A NOVEL APPROXIMATE INFERENCE APPROACH TO AUTOMATED CLASSIFICATION OF PROTEIN SUBCELLULAR LOCATION PATTERNS IN MULTI-CELL IMAGES

Shann-Ching Chen[1], Geoffrey J. Gordon[3], and Robert F. Murphy[1,2,3]

Department of Biomedical Engineering[1], Department of Biological Sciences[2], and Center for Automated Learning and Discovery[3], Carnegie Mellon University, Pittsburgh, PA 15213, USA

ABSTRACT

The subcellular location of proteins is most often determined by visual interpretation of fluorescence microscope images. In recent years, automated systems have been developed so that the protein pattern in a single cell can be objectively and reproducibly assigned to a location category. While these systems perform very well at recognizing all major subcellular structures, some similar patterns are not perfectly distinguished. Our goal here was to improve performance by considering more than one cell in a field. We describe how to construct a graphical model representation for a field of cells while taking into account the characteristics of the cell type being studied. We show that this approach provides improved performance on synthetic multi-cell images in which the true class of each cell is known, and that a new approximate inference method can provide this improved performance with significantly faster computation times than previous approaches.

1. INTRODUCTION

As the prospect of building meaningful models of biological systems grows through the acquisition of comprehensive information on protein *sequence, structure* and *activity*, it becomes increasingly important to have approaches that can provide information on the *subcellular location* of each protein as well. Such information is usually obtained using fluorescence microscopy to examine the distribution of fluorescently-tagged proteins. In recent years we have developed automated systems that can interpret such images with accuracy and reproducibility greater than visual examination [1].

These systems consist of machine classifiers and sets of informative numerical features (which we term SLFs, for Subcellular Location Features [1]) to describe protein distributions in the cell. Using large collections of HeLa cell images containing ten distinct subcellular patterns, the systems have achieved classification accuracies as high as 92% and 98% for 2D and 3D single cell images, respectively [1, 2]. The patterns of dissimilar classes can be distinguished quite well; however, there is still room to improve the classification accuracy for similar classes (such as endosomal and lysosomal proteins and different Golgi proteins).

We describe here an approach to improve this performance by constructing a graphical model to capture pattern information for more than one cell in a field. Graphical models have been extensively applied to problems in computer vision but have not previously been applied to the recognition of subcellular patterns in multi-cell images. Large numbers of such images [3] are increasingly being acquired both in projects aimed at determining the subcellular location of all proteins [4-7] and in drug screening by high-throughput microscopy [8].

A graphical model consists of an algorithm for constructing the graph itself and an algorithm for making inferences given the graph. In this paper, we first describe how to construct graphs for the problem of subcellular location classification. We next present a new inference algorithm, which we term prior updating, that permits inferences to be made for the resulting graphs. We then describe experimental results comparing performance and execution times for different inference methods.

2. CONSTRUCTING A GRAPHICAL MODEL

Given a field of cells in which each cell has been classified based solely on its SLFs, we consider how we can "revise" these assignments based on the number of classes most likely to be present in that field. To this end, we construct a graphical model with a node for each cell and edges connecting similar cells. There are two sorts of information we might consider when deciding which pairs of cells to connect with edges: similarity between cells in feature space, and closeness between cells in physical space. The relative importance of these two sources of information depends on how long the cells have been plated (t_{plate}) relative to their generation time (t_g), as well as how quickly the cells move (v_{trans}).

2.1. Physical and Feature Space Models

If the plating time is significantly greater than the generation time ($t_{plate} >> t_g$), each original cell is expected to divide a number of times before imaging. If in addition

v_{trans} is low, we may consider it likely that the classes of cells adjacent to one another are the same. In this case we can construct a graph by connecting two cells if the Euclidean distance between the centers of the cells in the field is low; we call this graph a *physical model*.

If t_{plate} is short relative to t_g, most cells will not have time to divide prior to being imaged. Even if t_{plate} is long, if v_{trans} is also large, related cells are likely to move away from one another after they divide. In either case, physical proximity of cells does not provide much information about their likely similarity. The only clues that we have about the number of classes present and the number of cells in each class are the similarities between cells in the SLF feature space. In this case, we construct a graph by connecting pairs of cells whose z-scored Euclidean distance in feature space is small; we call this graph a *feature space model*.

In either type of model, we need to pick a cutoff distance d_{cutoff} to determine which pairs of cells are close enough to be connected by edges. The units of d_{cutoff} are different for the two types of models, but large values result in graphs with many edges, while small values result in graphs with few edges. We define a parameter C, the graph complexity, and select d_{cutoff} so that C% of the edges are present in the graph.

3. INFERENCE METHODS

Given a graph of either of these types, we can turn it into a graphical model by providing an algorithm which trades off evidence from a single-cell classifier against the desire to make a cell's classification similar to the classifications of its neighbors. We can derive several such algorithms by interpreting the graph as a Bayes network, in which groups of neighboring nodes are encouraged to have similar classes through various types of potential functions.

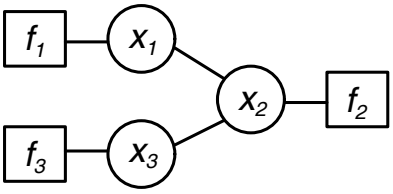

Figure 1. A collection of 3 test examples is arranged in a graph. f_i and x_i are the feature vector and the class label of example *i*, respectively. An edge connecting two nodes means that their values are directly related to one another.

3.1. Potts Potential and Voting Potential

Suppose that we have learned a classifier which maps the features *f* of an example to a probability distribution $P(x)$ over possible labels *x*. Now suppose that we have a collection f_1, f_2, f_3 of feature vectors for test examples with labels x_1, x_2, x_3 arranged in a graph, as shown in Figure 1. Informally, an edge connecting x_i and x_j means that labels x_i and x_j are likely to be the same.

If we interpret the graph as a Bayes net, each label x_i becomes a variable or node in the network. Whenever two nodes are connected by an edge we want to encourage their labels to be the same; one way to do so is by using the following pairwise potential function:

$$\varphi(x_1, x_2) = \begin{cases} \omega & x_1 = x_2 \\ 1 & otherwise \end{cases} \quad (1)$$

Here ω>1 is an arbitrary parameter which expresses how strongly we believe that x_1 and x_2 have the same label. The overall probability of a vector of labels *x* is:

$$P(x) = \frac{1}{Z} \prod_{nodes\ i} P(x_i) \prod_{edges\ i,j} \varphi(x_i, x_j) \quad (2)$$

where *Z* is the normalizing constant and $P(x_i)$ represents the evidence of node *i* for every possible label. The above potential is called the Potts potential, and the Bayes net with this potential is called the Potts model [9].

Unfortunately the Potts model does not perfectly capture our initial intuition about inference from labels of neighboring classes. To better capture this intuition, we define the voting potential function:

$$\varphi(x, v_1, v_2, ...) = \frac{\lambda / n + \sum_{k \in N(x)} I(v_k, x)}{m + \lambda} \quad (3)$$

where *x* is an arbitrary node in the graph, v_1, v_2, …v_m are *x*'s neighbors, *n* is the number of classes and *m* is the number of neighbors of *x* (not counting *x* itself). λ is a smoothing parameter: the smaller λ is, the more strongly *x*'s neighbors will influence *x*'s classification. *I* is an indicator function which is 1 when $v_k = x$ and 0 otherwise. $N(x)$ is the set of *x*'s neighbors. This voting potential function combines the evidence from all of node *x*'s neighbors into a summary vote which then influences *x*'s classification.

3.2. Inference methods on Bayes Nets

Given a Bayes net, we want to combine the evidence from the single-cell classifiers with the potential functions to infer the posterior probability of each class for each node. For some networks, an exact solution can be found using the belief propagation (BP) algorithm [10]. However, BP can only calculate the posterior probability correctly on graphs where there is at most one path between any two nodes. If there are loops in the graph, exact inference can be done in two ways: either by making a table of the joint probability distribution of all possible label vectors *x* and summing the appropriate entries, or by using the junction tree algorithm [11] to convert the loopy graph into a tree and then applying BP to the tree. We will write EIPP for exact inference with the Potts potential, and EIVP for exact inference on the

voting potential. In either case the computation time can be exponential in the length of the label vector x.

Exact inference is impractical for large graphs, and hence approximate methods are needed. Loopy belief propagation (LBP) iteratively applies belief propagation updates on a graph with loops and often gives good approximate inference when it converges [12]. We will write LBPP for LBP with the Potts potential, and LBVP for LBP with the voting potential. LBVP can still be too slow on large graphs, since its running time is exponential in the number of arguments to the largest potential function.

We have developed a fast approximation to LBVP which we term prior updating (PU), which accelerates computation of some LBP messages and ignores others. A complete description will be presented elsewhere (Chen, Gordon and Murphy, submitted). The data and code for the experiments below are available at http://murphylab.web.cmu.edu/software.

4. EXPERIMENTAL METHODS

4.1. 2D HeLa Image Set

We applied our methods on a set of fluorescence microscope images of HeLa cells created by introducing antibodies and molecular probes against proteins in major subcellular organelles [6]. The data set contains 862 single-cell images from ten classes, with each class having between 73 and 98 images. The true class of each image is known with certainty since the probe added to each slide is known.

4.2. Single-Cell Classifier

Feature set SLF16 [13], which has yielded the best single-cell classification results to date, was used to describe each single cell image. SLF16 contains 47 features of various types, including Zernike moment features, Haralick texture features, morphological features, and wavelet features. Descriptions of these features are available at http://murphylab.web.cmu.edu/services/SLF.

Given SLF16 features for each cell in our training set, we learned one Support Vector Machine (SVM) [14] for each class. The i^{th} SVM is trained to distinguish the i^{th} class from the union of all other classes. Each SVM uses an exponential radial basis function kernel with σ=7 and C=20, which were the optimal values for feature set SLF16 in our prior work [13]. To classify a test example, we fed it into each SVM, and the one with the highest output was assigned as the predicted class. To obtain posterior probabilities that are directly comparable between classes, after training each SVM we fit a sigmoid to its output scores using regularized maximum likelihood [15]. These posterior probabilities form the evidence at each node in our Bayes net.

5. RESULTS

We first evaluated whether the voting potential is better than the Potts potential at representing the information "neighboring nodes should have similar classes," using graphs that are small enough so that exact inference is achievable. We compared the performance of EIVP, PU, EIPP and LBPP to our SVM baseline classifier (using 16-fold cross-validation). The arrangement of the cells was synthetic, but the cell images were taken from the 2D HeLa dataset.

	EIPP	LBPP	EIVP	PU
Accuracy Improvement	1.58	1.58	2.84	3.04

Table 1. Results for graphical models of cell images in small graphs, in percentage points. Base accuracy of the single cell image classifier is 88.29%.

We built graphs containing 8 cells (4 each of two of the five possible classes), and fixed d_{cutoff} so that 50% of the possible edges were present. We used ω=1.3 for the Potts potential, and λ=2.5 for the Voting potential; these values were approximately optimal for a range of different Bayes nets. Table 1 shows the accuracy improvement in percentage points over the single cell classifier. The results suggest that the voting potential performs much better than the base classifier (one-tailed t-test: p=0.0066) and somewhat better than the Potts model (one-tailed t-test: p=0.0314). There is no significant difference between the exact and approximate inference methods for either model.

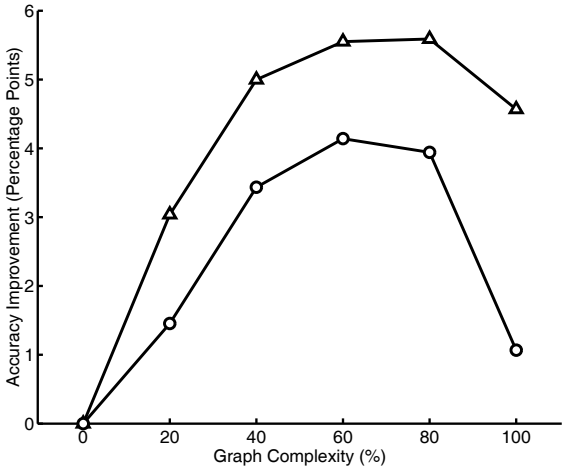

Figure 2. Results for graphical models of cell images in large graphs. The methods are PU (△) and LBPP (○).

Encouraged by the above results, we next considered if we can obtain similar improvement in large graphs using the full HeLa data set. The Bayes nets in this case are too large for the exact inference methods, so we only compared the performance of PU and LBPP to each other and to our SVM baseline classifier (using 12-fold cross-validation). We built a graph containing 12 cells (6 each of two of the 10 possible classes). In each trial, we varied d_{cutoff} to achieve levels of

connectivity ranging from 0% to 100% of the possible edges. Figure 2 demonstrates that PU can achieve good improvement in classification accuracy. LBPP can only improve the accuracy to a lesser degree. All differences in the graph are significant with $p<0.01$ except LBPP versus SVM at C=100.

Figure 3 compares the computational efficiency of the different inference methods and demonstrates that PU is much faster than competing algorithms. Each point in the figure shows the average inference time per trial on different sizes of graphs with various algorithms (note the logarithmic time scale). The exact inference methods take time exponential in the size of the graph and are impractical to run for graphs of more than 12 nodes, while the processing times of PU and LBPP are approximately linear in the size of the graph.

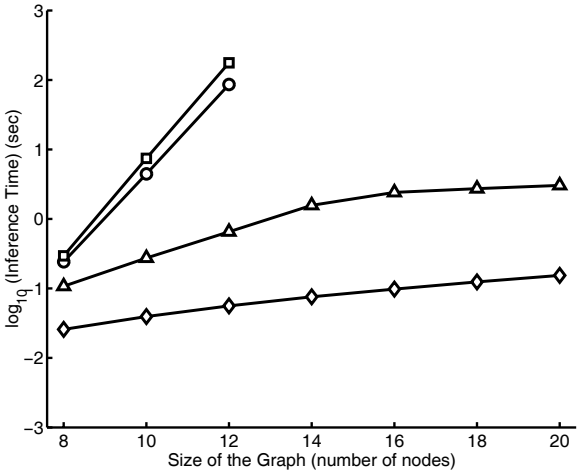

Figure 3. Inference Time vs. Graph Size with different inference methods. The methods are PU (◇), LBPP (△), EIVP (○), and EIPP (□). This experiment was conducted by 8-fold cross-validation for four similar classes (the endosomal and lysosomal proteins and two Golgi proteins).

6. CONCLUSION AND FUTURE WORK

We have presented a new solution to the problem of classifying multiple dependent examples in a protein subcellular location pattern recognition task. Our solution is based on a Bayes net with the voting potential function. In addition to the new Bayes net, we have presented a new inference algorithm called Prior Updating, an approximation to loopy belief propagation (which is itself an approximation to exact inference). Our experiments show that voting potential does better than Potts potential, and PU runs quickly and provides an accuracy improvement over the base classifier on large networks derived from real data.

Our work has particular implications for classification of patterns in images obtained by high-throughput or automated microscopy [7, 8]. Since high-throughput systems typically use low magnification, the number of cells per field is often high and the accuracy of single-cell classifiers is usually not perfect. By applying this method on multi-cell images made of real single cells and synthesized locations, we are able to verify that our scheme can be used for such systems to achieve significantly better performance.

7. ACKNOWLEDGMENTS

This work was supported in part by NSF grant EF-0331657.

8. REFERENCES

[1] K. Huang and R. F. Murphy, "From quantitative microscopy to automated image understanding," *J Biomed Optics*, vol. 9, pp. 893-912, 2004.

[2] X. Chen and R. F. Murphy, "Robust Classification of Subcellular Location Patterns in High Resolution 3D Fluorescence Microscopy Images," *Proc 26th Intl Conf IEEE Eng Med Biol Soc*, pp. 1632-1635, 2004.

[3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Proc. 2004 IEEE Conf on Computer Vision Pattern Recognition*, vol. 1, pp. 261-268, 2004.

[4] A. Kumar, S. Agarwal, J. A. Heyman, S. Matson, M. Heidtman, S. Piccirillo, L. Umansky, A. Drawid, R. Jansen, Y. Liu, K.-H. Cheung, P. Miller, M. Gerstein, G. S. Roeder, and M. Snyder, "Subcellular localization of the yeast proteome," *Genes Develop.*, vol. 16, pp. 707-719, 2002.

[5] X. Chen, M. Velliste, S. Weinstein, J. W. Jarvik, and R. F. Murphy, "Location proteomics - Building subcellular location trees from high resolution 3D fluorescence microscope images of randomly-tagged proteins," *Proc SPIE*, vol. 4962, pp. 298-306, 2003.

[6] S. Ghaemmaghami, W. K. Huh, K. Bower, R. W. Howson, A. Belle, N. Dephoure, E. K. O'Shea, and J. S. Weissman, "Global analysis of protein expression in yeast," *Nature*, vol. 425, pp. 737-41, 2003.

[7] C. Conrad, H. Erfle, P. Warnat, N. Daigle, T. Lorch, J. Ellenberg, R. Pepperkok, and R. Eils, "Automatic Identification of Subcellular Phenotypes on Human Cell Arrays," *Genome Research*, vol. 14, pp. 1130-1136, 2004.

[8] Z. E. Perlman, M. D. Slack, Y. Feng, T. J. Mitchison, L. F. Wu, and S. J. Altschuler, "Multidimensional Drug Profiling by Automated Microscopy," *Science*, vol. 306, pp. 1194-1198, 2004.

[9] R. Potts, "Some Generalized Order-Disorder Transformation," *Proc. Cambridge Philosophical Soc.*, vol. 48, pp. 106-109, 1952.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*: Morgan Kaufmann, 1988.

[11] C. Huang and A. Darwiche, "Inference in Belief Networks: a Procedural Guide," *Intl. J. Approximate Reasoning*, vol. 15, pp. 225-263, 1996.

[12] K. Murphy, Y. Weiss, and M. Jordan, "Loopy Belief Propagation for Approximate Inference - an Empirical Study," *Uncertainty in Artificial Intelligence*, pp. 467-475, 1999.

[13] K. Huang and R. F. Murphy, "Boosting Accuracy of Automated Classification of Fluorescence Microscope Images for Location Proteomics," *BMC Bioinformatics*, vol. 5, pp. 78, 2004.

[14] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 1-25, 1995.

[15] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers, MIT Press*, pp. 61-74, 1999.