Lecture 23 announcements

- HW 12 has been posted
- My office hours tomorrow will be changed to 12:00 - 1:00 PM

Lecture 23 outline

- Start Semi-Supervised Learning (SSL)
  - Introduction
  - Self-training algorithms
  - Mixture models and parametric classification (for supervised learning)

SEMI-SUPERVISED LEARNING (SSL)

For textbook: see HW 12 Reading.

TRAINING SET:

LABELLED INSTANCES $\mathcal{D}_L = \{(\underline{x}_i, y_i)\}_{i=1}^{l}$

AND

UNLABELLED INSTANCES $\mathcal{D}_U = \{\underline{x}_j\}_{j=l+1}^{l+u}$

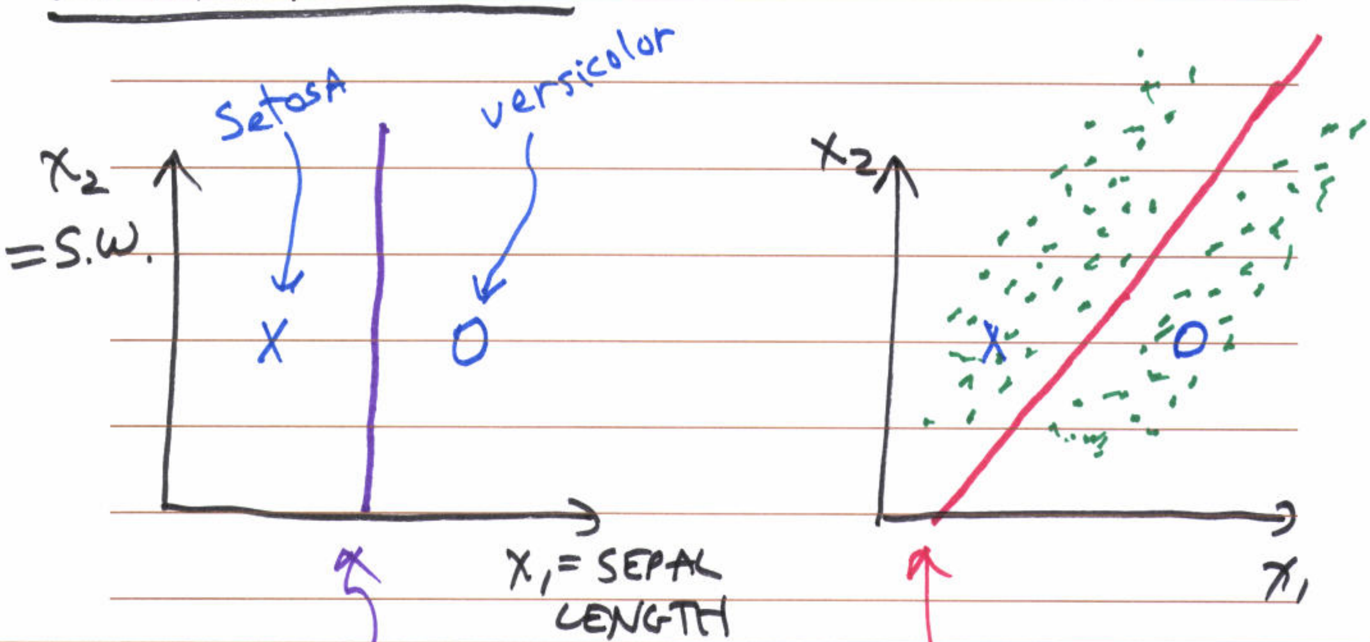
WHY?

1. CAN BE EXPENSIVE TO GET LABELS ON DATA POINTS,
2. OFTEN HAVE ACCESS TO PLENTIFUL UNLABELED DATA POINTS.

→ THE GOAL IS TO TRAIN A SYSTEM USING BOTH SETS $\mathcal{D}_L$ AND $\mathcal{D}_U$, AND GET BETTER OUT-OF-SAMPLE PERFORMANCE THAN TRAINING

ON $\mathcal{D}_L$ ALONE.

IS THIS POSSIBLE?



SUPERVISED LEARNING (SL) BOUNDARY

POTENTIAL SSL BOUNDARY.

ASSUMPTIONS

1. LABELED SAMPLES ARE REPRESENTATIVE (NOT OUTLIERS).
2. ALL DATA POINTS ARE DRAWN iid FROM (UNDERLYING DENSITY)

$$p(\underline{x} \mid y)$$

Assume consistent

UNLABELED: $p(\underline{x})$

LABELED: $p(\underline{x}|y)$, OR $p(y|\underline{x})$

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

TWO MAJOR TYPES OF SSL:

INDUCTIVE SSL

LEARNS $\hat{y} = \hat{f}(\underline{x})$ OVER ALL FEATURE SPACE $\mathcal{X}$.

TRANSDUCTIVE SSL

LEARNS $\hat{y}_i = \hat{f}(\underline{x}_i)\ \forall\ \underline{x}_i \in \mathcal{D}_U$.

SOME SSL MODELS / ALGORITHMS

SELF-TRAINING MODELS

SL ON $\mathcal{D}_L$

predict $\mathcal{D}_U$

USE PREDICTION OF (SOME OF) $\mathcal{D}_U$ FOR ADDITIONAL TRAINING

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* U.
5. *Remove a subset* S *from* U; *add* $\{(\mathbf{x}, \hat{f}(\mathbf{x})) | \mathbf{x} \in S\}$ *to* L.

→ ~~SA~~ DATA POINTS WITH HIGHEST CONFIDENCE OF $\hat{f}(\underline{x})$ PREDICTION.

ASSUMPTION: DATA POINTS WITH HIGHEST CONFIDENCE OF $\hat{f}(\underline{x})$ PREDICTION TEND TO BE CORRECT.

SPECIFIC EX: **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* $\mathbf{x} = \text{argmin}_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$.
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.

FIND THE UNLABELED DATA POINT $\underline{x}_j$ THAT IS CLOSEST TO A LABELED DATA POINT $\underline{x}'_i$.

$\Rightarrow \underline{x}_j$ IS S.

Excerpts are from Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2009)

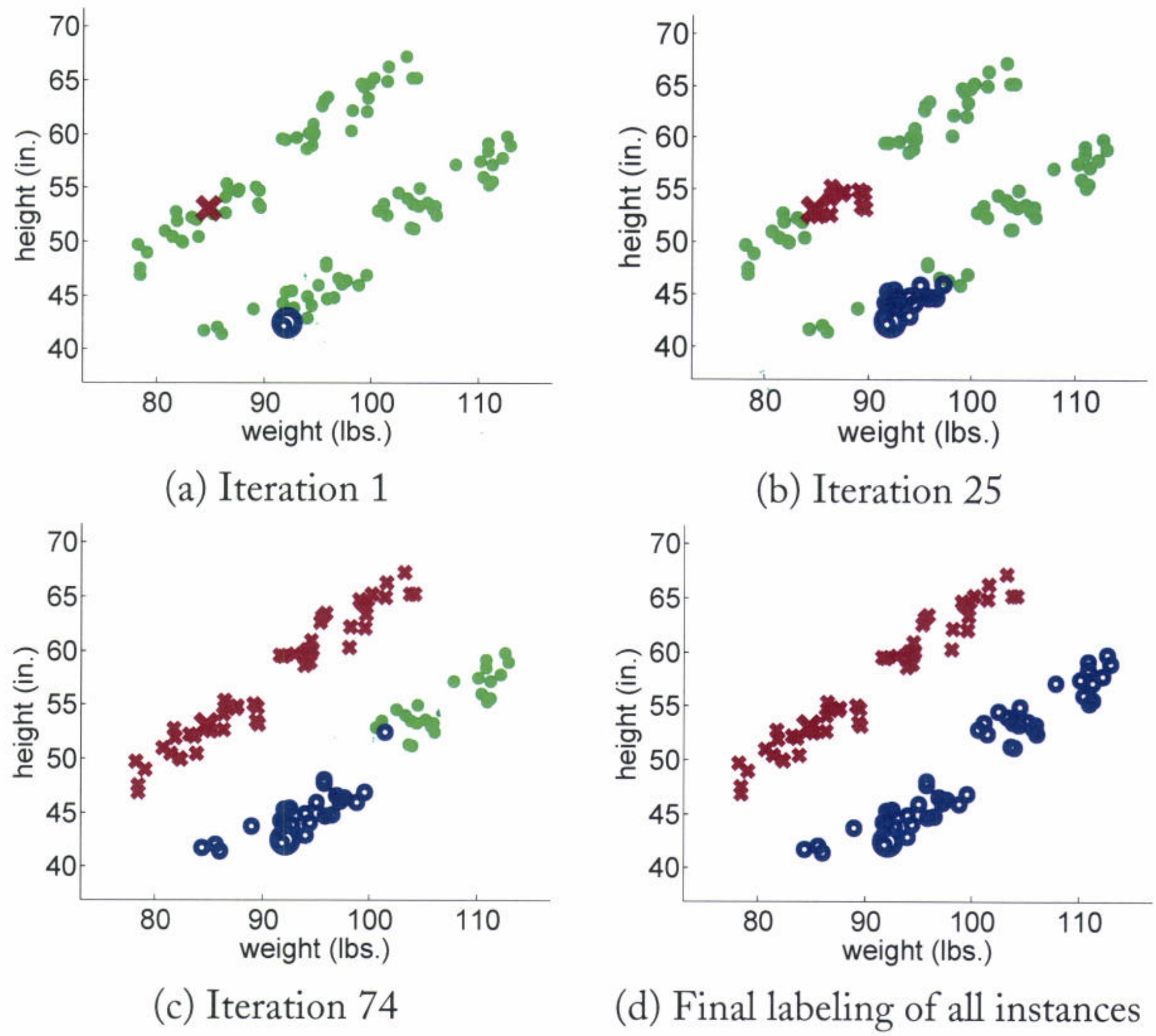


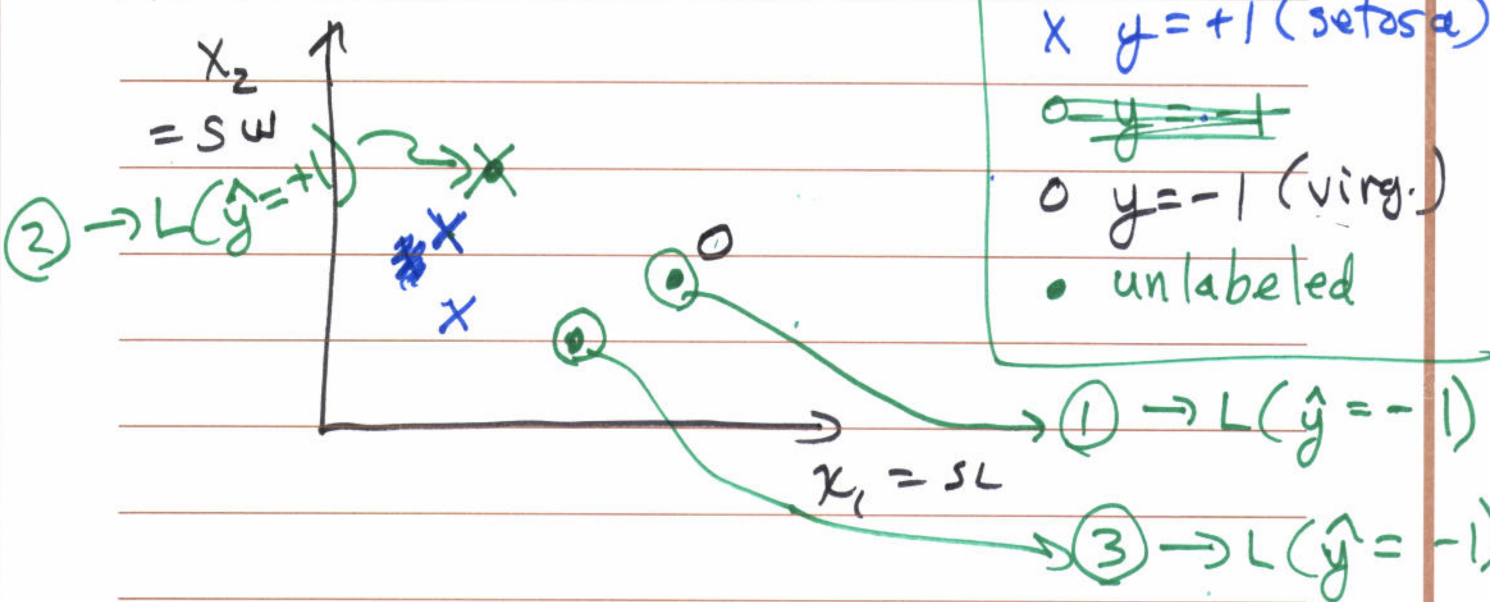
Figure 2.3: Propagating 1-nearest-neighbor applied to the 100-little-green-alien data.

THIS ALG. WORKS WELL IF DATA FORMS C DENSE, WELL-SEPARATED CLUSTERS (1 FOR EACH CLASS).

WHAT IF CLUSTERS AREN'T WELL SEPARATED?

Excerpts are from Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2009)

PROP. 1-NN ex:



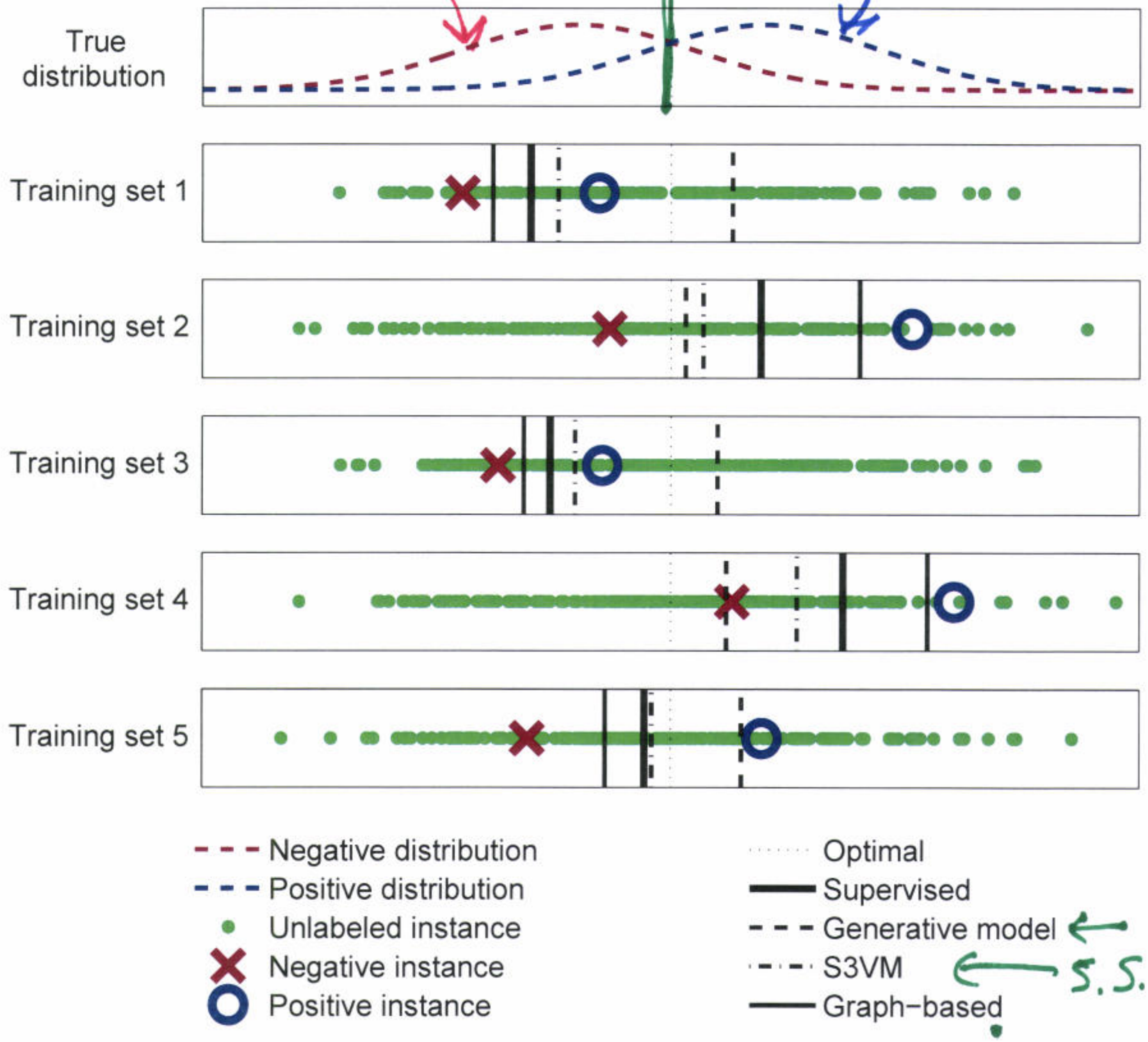
VARIANT – PROP. k-NN, USES k-NN CLASSIFIER INSTEAD.

C=2 CLASSES. 1D.

$p(y=1|x) = \alpha\, p(x|y=1)\, p(y=1)$

$p(y=2|x) = \alpha\, p(x|y=2)\, p(y=2)$

(*)



E.M.

S.S. SVM

Figure 2.2: Two classes drawn from overlapping Gaussian distributions (top panel). Decision boundaries learned by several algorithms are shown for five random samples of labeled and unlabeled training samples.

(*) — BAYES OPTIMAL DECISION BOUNDARY (min. E_{out}).

Excerpts are from Xiaojin Zhu and Andrew B. Goldberg, *Introduction to Semi-Supervised Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2009)

MIXTURE MODELS AND PARAMETRIC CLASSIFICATION (SL)

SUPPOSE WE MODEL $p(\underline{x} \mid y)$ AS A PARTICULAR pdf WITH SOME UNKNOWN PARAMETERS, e.g.:

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

class index

$\underline{\mu}_y$ AND $\underline{\underline{\Sigma}}_y$, $y = 1, 2, \cdots, C$ ARE UNKNOWN.
PARAMETERS $\underline{\Theta}$.

POSTERIOR PREDICTIVE $p(y \mid \underline{x}) = ?$

$$p(y \mid \underline{x}) = \frac{p(\underline{x} \mid y)\, p(y)}{p(\underline{x})}$$

PRIOR

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

$= $ A MIXTURE DENSITY.

IF WE KNOW $p(\underline{x} \mid y, \underline{\theta})$, HOW ESTIMATE $\underline{\theta}$?

ONE WAY: MLE:

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

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

$$= \underset{\underline{\theta}}{\operatorname{argmax}} \sum_{i=1}^{\ell} \ln \Big[\underbrace{p(\underline{x}_i \mid y_i, \underline{\theta})}_{\text{KNOWN.}} \; \underbrace{p(y_i \mid \underline{\theta})}_{\text{PRIOR ON } y.} \Big]$$