Semi-Supervised Learning

Wei Pan

Division of Biostatistics and Health Data Science, School of Public Health,
University of Minnesota, Minneapolis, MN 55455
`Email: panxx014@umn.edu`

PubH 8475/Stat 8056

Outline

- Mixture model: a generative model
  new: L_1 penalization for variable selection;
  Pan et al (2006, Bioinformatics)
- Transductive SVM (TSVM):
  Wang, Shen & Pan (2007, CM; 2009, JMLR)
- Self-supervised learning: DL
  Chen et al (2020)

Introduction

- Biology: Do human blood outgrowth endothelial cells (BOECs) belong to or are closer to large vessel endothelial cells (LVECs) or microvascular endothelial cells (MVECs)?
- Why important? BOECs are being explored for efficacy in endothelial-based gene therapy (Lin et al 2002), and as being useful for vascular diagnostic purposes (Hebbel et al 2005); in each case, it is important to know whether BOEC have characteristics of MVECs or of LVECs.

- Jiang (2005) conducted a genome-wide comparison: microarray gene expression profiles for BOEC, LVEC and MVEC samples were clustered; it was found that BOEC samples tended to cluster together with MVEC samples, suggesting that BOECs were closer to MVECs.
- Two potential shortcomings:
  1. Used hierarchical clustering; ignoring the known classes of LVEC and MVEC samples;
     Alternative? Semi-supervised learning: treating LVEC and MVEC as known while BOEC unknown (see McLachlan and Basford 1988; Zhu 2006 for reviews).
     Here it requires learning a novel class: BOEC may or may not belong to LVEC or MVEC.
  2. Used only 37 genes that best discriminate b/w LVEC and MVEC.
     Important: result may critically depend on the features or genes being used; the few genes might not reflect the whole picture.
     Alternative? Start with more genes; but ...
     A dilemma: too many genes might lead to covering true clustering structures; to be shown later.

- For high-dimensional data, necessary to have feature selection, preferably embedded within the learning framework – automatic/simultaneous feature selection.
- In contrast to sequential methods: first selecting features and then fitting/learning a model;
  Pre-selection may perform terribly;
  Why: selected features may not be relevant at all to uncovering interesting clustering structures, due to the separation between the two steps.
- A penalized mixture model: semi-supervised learning; automatic variable selection simultaneously with model fitting.

Semi-Supervised Learning via Standard Mixture Model

- Data
  Given n K-dimensional obs's: $x_1, \ldots, x_n$; the first n_0 do not have class labels while the last n_1 have.
  There are $g = g_0 + g_1$ classes: the first g_0 unknown/novel classes to be discovered. while the last g_1 known.
  $z_{ij} = 1$ iff x_j is **known** to be in class i; $z_{ij} = 0$ o/w.
  Note: z_{ij}'s are missing for $1 \leq j \leq n_0$.
- The log-likelihood is

$$\log L(\Theta) = \sum_{j=1}^{n_0} \log[\sum_{i=1}^{g} \pi_i f_i(x_j; \theta_i)] + \sum_{j=n_0+1}^{n} \log[\sum_{i=1}^{g} z_{ij} \pi_i f_i(x_j; \theta_i)].$$

- Common to use the EM to get MLE.

Penalized Mixture Model

- Penalized log-likelihood: use a weighted L_1 penalty;

$$\log L_P(\Theta) = \log L(\Theta) + \lambda \sum_i \sum_k w_{ik} |\mu_{ik}|,$$

  where w_{ik}'s are weights to be given later.
- Penalty: model regularization; Bayesian connection.
- Assume that the data have been standardized so that each feature has sample mean 0 and sample variance 1.
- Hence, for any k, if $\mu_{1k} = ... = \mu_{gk} = 0$, then feature k will not be used.
- L_1 penalty serves to obtain a sparse solution: μ_{ik}'s are automatically set to 0, realizing variable selection.

- EM algorithm: E-step and M-step for other parameters are the same as in the usual EM, except M-step for μ_{ik};

$$\hat{\pi}_i^{(m+1)} = \sum_{j=1}^{n} \tau_{ij}^{(m)} / n, \tag{1}$$

$$\hat{\sigma}_k^{2,(m+1)} = \sum_{i=1}^{g} \sum_{j=1}^{n} \tau_{ij}^{(m)} (x_{jk} - \hat{\mu}_{ik}^{(m)})^2 / n, \tag{2}$$

$$\hat{\mu}_i^{(m+1)} = \text{sign}(\tilde{\mu}_i^{(m+1)}) \left(|\tilde{\mu}_i^{(m+1)}| - \frac{\lambda}{\sum_j \tau_{ij}^{(m)}} V^{(m)} w_i \right)_+ \tag{3}$$

where

$$\tau_{ij}^{(m)} = \begin{cases} \frac{\pi_i^{(m)} f_i(x_j;\theta_i^{(m)})}{f(x_j;\Theta^{(m)})}, & \text{if } 1 \leq j \leq n_0 \\ z_{ij}, & \text{if } n_0 < j \leq n \end{cases} \tag{4}$$

$$\tilde{\mu}_i^{(m+1)} = \sum_{j=1}^{n} \tau_{ij}^{(m)} x_j / \sum_{j=1}^{n} \tau_{ij}^{(m)} \tag{5}$$

Model Selection

- To determine g_0 (and λ), use BIC (Schwartz 1978)

$$BIC = -2 \log L(\hat{\Theta}) + \log(n) d,$$

  where $d = g + K + gK - 1$ is the total number of unknown parameters in the model; the model with a minimum BIC is selected (Fraley and Raftery 1998).

- For the penalized mixture model, Pan and Shen (2007) proposed a modified BIC:

$$BIC = -2 \log L(\hat{\Theta}) + \log(n) d_e,$$

  where $d_e = g + K + gK - 1 - q = d - q$ with $q = \#\{\hat{\mu}_{ik} : \hat{\mu}_{ik} = 0\}$, an estimate of the "effective" number of parameters.

Real Data

- 28 LVEC and 25 MVEC samples from Chi et al (2003); cDNA arrays.
- 27 BOEC samples; Affy arrays.
- Combined data: 9289 unique genes in both data.
- Need to minimize systematic bias due to different platforms.
- 6 human umbilical vein endothelial cell (HUVEC) samples from each of the two datasets.
- Jiang studied 64 possible combinations of a three-step normalization procedure and identified the one maximizing the extent of mixing of the 12 HUVEC samples.
- Normalized the data in the same way

- $g_0 = 0$ or 1; $g_1 = 2$.
- 6 models: 1) 3 methods: standard, penalized with $w = 0$, and penalized with $w = 1$; 2 values of g_0: 0 or 1.
- The EM randomly started 20 times with the starting values from the K-means output.
- At convergence, used the posterior probabilities to classify BOEC samples, as well as LVEC and MVEC samples.
- Used 3 sets of the genes in the starting model.
- Using 37 genes best discriminating LVEC and MVEC:

Table: Semi-supervised learning with 37 genes. The BIC values of the six models (from left to right and from top to bottom) were 2600, 2549, 2510, 2618, 2520 and 2467 respectively.

	$g_0 = 0, g_1 = 2$					
	$\lambda = 0$		$\lambda = 5, w = 0$		$\lambda = 2, w = 1$	
Sample	1	2	1	2	1	2
BOEC	1	26	6	21	0	27
LVEC	24	4	25	3	25	3
MVEC	2	23	3	22	2	23

	$g_0 = 1, g_1 = 2$								
	$\lambda = 0$			$\lambda = 6, w = 0$			$\lambda = 3, w = 1$		
Sample	1	2	3	1	2	3	1	2	3
BOEC	13	1	13	17	1	9	16	0	11
LVEC	1	24	3	2	24	2	1	25	2
MVEC	0	1	24	2	1	24	0	2	23

Table: Numbers of the 37 features with zero mean estimates.

	$g_0 = 0,\ g_1 = 2$							
	$\lambda = 5,\ w = 0$			$\lambda = 2,\ w = 1$				
Cluster	1	2	All	1	2	All		
#Zeros	11	11	11	14	18	14		
	$g_0 = 1,\ g_1 = 2$							
	$\lambda = 6,\ w = 0$				$\lambda = 3,\ w = 1$			
Cluster	1	2	3	All	1	2	3	All
#Zeros	21	10	11	5	24	18	20	12

- Using top 1000 genes discriminating LVEC and MVEC;
- Using top 1000 genes with largest sample variances;
- —-similar results!

TSVM

- Labeled data: (x_i, y_i), $i = 1, ..., n_l$;
  Unlabeled data: (x_i), $i = n_l + 1, ..., n$.
- SVM: consider linear kernel; i.e.

$$f(x) = \beta_0 + \beta' x.$$

- Estimation in SVM:

$$\min_{\beta_0, \beta} \sum_{i=1}^{n_l} L(y_i f(x_i)) + \lambda_1 ||\beta||^2$$

- TSVM: aim the same $f(x) = \beta_0 + \beta' x$.

- Estimation in TSVM:

$$\min_{\{y^*_{n_l+1},\dots,y^*_n\},\beta_0,\beta} \sum_{i=1}^{n_l} L(y_i f(x_i)) + \lambda_1 ||\beta||^2 + \lambda_2 \sum_{i=n_l+1}^{n} L(y_i^* f(x_i))$$

- Equivalently (Wang, Shen & Pan 2007; 2009, JMLR),

$$\min_{\beta_0,\beta} \sum_{i=1}^{n_l} L(y_i f(x_i)) + \lambda_1 ||\beta||^2 + \lambda_2 \sum_{i=n_l+1}^{n} L(|f(x_i)|)$$

- Computational algorithms DO matter!
- Active research going on: e.g. with EHRs

Table: **Linear learning:** Averaged test errors as well as the estimated standard errors (in parenthesis) of SVM with labeled data alone, TSVM^{Light}, and TSVM^{DCA}, over 100 pairs of training and testing samples, in the simulated and benchmark examples.

Data	SVM	TSVM^{Light}	TSVM^{DCA}
Example 1	.345(.0081)	.230(.0081)	.220(.0103)
Example 2	.333(.0129)	.222(.0128)	.203(.0088)
WBC	.053(.0071)	.077(.0113)	.037(.0024)
Pima	.328(.0092)	.316(.0121)	.314(.0086)
Ionosphere	.257(.0097)	.295(.0085)	.197(.0071)
Mushroom	.232(.0135)	.204(.0113)	.206(.0113)
Email	.216(.0097)	.227(.0120)	.196(.0132)

Table: **Nonlinear learning with Gaussian kernel:** Averaged test errors as well as the estimated standard errors (in parenthesis) of SVM with labeled data alone, TSVM^{Light}, and TSVM^{DCA}, over 100 pairs of training and testing samples, in the simulated and benchmark examples.

Data	SVM	TSVM^{Light}	TSVM^{DCA}
Example 1	.385(.0099)	.267(.0132)	.232(.0122)
Example 2	.347(.0119)	.258(.0157)	.205(.0091)
WBC	.047(.0038)	.037(.0015)	.037(.0045)
Pima	.353(.0089)	.362(.0144)	.330(.0107)
Ionosphere	.232(.0088)	.214(.0097)	.183(.0103)
Mushroom	.217(.0135)	.217(.0117)	.185(.0080)
Email	.226(.0108)	.275(.0158)	.192(.0110)

Self-Supervised Learning

- Ref: Chen et al (2020);
  also called contrastive learning, semi-supervised learning.
- DL: used for pre-training/transfer learning; self-training.
- $f()$: a NN *base encoder*;
  a **target** NN up to the layer prior/close to output.
  Representation learning.
- $g()$: A small NN *projection head*.
  e.g. a FFN with 1 hidden layer, $g(h) = W_2\sigma(W_1 h)$.
- To train a new NN $f + g$: $f()$ then $g()$.
- Data augmentation: data/image transformations, e.g.,
  random cropping + resizing; rotating; cutting out; color
  distortions; Gaussian blurring; ...

- $x_i \Longrightarrow \tilde{x}_{2i-1} = t(x_i)$, $\tilde{x}_{2i} = t'(x_i)$.
- $h_{2i-1} = f(\tilde{x}_{2i-1})$, $h_{2i} = f(\tilde{x}_{2i})$,
  $z_{2i-1} = g(h_{2i-1})$, $z_{2i} = g(h_{2i})$
- Contrastive loss: $s_{i,j} = z_i' z_j / ||z_i|| ||z_j||$,

$$L(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} I(k \neq i) \exp(s_{i,k}/\tau)},$$

- The NN "$f + g$" is trained with each minibatch by

$$\min \frac{1}{2N} \sum_{k=1}^{N} [L(2k-1, 2k) + L(2k, 2k-1)].$$

- Take $f()$ and throw away $g()$
- Then train "$f()$ plus output layer(s)" with some labeled data.
  better than training "$f()$ plus output layer(s)" from scratch.
  Note: **no labels** for x_i's!