Thursday, 11/5/2020

EE 660

MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins

Lecture 22

Lecture 22 | **EE 660** | **Nov 5, 2020**

Lecture 22 announcements

- Homework 8 is due tomorrow.
- Homework 9 will be posted

Lecture 22 outline

- Semi-supervised learning (SSL) (part 2)
  - Self-training models
  - Mixture models and parametric classification (SL)
  - Mixture models and parametric classification (SSL)

Self-Training Models

Use their prediction $\hat{f}(\underline{x}^{(U)})$ for additional training.

[Self-tr. & prop 1NN algorithms]
[Small ex.]

Variant of prop. 1NN alg.: prop. kNN — use a kNN for the S.L. module.

[Prop. 1NN applied to 100 aliens data]

This alg. tends to work well when data forms C dense, well separated clusters, or C long chains of data that are separated.

SSL Self-Training Models [Zhu and Goldberg text]

Algorithm 2.4. Self-training. (wrapper)

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat:
3. Train $\hat{f}$ from L using supervised learning.
4. Apply $\hat{f}$ to the unlabeled instances in $\hat{U}$.
5. Remove a subset S from U; add $\{(\mathbf{x}, \hat{f}(\mathbf{x})) | \mathbf{x} \in S\}$ to L.

S: data points with highest confidence of $\hat{f}$ prediction.

Assumption: data pts. with highest confidence of $\hat{f}(\underline{x})$ tend to be correct.

Specific example:

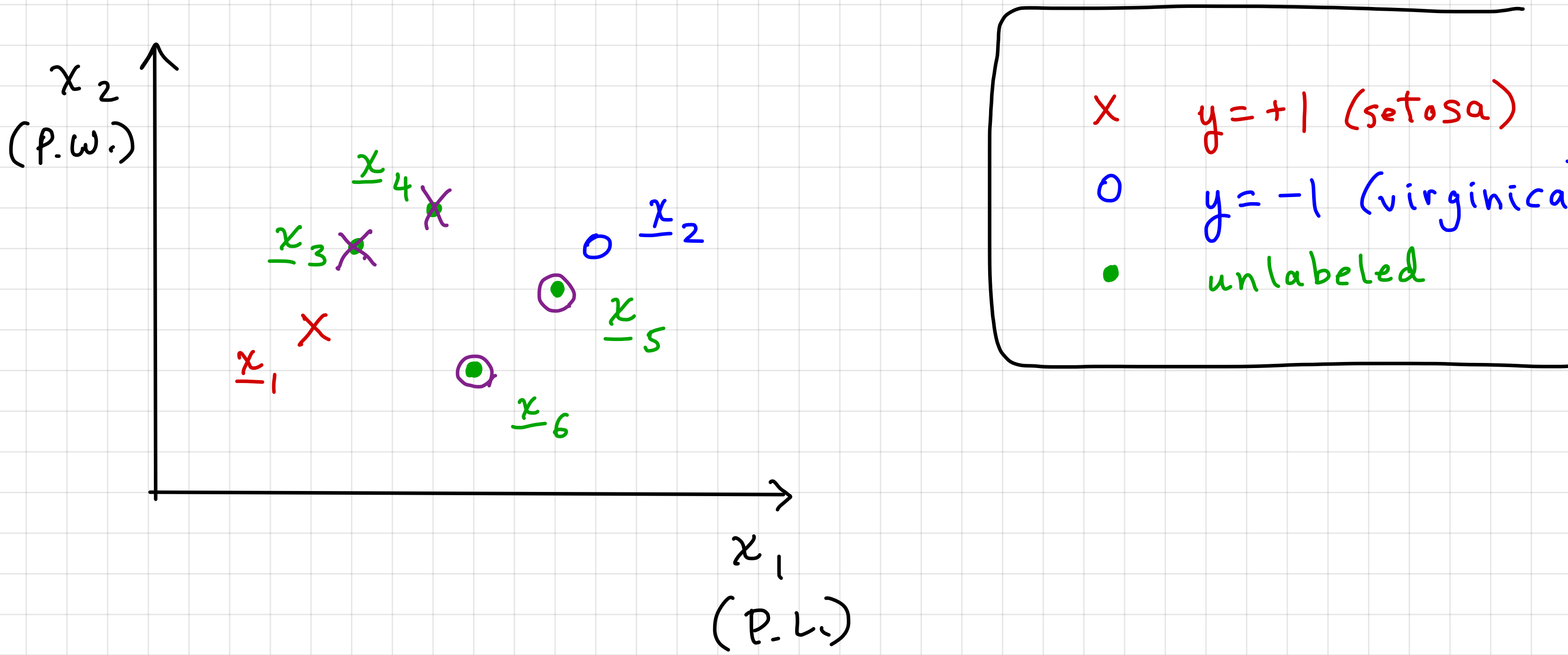
Algorithm 2.7. Propagating 1-Nearest-Neighbor.

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, distance function $d()$.
1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat until U is empty:
3. Select $\mathbf{x} = \operatorname{argmin}_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$. ← find closest 2 pts; one from L and one from U.
4. Set $\hat{f}(\mathbf{x})$ to the label of $\mathbf{x}$'s nearest instance in L. Break ties randomly.
5. Remove $\mathbf{x}$ from U; add $(\mathbf{x}, f(\mathbf{x}))$ to L.

pt. in S

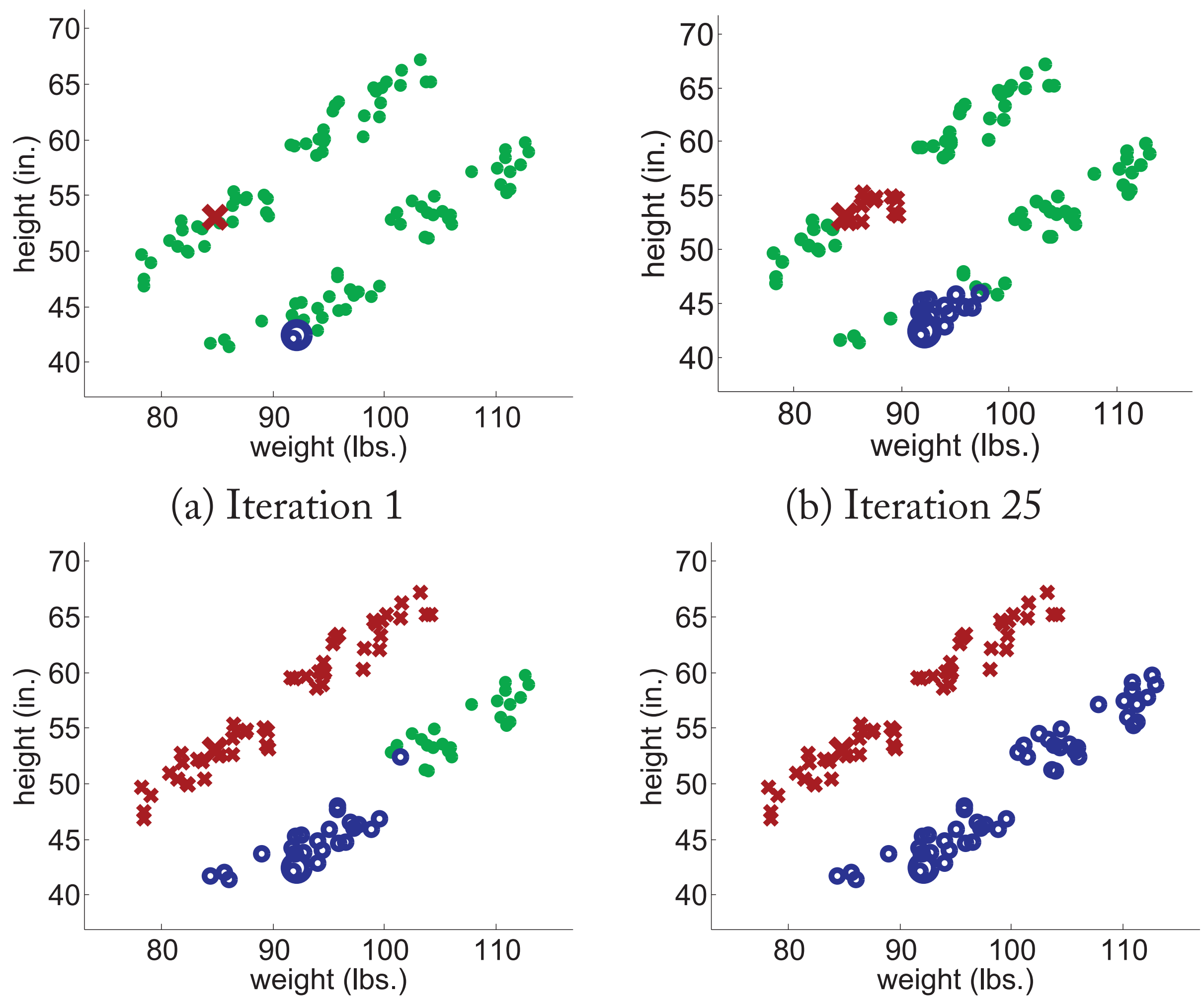
Excerpts are from Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Morgan and Claypool, 2009)

Propagating 1-NN example:



iteration #	closest unlabeled pt. $\underline{x}$	$\hat{f}(\underline{x})$	L	U
1	$\underline{x}_5$	-1	$\underline{x}_1, \underline{x}_2, (\underline{x}_5, -1)$	$\underline{x}_3, \underline{x}_4, \underline{x}_6$
2	$\underline{x}_3$	$+1$	$\underline{x}_1, \underline{x}_2, (\underline{x}_5, -1), (\underline{x}_3, +1)$	$\underline{x}_4, \underline{x}_6$
3	$\underline{x}_4$	$+1$	$\underline{x}_1, \underline{x}_2, (\underline{x}_5, -1), (\underline{x}_3, +1), (\underline{x}_4, +1)$	$\underline{x}_6$
4	$\underline{x}_6$	-1	all	$\emptyset$

p.5



(a) Iteration 1 (b) Iteration 25

(c) Iteration 74 (d) Final labeling of all instances

Figure 2.3: Propagating 1-nearest-neighbor applied to the 100-little-green-alien data.

Excerpts are from Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Morgan and Claypool, 2009)

Mixture Models and Parametric Classification – Supervised Learning (SL)

Suppose we model $p(\underline{x}|y)$ as a pdf with some unknown parameters, e.g.:

$$p(\underline{x}|y,\underline{\theta}) = N(\underline{x}|\underline{\mu}_y, \underline{\underline{\Sigma}}_y)$$

y = class index

$\underline{\mu}_y$ and $\underline{\underline{\Sigma}}_y$, $y=1,2,\cdots,C$, are unknown. $\Rightarrow$ parameters $\underline{\theta}$.

Posterior predictive $p(y|\underline{x}) = ?$

$$p(y|\underline{x}) = \frac{p(\underline{x}|y)\,p(y)}{p(\underline{x})}, \quad p(\underline{x}) = \sum_{y'} p(\underline{x}|y')\,p(y')$$

prior

= a mixture density.

To find $p(\underline{x}|y)$, how can estimate $\underline{\theta}$? (in SL)

One way: MLE

$$\underline{\hat{\theta}}_{MLE} = \underset{\underline{\theta}}{\operatorname{argmax}}\; p(\mathcal{D}|\underline{\theta}) = \underset{\underline{\theta}}{\operatorname{argmax}}\; \ln p(\mathcal{D}|\underline{\theta})$$

$$\underline{\hat{\theta}}_{MLE} = \underset{\underline{\theta}}{\operatorname{argmax}} \sum_{i=1}^{\ell} \ln p(\underline{x}_i, y_i \mid \underline{\theta})$$

$$= \underset{\underline{\theta}}{\operatorname{argmax}} \sum_{i=1}^{\ell} \ln \Big[p(\underline{x}_i \mid y_i, \underline{\theta}) \underbrace{p(y_i \mid \underline{\theta})}_{\text{prior on } y \,=\, \pi_y,\ y = \text{class index.}} \Big]$$

→ For a given fcn. $p(\underline{x} \mid y, \underline{\theta})$, solution gives $\underline{\hat{\theta}}_{MLE}$.

Given our estimates $\underline{\hat{\theta}}$, we have:

$$p(\underline{x}) = \sum_{y'} p(\underline{x} \mid y', \underline{\hat{\theta}})\, p(y' \mid \underline{\hat{\theta}})$$

Let $\pi_{y'} = p(y') = p(y' \mid \underline{\hat{\theta}})$

$$\therefore\ \underbrace{p(\underline{x}) = \sum_{y'} \pi_{y'}\, p(\underline{x} \mid y', \underline{\hat{\theta}}) = \text{a mixture density.}}$$

Mixture Models for SSL

We want to find $p(y\,|\,\underline{x})$

Let's model each class as a specified density with unknown parameters:

(1) $$p(\underline{x}, y\,|\,\underline{\theta}) = p(\underline{x}\,|\,y, \underline{\theta})\, p(y\,|\,\underline{\theta})$$ (assuming $\underline{\theta}$ is not random)

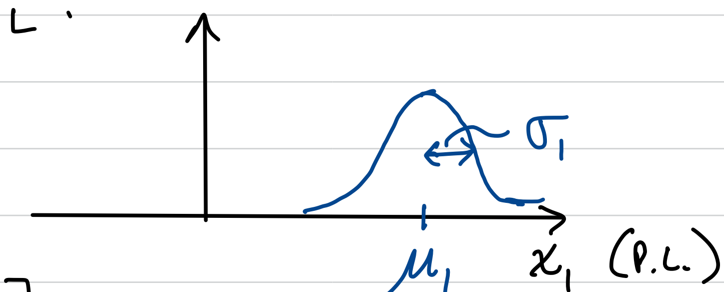
$$= \underbrace{p(\underline{x}\,|\,y, \underline{\theta})}_{\text{class-conditional density, conditioned on } \underline{\theta}.}\ \underbrace{p(y)}_{\pi_y}$$

Our model:

Labeled data $\mathcal{D}_L$:

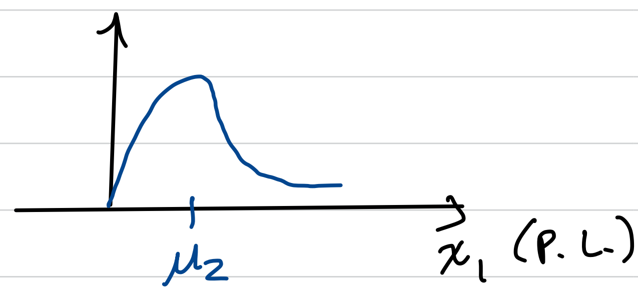
(setosa)

$p(\underline{x}\,|\,y=1, \underline{\theta})$



(virginica)

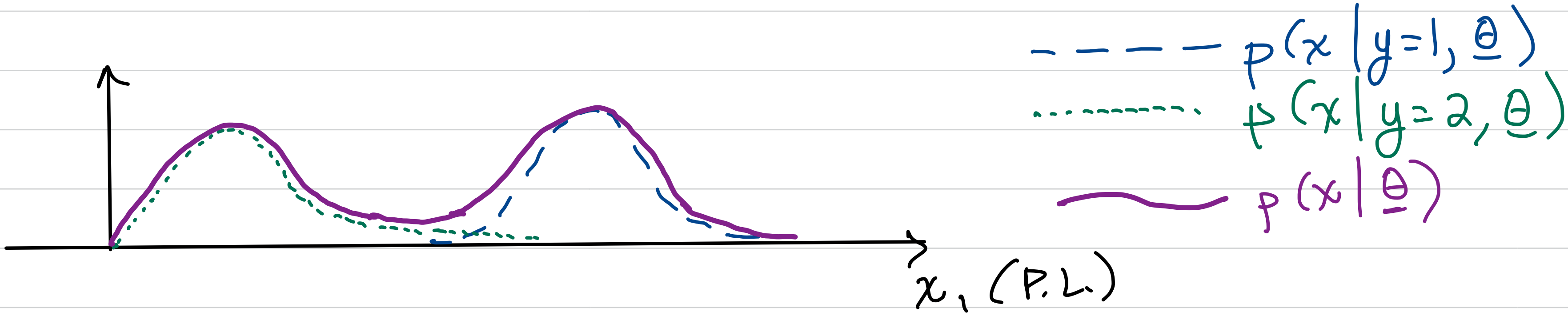
$p(\underline{x}\,|\,y=2, \underline{\theta})$



$$\underline{\theta} = \begin{bmatrix} \mu_1 \\ \sigma_1 \\ \mu_2 \\ \vdots \end{bmatrix}$$

Unlabeled data $\mathcal{D}_U$:

(2) $$p(\underline{x}|\underline{\theta}) = \sum_{y=1}^{c} \underbrace{p(\underline{x}|y,\underline{\theta})}_{\text{component density}} \overbrace{\underbrace{p(y|\underline{\theta})}_{\text{mixing parameter}}}^{\pi_y} = \text{a mixture density}$$



$\Rightarrow$ How do we use both models (for $\mathcal{D}_L$ and $\mathcal{D}_U$) together to estimate $\underline{\theta}$?

Find $\underline{\theta}$ from data using MLE

(3) $$\hat{\underline{\theta}}_{MLE} = \underset{\underline{\theta}}{\operatorname{argmax}}\ p(\mathcal{D}|\underline{\theta}) = \underset{\underline{\theta}}{\operatorname{argmax}}\ \ln p(\mathcal{D}|\underline{\theta})$$

$$p(\mathcal{D}|\underline{\theta}) = \prod_{i=1}^{l} p(\underline{x}_i, y_i|\underline{\theta}) \prod_{i=l+1}^{u} p(\underline{x}_i|\underline{\theta})$$

$$\ln p(\mathcal{D}|\underline{\theta}) = \sum_{i=1}^{l} \ln p(\underline{x}_i, y_i|\underline{\theta}) + \sum_{i=l+1}^{u} \ln p(\underline{x}_i|\underline{\theta})$$

(4) $$\ln p(\mathcal{D}|\underline{\theta}) = \underbrace{\sum_{i=1}^{l}\left[\ln p(\underline{x}_i|y_i, \underline{\theta}) + \ln p(y_i|\underline{\theta})\right]}_{\mathcal{D}_L} + \underbrace{\sum_{i=l+1}^{u} \ln\left[\sum_{y=1}^{C} p(\underline{x}_i|y, \underline{\theta})\underbrace{p(y|\underline{\theta})}_{\pi_y}\right]}_{\mathcal{D}_U}$$

Let $\mathcal{D} \triangleq \{\mathcal{D}_L, \mathcal{D}_U\}$;

and treat the unknown y_i in $\mathcal{D}_U$ as "hidden variables", denoted $\mathcal{H}$.

→ If we knew $\mathcal{H}$, we could: $\underline{\theta}_{MLE} = \underset{\underline{\theta}}{\operatorname{argmax}}\{\ln p(\mathcal{D}, \mathcal{H}|\underline{\theta})\}$ (as in SL). But, don't know $\mathcal{H}$.

→ Use Expectation maximization (EM) to estimate $\mathcal{H}$ and $\underline{\theta}$.