2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)
April 4-7, 2018, Washington, D.C., USA

SEMI-SUPERVISED LEARNING FOR CELL TRACKING IN MICROSCOPY IMAGES

Nisha Ramesh[†⋆] *Tolga Tasdizen* [†⋆]

[†] Department of Electrical and Computer Engineering, University of Utah, United States
[⋆] Scientific Computing and Imaging Institute, University of Utah, United States

ABSTRACT

This paper discusses an algorithm for semi-supervised learning to predict cell division and motion in microscopy images. The cells for tracking are detected using extremal region selection and are depicted using a graphical representation. The supervised loss minimizes the error in predictions for the division and move classifiers. The unsupervised loss constrains the incoming links for every detection such that only one of the links is active. Similarly for the outgoing links, we enforce at-most two links to be active. The supervised and unsupervised losses are embedded in a Bayesian framework for probabilistic learning. The classifier predictions are used to model flow variables for every edge in the graph. The cell lineages are solved by formulating it as an energy minimization problem with constraints using integer linear programming. The unsupervised loss adds a significant improvement in the prediction of the division classifier.

Index Terms— semi-supervised, cell division, tracking, microscopy, integer linear programming

1. INTRODUCTION

Tracking cells in time and studying their evolution is necessary to analyze cell growth, mobility and lineages. The difficulty in cell tracking is due to low signal to noise ratio, low contrast, high cell density and division of cells. Learning to predict different cell processes, such as cell divisions, transitions is an integral part to generating cell lineages. The process of annotating complete tracks in the training examples is expensive. Due to the limited annotated samples of division, it is challenging to predict cell division. These observations motivate us to study semi-supervised methods that use minimal training information.

There are two different perspectives on cell tracking. The first approach is tracking by model evolution. Mathematical models are used to represent cells, and they are propagated in time. This approach combines the segmentation and tracking problem, [1, 2, 3, 4]. These algorithms minimize an energy functional to evolve the mathematical representation of cells. Particle filters with various models for shape representations of cells have been used to track multiple objects [5, 3]. The main drawback of these methods is that all decisions are made with limited temporal information. It is also difficult to account for new appearances of cells.

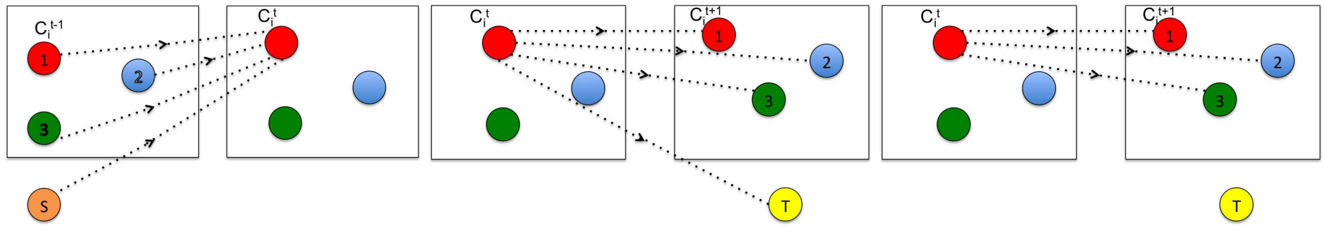
The second approach is tracking by detection. Here, the cells are detected, and then they are linked to form trajectories, [6, 7, 8]. A feasible solution is to aggregate strong detections which correspond to cells, which can be linked later to generate complete trajectories. By formulating the tracking problem as an integer program, priors can be learnt for the various cell processes. In [7], a Viterbi algorithm is used to add trajectories in a sequential manner. A joint segmentation and tracking approach, using spatially overlapping hypotheses generated using an ellipse fitting algorithm is proposed in [9]. The cell processes are modeled using flow variables. The optimal solution is obtained using integer programming with constraints. We use this inference method using network flows to determine the cell lineages. The novelty in our method is that the cell processes are learned in a semi-supervised framework. In [10] deep learning is used to generate the cell proposals. They link proposals in adjacent frames in a graphical model using edges representing different cellular events. The cell detection and tracking is then solved as the selection of a subset of cell and edge proposals.

In our method, tracking is performed on the detections generated using our semi-supervised framework for cell detection, where the unsupervised loss imposes constraints on the hierarchical region representation to choose at-most one cell in every path, [11]. Cells from adjacent time steps are linked to build a graphical representation. In this paper, we introduce an unsupervised loss to support the learning of classifiers to predict cell divisions and transitions. The unsupervised loss for the tracking problem imposes constraints on the incoming and outgoing links for all detected cells. Only one of the incoming links can be active and in the case of outgoing links, at-most two links can be active. The goal of our algorithm is to achieve better results with significantly smaller amount of training data in a "semi-supervised" framework. Using the predictions from the classifiers trained the cell lineages are inferred as described in [9] .

2. METHODS

We introduce a semi-supervised framework for training the transition and division classifier, to maximize the accuracy with limited training data. Individual detections are repre-

978-1-5386-3636-7/18/$31.00 ©2018 IEEE



(a) Incoming transition constraints (b) Outgoing transition constraints (c) Division constraints

Fig. 1: Path constraints for unsupervised loss

sented as C_i^t, where i is the index and t is the time frame and the set of all detections are represented as $\mathbf{C}$. Let $m : \mathcal{R}^{\mathcal{M}} \rightarrow \mathcal{B}$, $d : \mathcal{R}^{\mathcal{D}} \rightarrow \mathcal{B}$, be a binary classifier function used to identify true cell transitions and divisions respectively, where $\mathcal{M}$ is the dimensionality of the transition feature vector, $\mathcal{D}$ is the dimensionality of the division feature vector. The transition and division features are computed similar to the features extracted in [12]. We relax the binary classifier functions m and d to represent transition and division probability, using a differentiable classification function, such as a logistic sigmoid. A probabilistic model that has contributions from a supervised loss likelihood and an unsupervised loss likelihood is used for the learning. We train the transition classifier in a semi-supervised framework, where the score $m_{\{i\}\{j\}}^t$, predicts the transitions between node i at time t and j at time $t+1$. Similarly, we train the division classifier $d_{\{i\}\{k\}}^t$, to learn the likelihood of division of node i at time t into the k^{th} pair of child nodes at time $t+1$. Once, the classifiers are learnt, the tracking problem is formulated as a energy minimization problem with constraints as seen in [9]. The details of the algorithm is elucidated below.

2.1. Supervised Loss

Supervised learning tries to learn model parameters such that the error between the prediction and true estimate is minimized. The training set, $(\mathbf{X_m}, \mathbf{y_m})$, denotes the transition feature set matrix and annotations which indicate if it is a true transition. The training set, $(\mathbf{X_d}, \mathbf{y_d})$, denotes the division feature set matrix and annotations which indicate if it is a true division. $\bar{\mathbf{X}}_\mathbf{m}$, $\bar{\mathbf{X}}_\mathbf{d}$ has transition and division features respectively from both the training and test images. Let $\hat{\mathbf{y}}_\mathbf{m}$ be the transition probabilities estimated for the training features $\mathbf{X_m}$ using the classifier m, parametrized by $\mathbf{w_m}$. The supervised transition loss, can be represented as an i.i.d. Gaussian $\mathcal{N}(0, \sigma_{\mathbf{s_m}})$ which penalizes the prediction errors and tries to minimize errors from the true transitions.

$$P(\mathbf{y_m}|\mathbf{X_m}, \mathbf{w_m}, \sigma_{s_m}) = \frac{1}{(\sqrt{2\pi\sigma_{s_m}^2})^{N_m}} \exp\left(-\frac{\|\mathbf{y_m} - \hat{\mathbf{y}}_\mathbf{m}\|_2^2}{2\sigma_{s_m}^2}\right) \quad (1)$$

where N_m is the number of training samples for transitions. Let $\hat{\mathbf{y}}_\mathbf{d}$ be the division probabilities estimated for the training features $\mathbf{X_d}$ using the classifier d, parametrized by $\mathbf{w_d}$. The supervised division loss, can be represented as an i.i.d. Gaussian $\mathcal{N}(0, \sigma_{\mathbf{s_d}})$ which penalizes the prediction errors and tries to minimize errors from the true divisions.

$$P(\mathbf{y_d}|\mathbf{X_d}, \mathbf{w_d}, \sigma_{s_d}) = \frac{1}{(\sqrt{2\pi\sigma_{s_d}^2})^{N_d}} \exp\left(-\frac{\|\mathbf{y_d} - \hat{\mathbf{y}}_\mathbf{d}\|_2^2}{2\sigma_{s_d}^2}\right) \quad (2)$$

where N_d is the number of training samples for divisions.

2.2. Unsupervised Loss

We need to include the following constraints on the links between detected cells to accommodate different cell processes. (i) A cell may not have more than one ancestor. (ii) A cell may have a maximum of two descendants (transitions/cell divisions). The unsupervised loss also ensures, if no incoming links are selected, the appearance link, the link between the source and the node is selected. Similarly, if no outgoing links are selected, the disappearance link, the link between the target and the node is selected. Every cell is assigned a constant probability for appearance/disappearance in every frame.

Let n_i be the number of incoming links from cells at time $t-1$ in spatial proximity to the cell C_i^t. The constraint to enforce only a single ancestor/new appearance of a cell near boundaries can be formulated as

$$\begin{aligned} E_{in}(C_i^t) &= \bigvee_{j=0}^{n_i} \tilde{P}_j^t \bigwedge_{k=0}^{n_i-1} \tilde{m}_{\{jk\},\{i\}}^{t-1} \\ &= 1 - \prod_{j=0}^{n_i} \left(1 - \tilde{P}_j^t \prod_{k=0}^{n_i-1} \tilde{m}_{\{jk\},\{i\}}^{t-1}\right) \end{aligned} \quad (3)$$

$$\tilde{m}_{\{jk\},\{i\}}^{t-1} = \begin{cases} m_{\{k\},\{i\}}^{t-1} & \text{if } j = k \\ \neg\, m_{\{k\},\{i\}}^{t-1} & \text{otherwise} \end{cases} \qquad \tilde{P}_j^t = \begin{cases} P_a(C_i^t) & \text{if } j = n_i \\ 1 & \text{otherwise} \end{cases}$$

where we weight the term supporting the appearance of the cell with $P_a(C_i^t)$. This constraint implies that only one of the incoming links in Figure 1a can be true.

Similarly, let n_o be the number of outgoing links from the cell C_i^t to the cells in spatial proximity at time $t+1$. The constraint to enforce a single descendent (in case of transitions/disappearances) is formulated as in the previous case :

$$M_{out}(C_i^t) = \bigvee_{j=0}^{n_o} \tilde{P}_j^t \bigwedge_{k=0}^{n_o-1} \tilde{m}_{\{i\},\{jk\}}^t \tag{4}$$

$$\tilde{m}_{\{i\},\{jk\}}^t = \begin{cases} m_{\{i\},\{k\}}^t & \text{if } j = k \\ \neg\, m_{\{i\},\{k\}}^t & \text{otherwise} \end{cases} \qquad \tilde{P}_j^t = \begin{cases} P_d(C_i^t) & \text{if } j = n_o \\ 1 & \text{otherwise} \end{cases}$$

where we weight the term supporting the disappearance of the cell with $P_d(C_i^t)$. This constraint implies that only one of the outgoing links in Figure 1b can be true.

Lastly we need to include constraints to accommodate cell division of C_i^t to potential daughter pairs at time $t + 1$. Let $n_d = \frac{n_o(n_o-1)}{2}$ define the possible combination of pairs of daughter cells at time $t + 1$. The constraint enforces at-most one of the pairs to be selected and is formulated as

$$D_{out}(C_i^t) = \bigvee_{j=0}^{n_d-1} \bigwedge_{k=0}^{n_d-1} \tilde{d}_{\{i\},\{jk\}}^t \tag{5}$$

$$\tilde{d}_{\{i\},\{jk\}}^t = \begin{cases} d_{\{i\},\{k\}}^t & if\ j = k \\ \neg\, d_{\{i\},\{k\}}^t & otherwise \end{cases}$$

where, the division classifier, $d_{\{i\},\{k\}}^t$ rates triples of regions associated with the division. In Figure 1c the potential daughter pairs are ({1,2}, {2,3},{1,3}), the constraint ensures that at-most one of the three pairs is selected..

For any detection C_i^t, we need to account exclusively for a potential move or a division, hence, the term representing the outgoing transitions can be as written as follows:

$$E_{out}(C_i^t) = M_{out}(C_i^t) \neg D_{out}(C_i^t) + \neg M_{out}(C_i^t) D_{out}(C_i^t)$$

The combined term to account for incoming and outgoing transitions for C_i^t is given as :

$$\begin{aligned} F(C_i^t) &= E_{in}(C_i^t) \bigwedge E_{out}(C_i^t) \\ &= E_{in}(C_i^t) M_{out}(C_i^t) + E_{in}(C_i^t) D_{out}(C_i^t) \\ &\quad - 2E_{in}(C_i^t) \cdot D_{out}(C_i^t) \cdot M_{out}(C_i^t) \end{aligned} \tag{6}$$

To formulate the unsupervised loss, for every detection, $F(C_i^t)$ is computed as in Equation 6. Let $\tilde{\mathbf{F}}_{\mathbf{w_m},\mathbf{w_d}}(\mathbf{C})$, be a vector of $F(C_i^t)$, for all detections from all time frames. The unsupervised loss likelihood is an i.i.d. Gaussian $\mathcal{N}(0, \sigma_u)$ which penalizes the difference between each element of $\tilde{\mathbf{F}}_{\mathbf{w_m},\mathbf{w_d}}(\mathbf{C})$ and 1.

$$P(\mathbf{1}|\bar{\mathbf{X}}_\mathbf{m}, \bar{\mathbf{X}}_\mathbf{d}, \mathbf{w_d}, \mathbf{w_m}, \sigma_u) = \frac{1}{(\sqrt{2\pi\sigma_u^2})^{N_u}} \exp\left(-\frac{\left\|\mathbf{1} - \tilde{\mathbf{F}}_{\mathbf{w_m},\mathbf{w_d}}(\mathbf{C})\right\|_2^2}{2\sigma_u^2}\right) \tag{7}$$

where N_u is the number of detections in all time frames. This forces the predictions of the classifier to conform to the path constraints. The standard deviation parameters σ_{s_m}, σ_{s_d} and σ_u control the contributions of the training data and the unsupervised loss in the learning framework. Finally, we include a regularization term to prevent overfitting. This term constrains any abrupt change in the model parameters establishing the smoothness constraint for the solution.

$$P(\mathbf{w_m}) = \frac{1}{(\sqrt{2\pi})^M} \exp\left(-\frac{\|\mathbf{w_m}\|_2^2}{2}\right),\ P(\mathbf{w_d}) = \frac{1}{(\sqrt{2\pi})^D} \exp\left(-\frac{\|\mathbf{w_d}\|_2^2}{2}\right) \tag{8}$$

2.3. Bayesian formulation for parameter estimation

By applying Bayes' rule, we have the posterior distribution of $\mathbf{w_m}, \mathbf{w_d}$ as

$$\begin{aligned} P(\mathbf{w_m}, \mathbf{w_d} \mid \mathbf{X_m}, \mathbf{X_d}, \bar{\mathbf{X}}_\mathbf{m}, \bar{\mathbf{X}}_\mathbf{d}, \mathbf{y_m}, \mathbf{y_d}, \sigma_u, \sigma_{s_m}, \sigma_{s_d}) &\propto \\ P(\mathbf{w_m}) \cdot P(\mathbf{w_d}) \cdot P(\mathbf{1}|\bar{\mathbf{X}}_\mathbf{m}, \bar{\mathbf{X}}_\mathbf{d}, \mathbf{w_d}, \mathbf{w_m}, \sigma_u)& \\ \cdot P(\mathbf{y_m}|\mathbf{X_m}, \mathbf{w_m}, \sigma_{s_m}) \cdot P(\mathbf{y_d}|\mathbf{X_d}, \mathbf{w_d}, \sigma_{s_d})& \end{aligned} \tag{9}$$

The parameters $\mathbf{w_m}, \mathbf{w_d}, \sigma_{s_m}, \sigma_{s_d}, \sigma_u$ are learned using maximum a posteriori (MAP) estimation. The log-likelihood is given as

$$\begin{aligned} L = &\underbrace{N_u \log \sigma_u + \frac{1}{2\sigma_u^2} \left\|\mathbf{1} - \tilde{\mathbf{F}}_{\mathbf{w_m},\mathbf{w_d}}(\mathbf{C})\right\|_2^2}_{E_U} \\ &+ \underbrace{\frac{1}{2} \|\mathbf{w_m}\|_2^2}_{E_{C_m}} + \underbrace{\frac{1}{2\sigma_{s_m}^2} \|\mathbf{y_m} - \hat{\mathbf{y}}_\mathbf{m})\|_2^2 + N_m \log \sigma_{s_m}}_{E_{S_m}} \\ &+ \underbrace{\frac{1}{2} \|\mathbf{w_d}\|_2^2}_{E_{C_d}} + \underbrace{\frac{1}{2\sigma_{s_d}^2} \|\mathbf{y_d} - \hat{\mathbf{y}}_\mathbf{d})\|_2^2 + N_d \log \sigma_{s_d}}_{E_{S_d}} \end{aligned} \tag{10}$$

Minimizing the negative log-likelihood the parameters are estimated in a similar manner as seen in [11]. Our main contribution is learning the cell processes in a semi-supervised framework. To build the lineage tree for all cells we use an inference method from [9].

3. RESULTS

We use two datasets for testing our framework from the cell tracking challenge, [13]. The datasets used in the experiments are (i) Hela Dataset: It has two 92-frame sequences. (ii) SIM Dataset: It has six 50 to 100 frame sequences. There is a lot of variation in the appearance of the cells, some of them have low contrast, and some of them have a very noisy background. For the Hela dataset, we trained on one sequence and tested on the other sequence. Similarly, we trained on 5 sequences and tested on the 6th sequence in the SIM dataset. Random sampling was used when using only a fraction of the training data. We report the F-score, precision and recall scores for the migration and division classifiers along with a global tracking

Method (% of move/division training examples)	Migration			Division			
	Precision	Recall	F-score	Precision	Recall	F-score	TRA
Our method (20/20)	0.99	0.99	0.99 (0.99)	0.90	0.92	0.9098 (0.8041)	0.96
Our method (20/40)	0.99	0.99	0.99 (0.99)	0.89	0.94	0.9143 (0.8809)	0.96
Our method (20/60)	0.99	0.99	0.99 (0.99)	0.91	0.94	0.9247 (0.9039)	0.97
Our method (100/100) (transductive)	0.99	0.99	0.99 (0.99)	0.92	0.94	0.9298 (0.9296)	0.97
Türetken et al. [9] (100/100)	1.0	0.98	0.99	0.75	0.75	0.75	0.96

Table 1: Quantitative comparison for the synthetic dataset SIM-5 for varying percentages of training data. The F-scores for our method using only the supervised loss is given in parentheses.

Method (% of move/division training examples)	Migration			Division			
	Precision	Recall	F-score	Precision	Recall	F-score	TRA
Our method (20/20)	0.99	0.95	0.96 (0.96)	0.79	0.81	0.7998 (0.7331)	0.96
Our method (20/40)	0.99	0.96	0.97 (0.97)	0.81	0.82	0.8149 (0.7792)	0.97
Our method (20/60)	0.99	0.96	0.97 (0.97)	0.84	0.85	0.8449 (0.8211)	0.976
Our method ((100/100) (transductive)	0.99	0.96	0.97 (0.97)	0.86	0.87	0.8649 (0.8480)	0.98
Türetken et al. [9] (100/100)	0.99	0.96	0.97	0.83	0.86	0.84	0.97

Table 2: Quantitative comparison for Hela-2 dataset for varying percentages of training data. The F-scores for our method using only the supervised loss is given in parentheses.

metric (TRA) as defined in [13] for varying percentages of training data in Table 1 and 2. A cell migration event is considered to be successful if there is a true link between the cells at t and t+1. Similarly, we consider a division to be successful if it is reported at the right time frame and there are true links between the detected parent and daughter cells. We observe from Table 1 and 2 there there is a significant improvement in using the unsupervised loss when there is lower percentage of training data. Also, when we evaluate our framework in a "transductive" setting, by using all the examples from the training and testing dataset to compute the unsupervised loss, we see an improvement in the division classifier prediction in comparison to [9].

4. CONCLUSION

We proposed a semi-supervised framework for learning classifiers to predict cell division and transitions. Cell lineages are inferred using the predictions from our trained classifiers. Our semi-supervised learning can be incorporated as an additional block into existing algorithms for tracking cells where cell processes are learned. In the future, we would like to build a unified framework to detect, segment and track cells simultaneously.

5. REFERENCES

[1] O. Dzyubachyk et al., "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 852–867, March 2010.

[2] M. Maka et al., "Segmentation and shape tracking of whole fluorescent cells based on the chan-vese model," *IEEE Transactions on Medical Imaging*, vol. 32, no. 6, pp. 995–1006, June 2013.

[3] N. Ramesh and T. Tasdizen, "Cell tracking using particle filters with implicit convex shape model in 4D confocal microscopy images," in *ICIP*, Oct 2014, pp. 446–450.

[4] N. Ramesh, F. Mesadi, M. Cetin, and T. Tasdizen, "Disjunctive normal shape models," in *ISBI*, 2015, pp. 1535–1539.

[5] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, "A new hybrid bayesian-variational particle filter with application to mitotic cell tracking," in *ISBI*, 2011, pp. 1917–1920.

[6] M. Schiegg, et al., "Conservation tracking," in *ICCV*, Dec. 2013, pp. 2928–2935.

[7] K. E. G. Magnusson, J. Jaldn, P. M. Gilbert, and H. M. Blau, "Global linking of cell tracks using the viterbi algorithm," *IEEE Transactions on Medical Imaging*, vol. 34, no. 4, pp. 911–929, April 2015.

[8] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2054–2068, Oct 2016.

[9] E. Türetken, X. Wang, C. J. Becker, C. Haubold, and P. Fua, "Network flow integer programming to track elliptical cells in time-lapse sequences," *IEEE Transactions on Medical Imaging*, vol. 36, no. 4, pp. 942–951, April 2017.

[10] Saad Ullah Akram, Juho Kannala, Lauri Eklund, and Janne Heikkilä, "Cell tracking via proposal generation and selection," *CoRR*, vol. abs/1705.03386, 2017.

[11] N Ramesh, T. Liu, and T. Tasdizen, "Cell detection using extremal regions in a semi-supervised learning framework," *Journal of Healthcare Engineering*, vol. 2017: 4080874, June 2017.

[12] M. Schiegg, et al., "Graphical model for joint segmentation and tracking of multiple dividing cells," *Bioinformatics*, vol. 31, no. 6, pp. 948–956, 2015.

[13] M. Maka, et al., "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014.