Soybean Marathon Match 6

AdvancedEliteClassifier

Solution Descriptions

1. Overview

1.1 Key Thoughts

By reading the contents in the PowerPoint presentation, these important points can be extracted:

a) Superior yield is the most important. An advanced experimental usually has to out yield the checks.
b) Taking relative maturity into account when discussing yield performance is critically important. When comparing an experimental variety's performance to checks, it is critical they be close to the same maturity, or it becomes an unfair comparison.
c) An experimental variety that is the same maturity as a check variety and out yields the check is special and will be advanced, if the performance advantage is high enough (3% might be advanced, while 5% will likely be advanced).
d) An experimental variety that is earlier than a check variety and out yields the check is very special and will be advanced.

1.2 Reused Ideas

The general structure of the estimation work is based on the place 1 solution to Soybean Marathon Match 1 “EliteClassifier”. These ideas are reused:

a) A variety’s evaluation score is mainly determined by comparison with the other varieties in the same experiment.
What’s different from the original solution is that the basic comparison unit is not a trial (experiment-loccd-rep tuple) but an experiment. Instead of comparing the relative yield among trails, average yield is used to evaluate the score of each variety in a whole experiment.
b) The comparisons with checks are more important than non-checks.
c) Several optimization ideas (e.g. amount of checks, amount of trails, etc.).
d) Scoring standard is also reused along with things above, such as -10^{10} for checks and -10^{9} for varieties with the type RR2Y. Please refer to the corresponding steps (in section 2.3 “Classifying Phase”).

2. Implementation

2.1 Data Model

A total of 8 data types (other than `AdvancedEliteClassifier`) are used to describe the business data in the problem:

Name	Type	Important Members
MaturitySubZone	enum	Early, Mid, Late
VarietyType	enum	Unknown, Conventional, RoundupReady, RoundupReadyToYield
VarietyStats	struct	averageYield, maturityCount, averageMaturity, podColorsCount, pubescenceColorsCount, flowerColorsCount, emergencesCount, plantHeightsCount, score
Trail	struct	loccd, rep, yield, maturityNumber, podColor, pubescenceColor, flowerColor, emergence, plantHeight
Variety	struct	varietyId, type, relativeMaturity, isCheck, isElite, trails, stats
Experiment	struct	experimentId, year, varieties, checkCount
PhysicalLocation	struct	loccd, year, zone, band, subzone
AttributeStat	struct	total, elite

2.2 Process

The tasks that the `classify(vector<string>, vector<string>, vector<string>)` method does are:

1. Initialize the data fields in the `AdvancedEliteClassifier` instance;
2. Generate statistics by analyzing the training data and location data;
3. Load the real test data;
4. Do the classifying work.

Standard output is used to print out the status of the program and also provide performance data (time spent for each step) to guarantee that the speed of the algorithm is acceptable.

2.3 Training Phase

During the training phase (`AdvancedEliteClassifier::train`), all training data and location data in CSV format are parsed as instances of the data types defined above. The information that the training phase collects for later use includes:

a) Physical locations;

b) Attribute stats (total occurrences and elite occurrences) for: experiment year, variety type, amount of trails for a single variety, amount of check in a single experiment, pod color, pubescence color, flower color, emergence, plant height and harvest lodging;

c) Average maturity.

2.4 Classifying Phase

The steps for evaluating each variety are:

1. Building stats for the variety (`AdvancedEliteClassifier::buildVarietyStats`). This includes generating average yield and average maturity of the variety in different trails. Values and their occurrences of the other attributes like pod color and flower color are also analyzed and stored. (By the way, some other data is also collected but not used in the final solution - such as relative maturity, location data, standard deviations of yield and maturity number, etc. Tried to utilize them to improve the results but haven't found an effective way.)
2. If the variety is a check: `score = -10`^{10} and break
3. If the variety is RR2Y, conventional before 2002, or with NULL type: `score = -10`^{9} and break

4. If the variety has at least one valid plant height which is < 40: score = -10^8 and break
5. If the variety has an emergence of 7 or 8 in any trail: score = -10^8 and break
6. If the variety has a pod color of 3, 4, 5 or 7 in any trail: score = -10^8 and break
7. If the variety has a pubescence color of 5 or 6 in any trail: score = -10^8 and break
8. If the variety has a flower color of 4 in any trail: score = -10^8 and break
9. Calculate the base score of the variety by comparison with the other varieties in the same experiment, as described below:

```
For each variety X
    win = 0.0
    total = 0.0
    For each other variety Y in the same experiment
        weight = is Y check ? 3 : 1
        If the difference of relative maturity between X and Y:
            >= 1.5: weight *= 0
            1.0 - 1.5: weight *= 0.1
            0.5 - 1.0: weight *= 0.25
        If averageYield of X is more than Y:
            If averageYield of X / Y > 1.06:
                win += weight * (averageMaturity of X < Y ? 2.0 : 1.7)
            Else if averageYield of X / Y > 1.03:
                win += weight * (averageMaturity of X < Y ? 1.8 : 1.5)
            Else
                win += weight * (averageMaturity of X < Y ? 1.2 : 1.0)
        Else if averageYield of X and Y are close:
            win += weight / 3
        total += weight
    win = pow(win, 1.5)
    total = pow(total, 1.5)
    score = win / total * $10^7$
```

10. If type is conventional and year is not 2005: score *= 0.85
11. If type is RR1 and year is 2002 or 2006: score *= 1.05
12. If year is 2003: score *= 1.03
13. If the amount of trails is
    < 3: score *= 0.7
    7-9: score *= 1.03
14. If the amount of trails is (only if the total occurrence of this amount in test data is more than 1000)
    1-6: score *= 1.0 + elite ratio of this amount * 0.5
    > 6: score *= 1.0 + elite ratio of this amount
15. If the amount of non-NULL maturity number is
    < 10%: score *= 0.7
    > 90%: score *= 1.05
16. If the amount of check(s) in this experiment is:
    4: score *= 1.1

    5: `score *= 1.15`
    6-11: `score *= 1.1`

17. `score *= 1.0 + 5 * pow(elite ratio of the year, 0.3)`
18. If the variety has at least one pod color of 9: `score *= 0.95`
19. If the variety has valid maturity numbers, calculate yieldPerMaturity as averageYield / averageMaturity. If yieldPerMaturity < 0.4: `score *= 0.9`

About the conditions, parameters and their values:

- Some are reused from the place 1 solution to EliteClassifier, such as 2, 3, 10, 11, 13, 15, 16 and 17.
- Some are based on results of analysis on the training data, such as 4, 5, 6, 7, 8, 12, 14, 18 and 19. In general they are all related to the elite ratio (elite / total) for each corresponding attribute.
  For example, it has been found that the average yieldPerMaturity of elite varieties are much higher than non-elite varieties, and 0.4 is the average yieldPerMaturity of the non-elite varieties. That's why <0.4 is used as a condition in step 19 to reduce the score.
- The parameters in the calculations of step 9 are tuned manually. The logic is based on the knowledge described in the section 1.1 "Key Thoughts".

This covers the whole evaluation process for each variety. In the end the varieties are sorted by score, higher is better.