 10:01 AM
Reply-To: Kironide@gmail.com
To: Jonah Sinick <jsinick@gmail.com>

I've gotten through chapters 1-11 in Advanced R, including part of 12 but skipping a couple sections here and there (like the one about S4 classes in the OO primer).
[Quoted text hidden]

Jonah Sinick <jsinick@gmail.com> Mon, Feb 22, 2016 at 2:33 PM
To: Ali Bagherpour <ali.bagherp@gmail.com>, Andrew Ho <Kironide@gmail.com>, Chad Groft <clgroft@gmail.com>, David Bolin <david@bolin.at>, Jacob Pekarek <jpekarek@trinity.edu>, Jaiwithani <jaiwithani@gmail.com>, James Cook <cookjw@gmail.com>, Linchuan Zhang <email.linch@gmail.com>, Matthew Gentzel <magw6270@terpmail.umd.edu>, Olivia Schaefer <taygetea@gmail.com>, Sam Eisenstat <sam.eisenst@gmail.com>, Tom Guo <tomguo4@gmail.com>, Trevor Murphy <trevor.m.murphy@gmail.com>

Some more:

- Check out the reading on cross validation and regularized linear regression that I sent you last Friday. It's not essential that you follow everything, but you should read enough to feel comfortable with what's going on.

- *Applied Predictive Modeling* (referenced below) introduces the "caret" package. You can use this in conjunction with glmnet as described in

  http://stats.stackexchange.com/questions/69638/does-caret-train-function-for-glmnet-cross-validate-for-both-alpha-and-lambda

  Whereas cv.glmnet picks an appropriate sequence of lambdas, the caret package does not, see, e.g. http://stats.stackexchange.com/questions/88756/r-how-to-let-glmnet-choose-lambda-range-when-using-caret. I suggest running cv.glmnet to get a rough sense for what the range of lambdas should be, and then use that when doing grid search with caret.

  If cross validation becomes painfully time consuming, you can reduce the range of lambdas and range of alphas checked, or use a subset of the data (see below).

- Note that there's a single **categorical** variable "Depth" in the data, with levels "Topsoil" and "Subsoil".

  The most straightforward way of dealing with this is to encode the variable as a dummy variable and use it in the model. But the model then won't take into account potential variation in the strength of single of bandwidths by category (something that may or may not result in substantially worse predictive power).

  Another way of incorporating this into the model is to take all features, scale them, and then for each scaled feature construct a feature that's 0 for "Topsoil" and the original value for "Subsoil."

  Yet way is to train separate models for samples from "Topsoil" and from "Subsoil," then train a model on all samples, then fit a linear model with the three predictors

  SubsoilPredictions, TopsoilPredictions, PredictionsFromAll.

Have fun!
Jonah

==========

From Friday:

Some of you are looking to learn more about regularization and cross validation. For this, I recommend:

- **Cross Validation:** Chapter 4 of *Applied Predictive Modeling* and Chapter 5 of *Introduction to Statistical Learning* (both also uploaded to Google Drive).
- **Regularized Linear Regression:** Section 6.4 of *Applied Predictive Modeling*, Section 6.2 of *Introduction to Statistical Learning* andLecture 7 from Andrew Ng's Coursera course on machine learning.

Try multiple sources and see which one you find it easiest to learn from.

On Mon, Feb 22, 2016 at 9:59 AM, Jonah Sinick <jsinick@gmail.com> wrote:
[Quoted text hidden]