Capstone Report MLND

Estevao Vieira

March 2017

1 Definition

1.1 Project Overview

Bright Yellow cells are a model cell line for studying plants. They are largely used in cell cycle analysis, such as morphology and differential gene expression. To estabilish relations between gene expression during the cell cycle and the plant's phenotype, we have to understand the expression pattern of the gene of interest.

While the technology needed for single cell expression analysis is not yet stabilished for plants, it is possible to determine this expression pattern by using populations of cells. For this analysis to be reliable, the population has to be majoritarily in the same cell cycle phase. For this, a cell cycle synchronization is commonly used[1].

To make any conclusions acceptable, one crucial step is that of the syncronization validation (i.e. showing that the cells were in fact well synchonized), for without sure that the procedure worked and the cells were synchronized, any later result is meaningless[1].

1.2 Problem Statement

For the synchronization validation, in the case of tobacco BY-2 cells, populations of cells have to be observed after 8 to 10 hours after the end of procedure and the proportion of cells in the mitotic stage has to be close to 70%. The present project aims at automating most of this process of validating by detecting if a given BY cell is in the mitotic stage. Two binary classifications have to be applied in sequence:

1. Photo classifier: Separate good photos from bad photos (this is made clearer from the dataset explained below)

2. Phase classifier: Among the good photos, separate dividing cells (mitosis) from resting cells (interphasis)

1.3 Evaluation Metrics

For evaluation of the model's performance, sets of images classified in consensus by a pool of biologists will be compared to the results from the model, and the performance will be given by the F-score (the harmonic mean between precision and recall) for the mitosis detection. $F_{score} = 2\frac{precision \times recall}{precision + recall}$ This performance is well suited for our purposes, given that we both want a high detection rate for mitosis and do not want false positives.

The precision is calculated as the fraction of true positives (TP) over the sum of all classified as positive, both true and false(FP), $precision = \frac{TP}{TP+FP}$. In a sense, it is a measure of how confident we can be of a positive classification: if precision is 1, we can be trust that examples classified as positive are in fact positive.

The recall is calculated as the fraction of positive examples that have been correctly labeled as positive, $recall = \frac{TP}{TP+FN}$. If both precision and recall are equal to one (and thus the F-score is also 1), the model is flawless.

2 Analysis

2.1 Data Exploration and visualization

- The dataset consists in 2673 photos of single BY-2 cells, labeled by their mitotic stage (interphase or mitosis) or labeled as Unknown, Unclassifiable, Not a cell
- Original photos have been taken at GaTE lab at University of São Paulo, using DAPI stain, that colors genetic material in fluorescent blue, and have been cropped automatically around cells.
- The dataset photos are 100x100 pixels, RGB.
- The stages are not represented in the same proportion, there are:
  - 52 unclassifiable photos
  - 50 'not a cell'
  - 1358 considered interfase:
  - 456 labeled mitosis
  - 753 'Unknown' photos
  - 4 abnormal-labeled photos
- The labels have been provided by a pool of biologists
  - Interphase or Mitoses are labels that say something about the cell phase.
  - Unknown is a label given when answer is very unsure.

- Unclassifiable labeled photos have cells broken apart in some part of the process, ”dead cells”.
- Not a cell means that the photo has been wrongly cropped, such that there is nothing to classify.

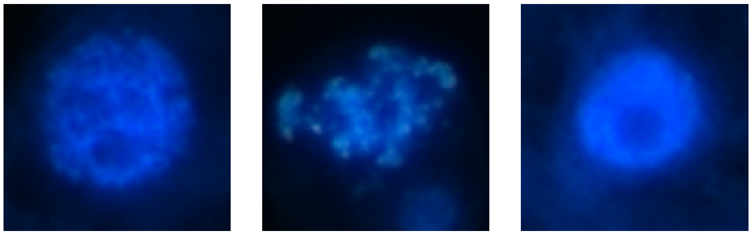

Fig 2.1: Two cells during mitosis in the left, and an interphasis cell on the right.

2.2 Algorithms and Techniques

Many traditional algorithms of supervised classification will be compared, given as inputs two features proven to be successful in texture analysis, i.e Histogram of Gradients (HoG) and Local Binary Patterns (LBP)[2], and for each classification step the best one will be chosen.

The classifier algorithms that are tested in greater detail are the following:

- Decision Tree
- Random Forest
- Support Vector Machine(Radial basis function)
- Logistic Regression

Decision Tree classifiers work by growing a sequence of decisions(questions) about features on the dataset, and separating the samples according to the response (e.g. is feature 3 bigger than .3?). Each of these binary decisions are measured according to their goodness in separating the different classes, which is measured by the information gain(i.e the difference in entropy before and after the separation), or by the gini impurity (i.e probability of misclassification for random samples of a node using random labels according to proportion). The maximum number of decisions is the maximum depth of a tree, and there may be overfit in overgrown trees.

Random Forest is an ensemble methor for classification, known for its resistance to overfitting. In this method, multiple decision trees are trained in different subsets of the training data, and a majority vote is taken to classify a new sample.

Support Vector Machines are a class of machine learning algorithms in which the separation boundaries between classes are calculated to maximize the distance(called ”margin”) between itself and the closest points(”support vectors”). This maximization is managed by the Cost of proximal points, such that higher cost parameters set the margin 'harder', using less support vectors but increasing the influence of each, trading stability for error penalty. The kernel defines the measuring of distances, and in our case the radial basis function kernel switches to gaussian distances instead of euclidean, with a gaussian width inverse to the γ parameter.

Logistic regression classifiers fit a logistic curve to the probability of the parameters' sample being of a given class, and the classification is taken by thresholding the probability distribution in the 50% chance. The best fit is calculated via gradient descent, and the error is regularized by a cost parameter that holds parameters from getting too big.

The classifiers' performance are measurable by comparing predicted to real labels, via the evaluation metrics explained below.

2.3 Benchmark Model

In this domain, the process of classification is usually by manual classification of cells via visual inspection and comparisson to standard images[1], and lacks a stabilished automated model.

Three benchmarks will be compared to our results:

- Uniform random guessing
- Biased random guessing (accounting for unbalanced dataset)
- Support vector machine with simpler features (more on methodology)

These benchmarks were chosen to enable us to account for how much our model improves the results when compared to naive classifications (from guessing benchmarks) and to test how much our features improve results in relation to the simpler ones of the third benchmark.

Both classification tasks (photo and phase) will have their own benchmark generated as above, each with their own parameters.

3 Methodology

3.1 Classification Benchmark

The benchmark model will be a Support Vector Machine classifier acting upon simple features, i.e the fourier transform of the original photo (of the blue channel, which is the only important because of the stain). This benchmark will be generated by grid searching an Radial Basis Function kernel for C values [0.01,0.1,1,10,100] and γ values [0.25,0.5,0.75,...,4] via 5-fold cross validation.

3.2 Preprocessing

- Discard the abnormal-labeled and wrong sized photos

- Discard cells classified as "Unknown".
- Get only the blue part of RGB, since this loses very little information in this case and well reduces the dimensionality.
- Get the frequency domain representation of the images (fourier transform), here considered the 'simple features'
  - For the 100x100 images, total of 20.000 features were generated
  - 10.000(100x100) features of power
  - 10.000(100x100) features of phase
- Get the Histogram of Gradients and Local Binary Pattern of cells.
  - HoG parameters are orientations=8, pixelsPerCell = (8,8) , cellsPerBlock = (3,3), returning 7200 features
  - The LBP parameters P and R range from 4 to 20 and 2 to 10, respectively, in steps of 2. Each result is resumed in a 50-bin histogram, for a total of $9 \times 5 \times 50 = 2250$ features.
- Apply PCA on the data, separately on HoG and LBP, to reduce the dimension space.
- Split data into train and test in a stratified k-fold manner for validation.
- Encode labels

3.3 Implementation

1. Separate photos into good photos (interphase, mitosis) and bad photos (Unclassifiable, Not a cell) for the photo classifier.
2. Train in a 5-fold manner the following eight classifiers(with default parameters):
   - Random Forest
   - k-Nearest Neighbors
   - Support vector machine (Radial basis function)
   - Linear Support vector machine
   - Decision Tree
   - Logistic Regression
   - Gaussian Process Classifier
   - Naive Bayes
3. Grid search the two best classifiers for the best parameters, and choose the best of them (more on the grid search below)
4. Separate good photos into mitosis and interphasis.
5. Repeat steps 2 and 3

3.3.1 Grid searching

All grid searchs were made via 5-fold cross validation.

The Random Forest classifier search for parameters ranged across three parameters, the minimum of samples to split a given branch, which took values [2, 10, 20, 40, 80, 160], the criterion used for selecting the best split, between Gini Impurity and Information Gain, and the maximum allowed depth, that took values [2,3,4,5,6].

The grid search for parameters on the SVM classifier (RBS) had seven possible values for the Cost parameter .01, .1, 1, 2, 4, 8, 16, and twenty values for the inverse width γ parameter, from .1 to 2 times the default value (which is 1/number of features), in incremental steps of 0.1.

The Logistic Regression classifier had the same 7 values of C (regularization cost) referenced above and two weighting of classes, same weigh, and balanced weights (which makes the product $ClassWeight \times NumberOfSamples$ equals the same for both classes).

The Decision tree classifier grid had the two criterions Gini Impurity and Information Gain, the minimum of samples to split a given branch(same as random forest) and class weighting.

3.4 Refinement

For each classifying task one classifier was refined in this section: the classifier with best results after the above grid searching.

For each one, three steps were applied:

1. Finer grid search close to the best parameters of the first search
2. Use features without PCA
3. Add fourier transform features (without PCA)
4. Another grid search with proximal parameters

3.4.1 Finer grid details

For the photo classifier(Radial Basis Function SVM), grid parameters were C = [.8, .9, 1, 1.1, 1.2] and γ = [.5, 1, 1.5, ..., 6.5, 7]$\times 10^{-5}$.

For the phase classifier(Logistic Regression), grid parameters were C = [0.06,0.08,0.1,0.12,0.16,0.2,0.5]

Fourier transforms were separated into phase and amplitude, summing 20000 features, adding up to 29450 features total.

4 Results

4.1 Preprocessing

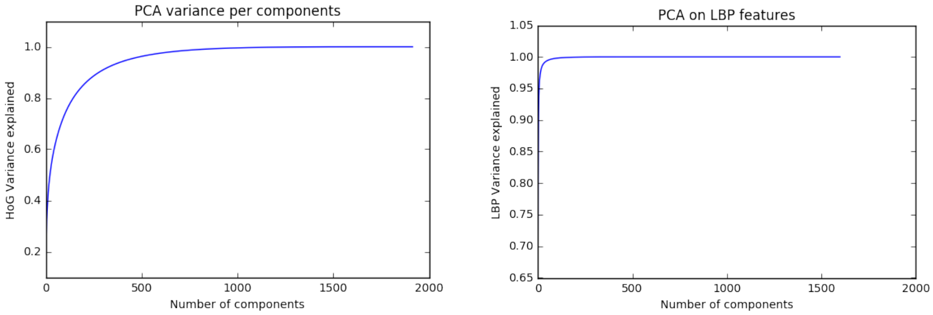

Fig 4.1: Explained variance reaches 99% with 1500 components, and subsequent ones have each less than 0.01% variance explained.

4.2 Model Evaluation and Validation

4.2.1 Comparisson of classifiers

The results of the photo classifier are resumed in the table below, for the 5-fold cross validation:

Photo Classifier	Mean score	σ
k-Nearest Neighbors	.973	.001
Support vector machine (Radial basis function)	.981	.003
Linear Support vector machine	.552	.161
Gaussian Process Classifier	.478	.178
Naive Bayes	.785	.113
Decision Tree	.962	.008
Random Forest	.981	.003
Logistic Regression	.545	.169
Benchmark random guessing	.644	0.009
Benchmark biased guessing	.334	0.008
Benchmark simple features	.981	.003

The best classifiers for the photo were the Support Vector Machine with Radial Basis Function kernel, and the Random Forest.

The results for the phase classifier are showed in tha table below.

Phase Classifier	Mean score	σ
k-Nearest Neighbors	.474	.096
Support vector machine (Radial basis function)	.436	.096
Linear Support vector machine	.549	.065
Gaussian Process Classifier	.436	.096
Naive Bayes	.464	.108
Decision Tree	.561	.075
Random Forest	.425	.065
Logistic Regression	.643	.080
Benchmark random guessing	.331	.010
Benchmark biased guessing	.223	.009
Benchmark simple features	.436	.096

The best classifiers for the cell phase were the Decision Tree and the logistic regression.

4.2.2 Grid searching

After grid searching the best two classifiers, the results were as follows

Photo Classifier	Mean score	σ	Parameters
Support vector machine (RBF)	.982	.003	C=1, $\gamma = 6.66 \times 10^{-5}$
Random Forest	.980	.003	MaxDepth=6, MinSamples=2
Phase Classifier	Mean score	σ	Parameters
Decision Tree	.564	.038	Entropy ,MinSamples=2, balanced
Logistic Regression	.645	.073	C=.16, balanced

4.2.3 Refinement

Both classifiers results are summarized in the following table

Photo Classifier	Mean score	σ	
Finer grid	.982	.003	
No PCA	.981	.003	
No PCA + fourier	.981	.003	
No PCA + fourier, Grid	.981	.003	
Phase Classifier	Mean score	σ	
Finer grid	.649	.074	
No PCA	.675	.054	
No PCA + fourier	.713	.023	
No PCA + fourier, Grid	.713	.021	

After refinement, the best result for the photo classifier was no better than before refinement (.982). For the phase classifier, the best results (.713) were found by maintaining all features (not applying PCA) and using both texture features and fourier features, and were a substantial increase (.713 - .499 = .214) over the previous results.

4.3 Justification

4.3.1 The benchmarks

Three benchmarks were proposed in this project:

- Uniform random guessing
- Biased guessing(probability equal to relative frequency)
- Simple features classifier (Frequency representation via fourier transform)

Among them, a hierarchy of best results were clearly observable: the simple features always did better than random guessing, and random guessing was always better than biased guessing. The biased benchmark represents a model that favors the most frequent class, but has worst results in the infrequent one. The prevalence of uniform over biased random is an indication of the adequateness of the chosen scoring function, considering that we wanted a score that did not favored one class over another, despite the imbalance in the number of samples from each. As the simple features has the best results among benchmarks, it was the most compared to subsequent results, for it is clear that improvement over the simple features benchmark means superior results to the random guessing as well.

4.3.2 Photo Classifier

The photo classifier final results were the almost the same same as the simple features benchmark (.982 vs .981), suggesting that the new features do not really improve performance in this task (the .001 increase was considered negligible, as it is less than a third of the standard deviation). This is not really an obstacle, given that this benchmark has an already high performance, being enough to suffice the needs of our problem, and showing a much bigger score than the .644 of random guessing. The final score suggests an error of about two photos in a hundred, and given that bad photos are not very common (4% of the dataset, 102/2673), this errors don't stain our final results. The problem of classifying good photos is thus well solved by the benchmark.

4.3.3 Phase Classifier

The phase classifier final results were much better than the simple feature benchmark (.713 vs .436), and even better in comparison to the random guessing(.331). The texture features, along with the original fourier transform, were capable of reaching results that were not possible with each set of features alone.

Our model was tested for robustness by fitting and scoring with subsets of 2000 features(for all samples) changed to zero, and as the mean score never fell below .66, we can say that is robust to problems in features' subsets. The score also did not significantly changed for small changes in the cost parameter, indicating robustness also in the tuning.

The model have a small standard deviation in its scores(.021), and so we can trust its results to have an acceptable rate of correctness(above 70%), but we do not trust its individual classifications. The results are unfortunatelly not enough for the task in hand, as they do not match our expectations to subtitute humans in this task. Results above .8 of score would match our expectations.

5 Conclusion

5.1 Free-Form Visualization

The graphics below show the dispersion of classes among features selected from their weighing in the final classifier function. We can see that mitosis cells have a higher variance on their features, but there is much superposition. In agreement, errors seems to be more dense where the superposition is higher, as can be seen on the closer views.

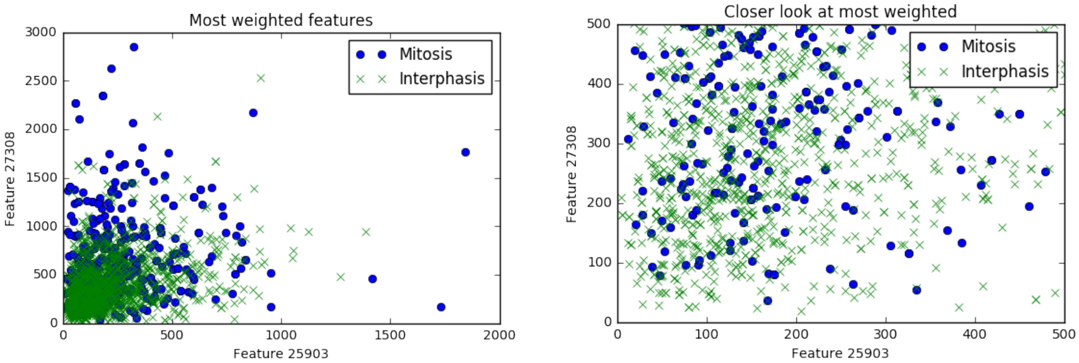

Fig 5.1: True labels dispersed over a two-features plane. Left: Whole view, Right: Zoom in on most dense part.

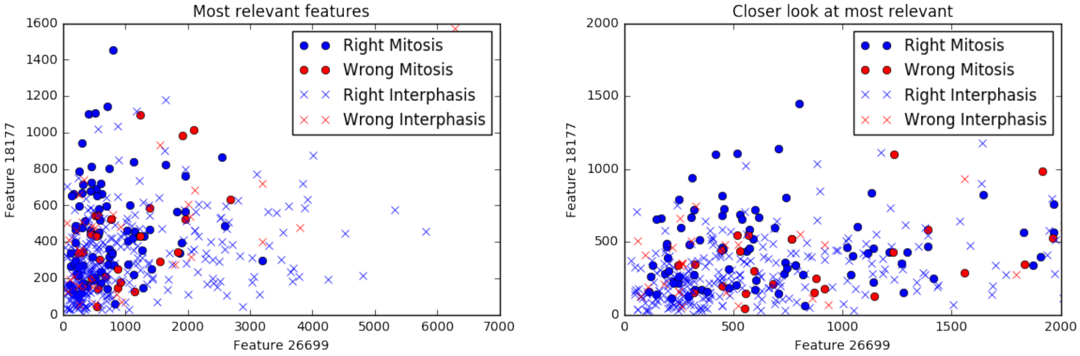

Fig 5.2: Predicted versus correct labels. The form (circle or X) indicates the true label, and the color (blue or red) indicates correct versus incorrect results. Left: Whole view, Right: Zoom in on most dense part.

5.2 Reflection

The process done can be summarized as follows:

1. A problem was posed
2. A dataset was acquired for the solution
3. The dataset was preprocessed
   (a) Errors and irrelevant information were discarded
   (b) New features were generated from the data
   (c) Dimensionality reduction techniques were applied
4. Benchmarks were created and applied to data
5. Classifiers were compared and selected
6. Classifiers were refined for the tasks
7. The final results were compared to the benchmarks

The most difficult part was part 6. The refinement process is not as straightfoward as the other ones, which leaves much space for different approaches, which complicates the process as it is not clear what is the right one. This freedom was also compelling, which is why part 6 was also one of the most interesting. The other most interesting part was 3b. As the application of new features made necessary some research, part 3b included a curious peek at the gargantuous field of computer vision, and at the huge amount of algorithms created and theorized to facilitate tasks like the present one.

5.3 Improvement

For the phase classifier, there are improvements that could be made upon the used algorithms, as the input of more processed features or the application of some kind of semi supervised learning for using the "Unknown" photos of the dataset. Also, more samples may be added to the dataset whenever possible.

The state of the art nowadays for image and pattern recognition resides in deep learning techniques, which I would consider using if I knew how.

The Photo classifiers trained didn't got better results than the simple features benchmars, which leaves the question if a better solution exists. As we encountered a very high f-score of .98 consistent between classifiers, we believe that errors may be due to outliers, and would not be improved by other models.

References

[1] Kumagai-Sano F, Hayashi T, Toshio Sano T, Hasezawa S, Cell cycle synchronization of tobacco BY-2 cells, Nature Protocols 2006, Published online 11 January 200[1].

[2] Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Huang, T. (2011, June). Large-scale image classification: fast feature extraction and svm training. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 1689-1696). IEEE.