Concept Drift and Sequential Data

Connections between multi-label and time-series learning

Jesse Read

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

université PARIS-SACLAY

02 December, 2016

Outline

Multi-labelled Data

$\mathbf{x} =$ 

$$\mathbf{y} = \{\texttt{sunset}, \texttt{foliage}\}$$
$$\equiv [1, 0, 1, 0, 0, 0]$$

i.e., multiple labels per instance instead of a single label.

Multi-label Learning

The task of building a model to map D inputs to L outputs:

$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ • dataset

$\mathbf{x}_i = [x_1^{(i)}, \ldots, x_D^{(i)}]$ • instance, where $x_j \in \mathbb{R}$

$\mathbf{y}_i = [y_1^{(i)}, \ldots, y_L^{(i)}]$ • label assignment $y_j \in \{0, 1\}$

$\mathbf{h} : \mathcal{X} \to \mathcal{Y}$ • multi-label model

$\hat{\mathbf{y}} = h(\tilde{\mathbf{x}})$ • multi-label classification

$\epsilon = E(\hat{\mathbf{y}}, \mathbf{y})$ • multi-label evaluation

Multi-label Learning

$$\mathcal{L} = \{\texttt{sunset}, \texttt{people}, \texttt{foliage}, \texttt{beach}, \texttt{urban}, \texttt{field}\} \quad (L = 6)$$

$\mathbf{x}_i =$ 

$$\begin{aligned} \hat{\mathbf{y}}_i &= h(\mathbf{x}_i) \\ &= [1, 0, 1, 0, 0, 0] \Leftrightarrow \{\texttt{sunset}, \texttt{foliage}\} \\ &\in \{0,1\}^6 \Leftrightarrow \hat{Y}_i \subseteq \mathcal{L} \end{aligned}$$

i.e., multiple labels per instance instead of a single label.

Single-label vs. Multi-label

Table: Single-label $Y \in \{0, 1\}$

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	0
0	0.9	1	0	1	1
0	0.0	1	1	0	0
1	0.8	2	0	1	1
1	0.0	2	0	1	0
0	0.0	3	1	1	?

Table: Multi-label $Y \subseteq \{\lambda_1, \ldots, \lambda_L\}$

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	$\{\lambda_2, \lambda_3\}$
0	0.9	1	0	1	$\{\lambda_1\}$
0	0.0	1	1	0	$\{\lambda_2\}$
1	0.8	2	0	1	$\{\lambda_1, \lambda_4\}$
1	0.0	2	0	1	$\{\lambda_4\}$
0	0.0	3	1	1	?

Single-label vs. Multi-label

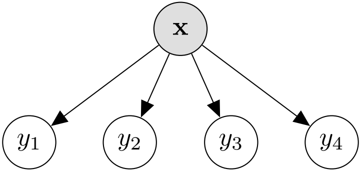
Table: Single-label $Y \in \{0, 1\}$

X_1	X_2	X_3	X_4	X_5	Y
1	0.1	3	1	0	0
0	0.9	1	0	1	1
0	0.0	1	1	0	0
1	0.8	2	0	1	1
1	0.0	2	0	1	0
0	0.0	3	1	1	?

Table: Multi-label $[Y_1, \ldots, Y_L] \in 2^L$

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
1	0.1	3	1	0	0	1	1	0
0	0.9	1	0	1	1	0	0	0
0	0.0	1	1	0	0	1	0	0
1	0.8	2	0	1	1	0	0	1
1	0.0	2	0	1	0	0	0	1
0	0.0	3	1	1	?	?	?	?

Binary Relevance (BR)

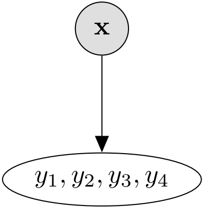


$$\hat{y}_j = h_j(\mathbf{x}) = \underset{y_j \in \{0,1\}}{\operatorname{argmax}}\, p(y_j|\mathbf{x}) \quad \bullet \text{ for each } j = 1, \ldots, L$$

- recall: $y_j^{(i)} = 1$ if the j-th label is relevant to (associated with/assigned to) the i-th instance.
- independent L models (one for each label)
- the j-th model predicts the relevance of the j-th label
- but labels are not independent!

Two Alternatives

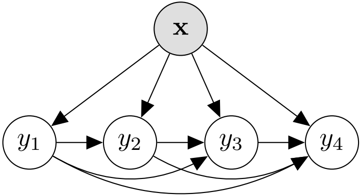
Meta Labels



$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x})$$

- goal: reduce size of $\mathcal{Y}$ (i.e., distinct combinations)

Classifier Chains

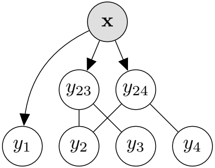


$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \{0,1\}^L}{\operatorname{argmax}}\, \underbrace{p(y_1|\mathbf{x}) \prod_{j=2}^{L} p(y_j|\mathbf{x}, y_1, \ldots, y_{j-1})}_{\text{chain rule}}$$

- goal: reduce connectivity among $Y_1, \ldots, Y_L$

Two Alternatives

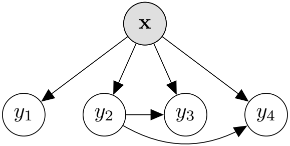
Meta Labels



$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathbf{x})$$

- goal: reduce size of $\mathcal{Y}$ (i.e., distinct combinations)

Classifier Chains



$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \{0,1\}^L} \underbrace{p(y_1|\mathbf{x}) \prod_{j=2}^{L} p(y_j|\mathbf{x}, y_1, \ldots, y_{j-1})}_{\text{chain rule}}$$

- goal: reduce connectivity among $Y_1, \ldots, Y_L$

Outline

Concept Drift

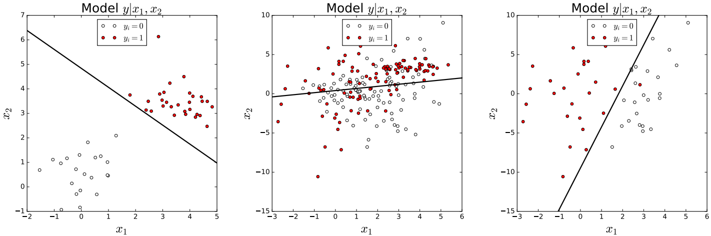


Figure: Single-labelled data and model at $t = 1, \ldots, 50$ (left) and $t = 0, \ldots, 200$ (center) and $t = 150, \ldots, 200$ (right). Concept-drift occurs over $t = 50, \ldots, 150$.

Concept Drift

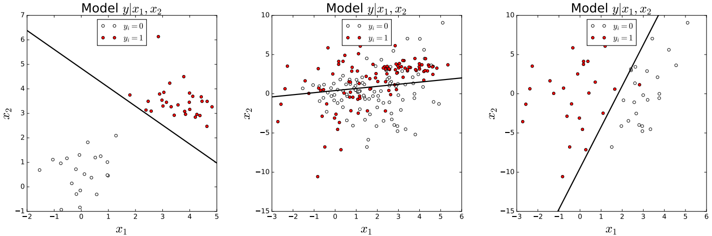


Figure: Single-labelled data and model at $t = 1, \ldots, 50$ (left) and $t = 0, \ldots, 200$ (center) and $t = 150, \ldots, 200$ (right). Concept-drift occurs over $t = 50, \ldots, 150$.

- Model becomes invalid as the concept drifts
- Multi-label concept drift involves also the *label variables*.

Concept Drift

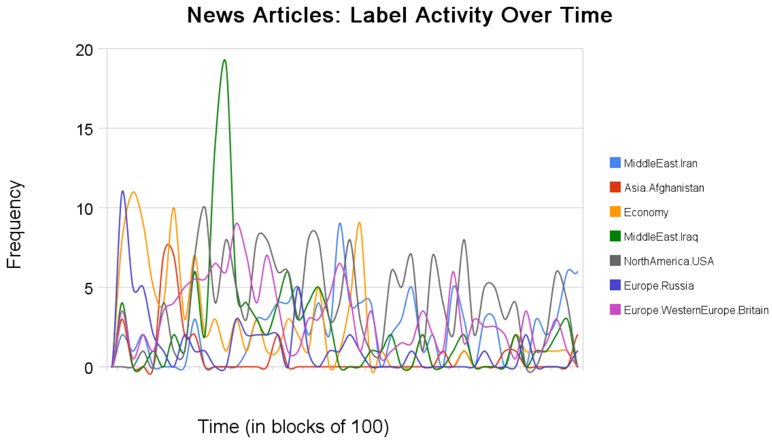


Figure: Label frequency / month over time (until about 2007)

Dealing with Concept Drift

Possible approaches to *detecting* and *responding to* concept drift:

- Just ignore it – batch models must be replaced anyway, kNN and SGD adapt; in other cases can use weighted ensembles/fading factor
- Monitor a predictive performance statistic with a change detector, and reset models
- Monitor the distribution with a change detector, and reset/recalibrate models

(similar to single-labelled data, except more complex measurement)

Detection via Monitoring Accuracy

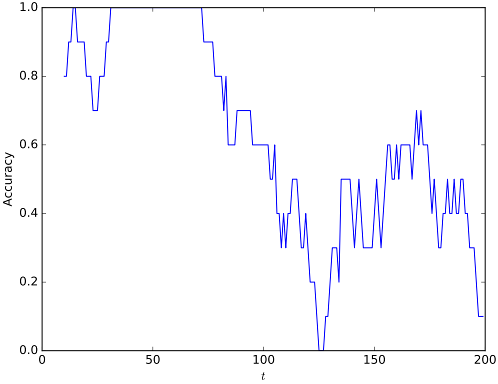


Figure: Accuracy through concept drift ($t = 50, \ldots, 150$).

Label Correlation

Are labels Y_1 and Y_2 correlated (linearly dependent)? This can be quantified with, e.g., Pearson's correlation coefficient:

$$
\begin{aligned}
\rho_{Y_1,Y_2} &= \frac{\mathsf{Cov}(Y_1, Y_2)}{\mathsf{Std}(Y_1)\mathsf{Std}(Y_2)} && (1) \\
&= \frac{\mathbb{E}[(Y_1 - \mu_1)(Y_2 - \mu_2)]}{\sqrt{\mathbb{E}[(Y_1 - \mu_1)^2]}\sqrt{\mathbb{E}[(Y_2 - \mu_2)^2]}} && (2) \\
&= \frac{\sum_{i=1}^{N}[(y_{i1} - \bar{y}_1)(y_{i2} - \bar{y}_2)]}{\sqrt{\sum_{i=1}^{N}[(y_{i1} - \bar{y}_1)^2]\sum_{i=1}^{N}[(y_{i2} - \bar{y}_2)^2]}} && (3)
\end{aligned}
$$

Label Dependence

For more general dependence, one can consider the entropy-based mutual information:

$$I(Y_1, Y_2) = \sum_{y_1 \in \{0,1\}} \sum_{y_2 \in \{0,1\}} p(y_1, y_2) \log \left(\frac{p(y_1, y_2)}{p(y_1) p(y_2)} \right)$$

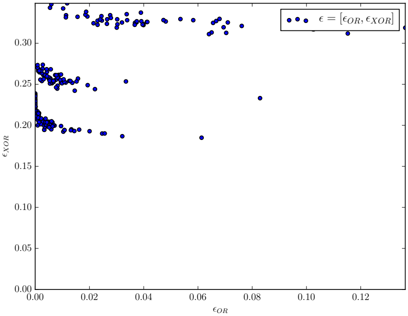
where $p(y_1, y_2)$ is the joint probability, and $p(y_1)$ is the marginal probability. Notice that in the case of independence, $p(y_1, y_2) = p(y_1)p(y_2)$ and thus $\log 1 = 0$.
Where to get these probability distribution functions p? We use, for example,

$$p_{\mathbf{x}}(y_1 = 1, y_2 = 1) \approx f_1^{\mathsf{CC}}(\mathbf{x}) \cdot f_2^{\mathsf{CC}}(\mathbf{x}, 1)$$

$$p_{\mathbf{x}}(y_1 = 1) p_{\mathbf{x}}(y_2 = 1) \approx f_1^{\mathsf{BR}}(\mathbf{x}) \cdot f_2^{\mathsf{BR}}(\mathbf{x})$$

Detection via Monitoring Distribution

Recall the distribution of errors



This shape may change over time – and structures may need to be adjusted to cope (recall: changing structure may improve performance)

Multi-label Concept Drift

Consider the relative frequencies of labels Y_1 and Y_2 at time t,

$$\mathbf{C}_t = \frac{1}{t}\mathbf{Y}^\top\mathbf{Y} = \begin{bmatrix} \tilde{p}_1 & \tilde{p}_{1,2} \\ \tilde{p}_{2,1} & \tilde{p}_2 \end{bmatrix}$$

where $\tilde{p}_{1,2} > \tilde{p}_1\tilde{p}_2$ indicates marginal dependence!

Possible drift (where $\mathbf{C}_t \neq \mathbf{C}_{t+1}$):

- p_1 increases (label Y_1 relatively more frequent)
- p_1 and p_2 both decrease (label cardinality decreasing)
- $p_{1,2}$ changes relative to $p_1 p_2$ (change in marginal dependence relation between the labels)

Multi-label Concept Drift

And when conditioned on input $\mathbf{x}$, we consider the relative frequencies/values of the errors, where, e.g., $E_{ij} = (y_j^{(i)} - \hat{y}_j^{(i)})^2$:

$$\mathbf{C}_t = \frac{1}{t}\mathbf{E}^\top\mathbf{E} = \begin{bmatrix} \tilde{p}_1 & \tilde{p}_{1,2} \\ \tilde{p}_{2,1} & \tilde{p}_2 \end{bmatrix}$$

(if conditional independence, then $\tilde{p}_{1,2} \approx \tilde{p}_1 \cdot \tilde{p}_2$).

Possible drift (where $\mathbf{C}_t \neq \mathbf{C}_{t+1}$):

- p_1 increases (more errors on 1-th label)
- p_1 and p_2 both increase (more errors)
- $p_{1,2}$ changes relative to p_1, p_2 (change in conditional dependence relation)

Outline

Data Streams

$$y_t = h(\mathbf{x}_t) + \epsilon_t$$

- $\mathbf{x}_t \sim p_\theta$, comes i.i.d. from distribution p
- θ (i.e., which defines the distribution) may change over time (concept drift: sudden, gradual, repetitively, ...)
- But the usual implicit assumption is made that

$$p(y_t|\mathbf{x}_t) = p(y_t|\mathbf{x}_t, y_{t-1}, \mathbf{x}_{t-1})$$

- But should we make this assumption? Is there time dependence?

Temporal dependence

The auto-correlation function (basically Pearson's correlation coefficient of a variable with itself, lagged $+1$),

$$\rho_{Y_t,Y_{t+1}} = \frac{\mathsf{Cov}(Y_t, Y_{t+1})}{\mathsf{Std}(Y_t)\mathsf{Std}(Y_{t+1})} \tag{4}$$

$$= \frac{\sum_{t=1}^{T-1}[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{\sqrt{\sum_{t=1}^{T-1}(y_t - \bar{y})^2 \sum_{t=2}^{T}(y_t - \bar{y})^2}} \tag{5}$$

$$\approx \frac{\sum_{t=1}^{T-1}[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{\sum_{t=2}^{T}(y_t - \bar{y})^2} \tag{6}$$

(NB: for large T, the difference in the mean of $Y_1, \ldots, Y_{T-1}$ and of $Y_2, \ldots, Y_T$ can be ignored, hence Eq. (6).)

Temporal dependence

The auto-correlation function (basically Pearson's correlation coefficient of a variable with itself, lagged $+1$),

$$\rho_{Y_t, Y_{t+1}} = \frac{\mathsf{Cov}(Y_t, Y_{t+1})}{\mathsf{Std}(Y_t)\mathsf{Std}(Y_{t+1})} \tag{4}$$

$$= \frac{\sum_{t=1}^{T-1}[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{\sqrt{\sum_{t=1}^{T-1}(y_t - \bar{y})^2 \sum_{t=2}^{T}(y_t - \bar{y})^2}} \tag{5}$$

$$\approx \frac{\sum_{t=1}^{T-1}[(y_t - \bar{y})(y_{t+1} - \bar{y})]}{\sum_{t=2}^{T}(y_t - \bar{y})^2} \tag{6}$$

(NB: for large T, the difference in the mean of $Y_1, \ldots, Y_{T-1}$ and of $Y_2, \ldots, Y_T$ can be ignored, hence Eq. (6).)

We can generalise to $\rho(k)$ to consider the correlation from y_t and y_{t+k} for any lag k (may even be negative).

Temporal dependence

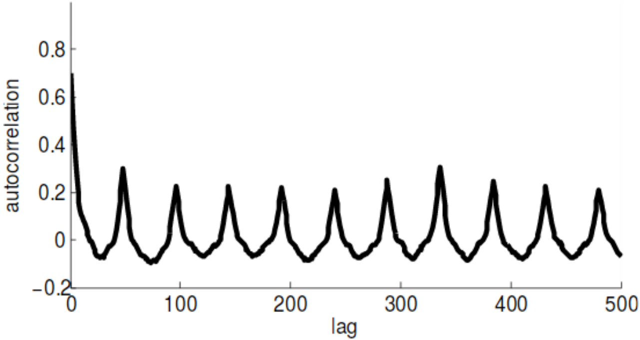


Figure: Auto-correlation function on the Electricity dataset, for $k = 1, 2, \ldots, 500$; source: Indrė Žliobaitė arXiv:1301.3524v1, Jan 2015.

Outline

Streams with Time Dependence

At time t, we see instance x_t, and we wish to make a classification, (e.g., Naive Bayes)

$$\begin{aligned} \hat{y}_t &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(y_t | x_t) \\ &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(x_t | y_t) p(y_t) \end{aligned}$$

- Not a problem, we maintain counts of $y_t = 1$ vs $y_t = 0$, and $x_t = v, y_t = k$ for all values of $v \in \mathcal{X}, k \in \mathcal{L}$
- At time $t + 1$ we get y_t; we can now update counts with (x_t, y_t)!
- We can also measure $\epsilon_t = E(y_t - \hat{y}_t)$, look for drift, etc.

Streams with Time Dependence

At time t, we see instance x_t, and we wish to make a classification, (e.g., Naive Bayes)

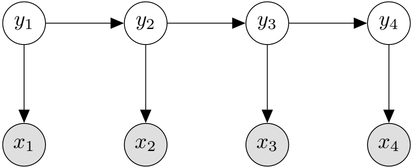
$$\begin{aligned}\hat{y}_t &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(y_t|x_t) \\ &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(x_t|y_t)p(y_t)\end{aligned}$$

- Not a problem, we maintain counts of $y_t = 1$ vs $y_t = 0$, and $x_t = v, y_t = k$ for all values of $v \in \mathcal{X}, k \in \mathcal{L}$
- At time $t + 1$ we get y_t; we can now update counts with (x_t, y_t)!
- We can also measure $\epsilon_t = E(y_t - \hat{y}_t)$, look for drift, etc.

But what if the value at y_t affects the value at y_{t+1} (i.e., the stream exhibits *time* dependence)?

Hidden Markov Model

A generative approach,

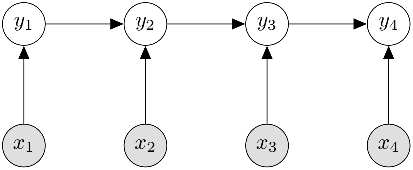


$$
\begin{aligned}
\hat{y}_t &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(\mathbf{y}|\mathbf{x}) \\
&= \operatorname*{argmax}_{y_t \in \{0,1\}} p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}) \\
&= \operatorname*{argmax}_{y_t \in \{0,1\}} p(y_1) \prod_{t=2}^{T} p(x_t|y_t) p(y_t|y_{t-1})
\end{aligned}
$$

recall: $\mathbf{y} = [y_1, \ldots, y_T]$, $\mathbf{x} = [x_1, \ldots, x_T]$.

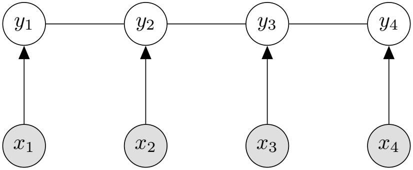
Maximum Entropy Markov Model (MEMM)

A discriminative approach,



$$
\begin{aligned}
\hat{y}_t &= \operatorname*{argmax}_{y_t \in \{0,1\}} p(\mathbf{y}|\mathbf{x}) \\
&= \operatorname*{argmax}_{y_t \in \{0,1\}} p(y_1|x_1) \prod_{t=2}^{T} p(y_t|y_{t-1}, x_t)
\end{aligned}
$$

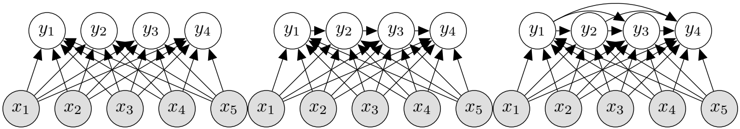
[Linear Chain] Conditional Random Fields (CRFs)



$$
\begin{aligned}
\hat{\mathbf{y}} &= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^T} p(\mathbf{y}|\mathbf{x};\mathbf{w}) \\
&\approx \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^T} \prod_{t=2}^{T} f_t(y_{t-1}, y_t, \mathbf{x})
\end{aligned}
$$

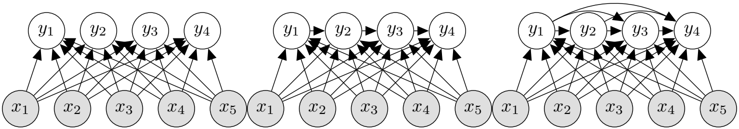
Avoids the label bias / error propagation problem.

Time indices = label indices



- Labels indices can correspond to steps in time (or space)
- Many existing multi-label methodologies can be applied

Time indices = label indices



- Labels indices can correspond to steps in time (or space)
- Many existing multi-label methodologies can be applied

Some differences in sequential data:

- Time dependence (= label dependence!)
- Input observation may be different to each label
- Specific domain assumptions and features

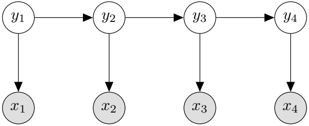
From MEMM to CC

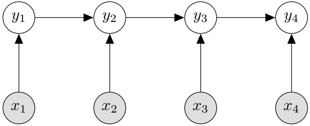
In an HMM, the filtering task / forward algorithm:

$$\hat{y}_t = \underset{y_t \in \{0,1\}}{\operatorname{argmax}}\, p(x_t|y_t)\,p(y_t|y_{t-1})$$

(assume that we have already y_{t-1}, therefore smoothing pass not necessary). We can model this as a discriminative classifier with a Max. Entropy Markov Model (MEMM):

$$\begin{aligned}\hat{y}_t &= \underset{y_t \in \{0,1\}}{\operatorname{argmax}}\, p(y_t|x_t, y_{t-1}) \\ &= \underset{y_t \in \{0,1\}}{\operatorname{argmax}}\, \frac{1}{Z(y_{t-1}, x_t)} \exp\left\{\sum_k f_k(y_t, x_t, y_{t-1})\right\}\end{aligned}$$





From MEMM to CC

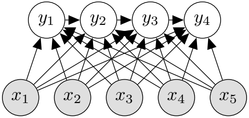
Now assume

- drop normalization constant,
- general input $\mathbf{x}$, generic classifier h, and $t := j$, and that
- we *don't* have y_{t-1} and must input prediction $\hat{y}_t$.

then,

$$
\begin{aligned}
\hat{y}_j &= h(\mathbf{x}, \hat{y}_{t-1}) \\
&= \operatorname*{argmax}_{y_j \in \{0,1\}} f_j(y_j, \mathbf{x}, \hat{y}_{j-1}; \mathbf{w}_j) \\
&\approx \operatorname*{argmax}_{y_j \in \{0,1\}} p(Y_j = y_j | X = \mathbf{x}, Y_{j-1} = \hat{y}_{j-1})
\end{aligned}
$$

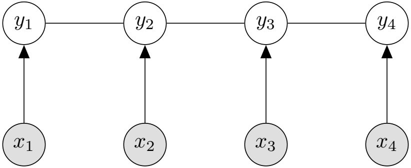
we obtain a *singly-linked*, greedy, classifier chain (CC)!

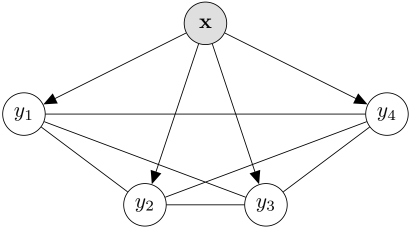


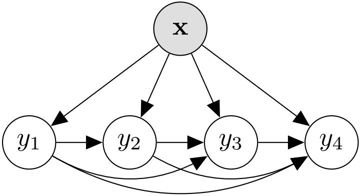
From CRF to PCC

$$
\begin{aligned}
\hat{\mathbf{y}}_t &= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} p(\mathbf{y}|\mathbf{x};\mathbf{w}) \\
&= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} \exp\left\{\sum_k w_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\} \quad \bullet \text{ CRF inference} \\
&= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} \prod_{t=1}^{T} \exp\left\{w_t \cdot f_t(y_t, y_{t-1}, \mathbf{x}_t)\right\} \quad \bullet\ e^{a+b} = e^a e^b \\
&= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} \prod_{t=1}^{T} f_t(y_t, y_{t-1}, \mathbf{x}_t; \mathbf{w}_t) \quad \bullet \text{ a generic fn } f \\
&= \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} f_1(y_1, \mathbf{x}) \prod_{j=2}^{L} f_j(y_1, \ldots, y_{j-1}, \mathbf{x}) \quad \bullet \text{ PCC} \\
&\approx \operatorname*{argmax}_{\mathbf{y}\in\{0,1\}^L} p(y_1|\mathbf{x}) \prod_{j=2}^{L} p(y_j|y_1, \ldots, y_{j-1}, \mathbf{x})
\end{aligned}
$$

From CRF to PCC

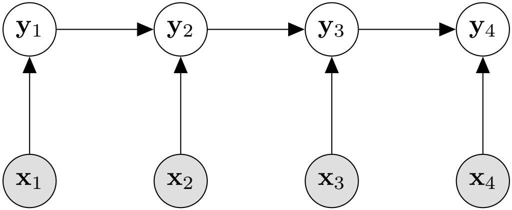






- PCC is a flexible version of a CRF (wrt loss function, base classifier, ...)

Across Labels *and* Time



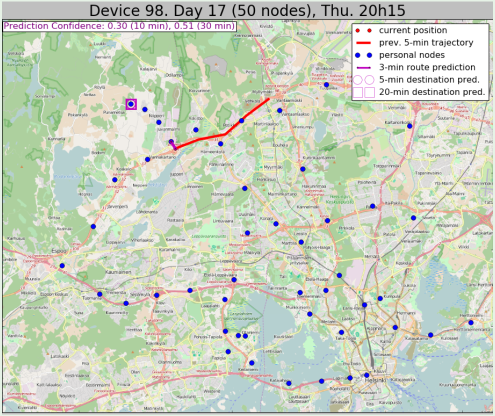
Each index is time, containing multiple labels:

$$\mathbf{y}_t = [y^{(1)}, \ldots, y^{(L)}]$$

for L labels and time $t = 1, \ldots, T$.

Across Labels *and* Time

Route Estimation/Prediction



- Predict in time $t = 1, \ldots$ for L travellers

Outline

Time-Series Data Mining

Tasks in 'time series' mining.

- query by content
- anomaly detection
- motif discovery
- prediction (forecasting)
- clustering (whole series)
- classification (whole series)
- segmentation (e.g., change point detection)

It is all about the 'shape' of the data.

Outline

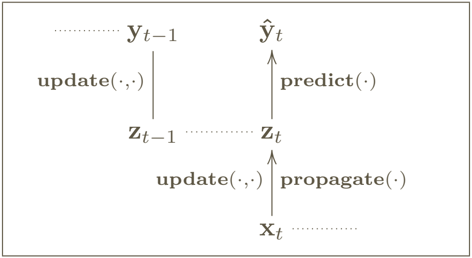
Unlabelled Instances

In many applications it is unrealistic to expect labels for every instance. What to do about this?

- Ignore instances with no label
- Use active learning to get good labels
- Use predicted labels (self-training)
- Use an unsupervised process for example clustering, latent-variable representations.

Unlabelled Instances

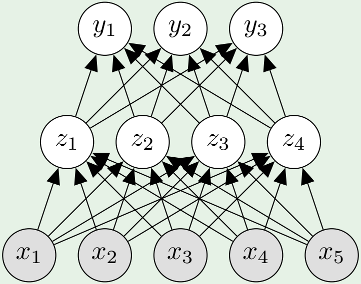
- Use an unsupervised process for example clustering, latent-variable representations.
  1. $\mathbf{z}_t = g(\mathbf{x}_t)$
  2. $\hat{\mathbf{y}}_t = h(\mathbf{z}_t)$
  3. update g with $(\mathbf{x}_t, \mathbf{z}_t)$
  4. update h with $(\mathbf{z}_{t-1}, \mathbf{y}_{t-1})$ (*if* y_{t-1} is available)



Unlabelled Instances

- Use an unsupervised process for example clustering, latent-variable representations.

Example



- z-variables are learned from input $\mathbf{x}_t$ only/primarily
- model $h : \mathcal{Z} \rightarrow Y$ updated when labels $\mathbf{y}_t$ available

Outline

Summary

- Multi-label classification can be adapted to the data-stream environment
- This context incurs particular challenges: modelling label dependence is important, but this is difficult in a dynamic environment (concept drift)
- Temporal dependence may exist in data streams: dependence exists across time
- Strong parallels exist between multi-label learning (dependence among labels) and sequence learning (dependence across time) problems
- Possible applications exist to time series learning

Concept Drift and Sequential Data

Connections between multi-label and time-series learning

Jesse Read

02 December, 2016

Outline

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

HAMMING LOSS

$$= \frac{1}{NL} \sum_{i=1}^{N} \sum_{i=1}^{L} \mathbb{I}[\hat{y}_j^{(i)} \neq y_j^{(i)}]$$

$$= 0.20$$

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

0/1 LOSS

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{\mathbf{y}}^{(i)} \neq \mathbf{y}^{(i)})$$

$$= 0.60$$

Often used as EXACT MATCH (1−0/1 LOSS)

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1 0 0 1]

Jaccard Index

$$
\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N} \frac{|\hat{\mathbf{y}}^{(i)} \wedge \mathbf{y}^{(i)}|}{|\hat{\mathbf{y}}^{(i)} \vee \mathbf{y}^{(i)}|} \\
&= \frac{1}{5}\left(\frac{1}{3} + 1 + 1 + \frac{1}{2} + \frac{1}{2}\right) \\
&= 0.67
\end{aligned}
$$

(Where $\vee$ and $\wedge$ are the logical OR and AND operations, applied vector-wise)

Multi-label Evaluation Metrics

We can evaluate posterior probabilities/confidences directly.

	$\mathbf{y}^{(i)}$	$[p(y_j\|\tilde{\mathbf{x}}^{(i)}])]_{j=1}^{L}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]

LOG LOSS – like HAMMING LOSS, to encourage good 'confidence',

- $y_j = 1, h_j(\tilde{\mathbf{x}}) = 0.4$ incurs loss of $-\log(0.4) = 0.92$
- $y_j = 1, h_j(\tilde{\mathbf{x}}) = 0.1$ incurs loss of $-\log(0.1) = 2.30$

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$[p(y_j \vert \tilde{\mathbf{x}}^{(i)}])]_{j=1}^{L}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]

RANKING LOSS – to encourage good ranking;
evaluates the average fraction of label pairs miss-ordered for $\tilde{\mathbf{x}}$:

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{(j,k):y_j > y_k} \left(\mathbb{I}[r_i(j) < r_i(k)] + \frac{1}{2} \mathbb{I}[r_i(j) = r_i(k)] \right)$$

where $r_i(j)$:= ranking of label j for instance $\tilde{\mathbf{x}}^{(i)}$

Multi-label Evaluation Metrics

	$\mathbf{y}^{(i)}$	$[p(y_j \mid \tilde{\mathbf{x}}^{(i)}])]_{j=1}^{L}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.1 0.2]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]

RANKING LOSS – to encourage good ranking;
evaluates the average fraction of label pairs miss-ordered for $\tilde{\mathbf{x}}$:

$$\frac{1}{5}\left(\frac{1}{4} + \frac{0}{4} + \frac{0}{4} + \frac{1.5}{4} + \frac{1}{4}\right)$$

Multi-label Evaluation Metrics

Other metrics used in the literature:

- ONE ERROR – if top ranked label is not in set of true labels
- COVERAGE – average "depth" to cover all true labels
- PRECISION
- RECALL
- macro-averaged F-MEASURE (ordinary averaging of a binary measure)
- micro-averaged F-MEASURE (labels as different instances of a 'global' label)
- PRECISION vs. RECALL curves

0/1 loss vs. Hamming loss

Example: 0/1 loss vs. Hamming loss

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[1 0 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

- Ham. Loss 0.3
- 0/1 Loss 0.6

0/1 loss vs. Hamming loss

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[1 0 **1** 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 **1** 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 1 **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[1 0 1 **0**]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

Optimizing HAM. LOSS …

- HAM. LOSS 0.2
- 0/1 LOSS 0.8

0/1 loss vs. Hamming loss

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[**0** **1** 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 **0** **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[**0** **1** 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

Optimizing 0/1 LOSS ...

- HAM. LOSS 0.4
- 0/1 LOSS 0.4

0/1 loss vs. Hamming loss

Example: 0/1 LOSS vs. HAMMING LOSS

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{y}}^{(i)}$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[**0** **1** 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[0 1 1 0]	[0 **0** **1** 0]
$\tilde{\mathbf{x}}^{(4)}$	[1 0 0 0]	[**0** **1** 1 1]
$\tilde{\mathbf{x}}^{(5)}$	[0 1 0 1]	[0 1 0 1]

- HAMMING LOSS minimized by binary relevance
- 0/1 LOSS minimized by chain and label powerset methods
- Cannot minimize both at the same time!

For general evaluation, ***use multiple and contrasting evaluation measures!***

Going from confidence to labels

Many methods output

- probabilistic information; or
- votes from an ensemble process

Example: Ensemble of 3 multi-label models

For some test instance $\tilde{\mathbf{x}}$ ...

	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
$\mathbf{h}^1(\tilde{\mathbf{x}})$	1	0	1	0
$\mathbf{h}^2(\tilde{\mathbf{x}})$	0	1	1	0
$\mathbf{h}^3(\tilde{\mathbf{x}})$	1	0	1	0
$p(\mathbf{y}_j \mid \mathbf{x}) \approx \frac{1}{3}\sum_{m=1}^{3} y_j^m$	0.67	0.33	1.00	0.00
$\hat{\mathbf{y}} \in \{0,1\}^3$	?	?	?	?

We may want to evaluate these directly (e.g., LOG LOSS); but we usually need to convert them to binary values ($\hat{\mathbf{y}}$).

Threshold Selection

Use a threshold of 0.5 ?

$$\hat{y}_j = \begin{cases} 1, & \hat{p}_j(\tilde{\mathbf{x}}) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\mid\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\mid\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

...but would eliminate two errors with a threshold of 0.4 !

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Use *ad-hoc* threshold, e.g., 0.5
  - how to know which threshold to use?

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\|\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\|\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Select a threshold from an internal validation test, e.g., $\in \{0.1, 0.2, \ldots, 0.9\}$
  - slow

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\|\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\|\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Calibrate a threshold such that $\text{LCard}(\mathbf{Y}) \approx \text{LCard}(\hat{\mathbf{Y}})$
  - e.g., *training data* has label cardinality of 1.7;
  - set a threshold t such that the label cardinality of the *test* data is as close as possible to 1.7

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\mid\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\mid\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Calibrate L thresholds such that each $\text{LCARD}(\mathbf{Y}_j) \approx \text{LCARD}(\hat{\mathbf{Y}}_j)$
  - e.g., the frequency of label $y_j = 1$ is 0.3,
  - set a threshold t_j such that $\hat{y}_j = 1$ with frequency as close as possible to 0.3

Threshold Selection

Example with threshold of 0.5

	$\mathbf{y}^{(i)}$	$\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)})$	$\hat{\mathbf{y}}^{(i)} := \mathbb{I}[\hat{\mathbf{p}}(\mathbf{y}\vert\tilde{\mathbf{x}}^{(i)}) \geq 0.5]$
$\tilde{\mathbf{x}}^{(1)}$	[1 0 1 0]	[0.9 0.0 0.4 0.6]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(2)}$	[0 1 0 1]	[0.1 0.8 0.0 0.8]	[0 1 0 1]
$\tilde{\mathbf{x}}^{(3)}$	[1 0 0 1]	[0.8 0.0 0.1 0.7]	[1 0 0 1]
$\tilde{\mathbf{x}}^{(4)}$	[0 1 1 0]	[0.1 0.7 0.4 0.2]	[0 1 0 0]
$\tilde{\mathbf{x}}^{(5)}$	[1 0 0 0]	[1.0 0.0 0.0 1.0]	[1 0 0 1]

Possible thresholding strategies:

- Calibrate L thresholds such that each $\text{LCARD}(\mathbf{Y}_j) \approx \text{LCARD}(\hat{\mathbf{Y}}_j)$
  - e.g., the frequency of label $y_j = 1$ is 0.3,
  - set a threshold t_j such that $\hat{y}_j = 1$ with frequency as close as possible to 0.3

...but be careful not to overfit!