Probabilistic Label Trees in Vowpal Wabbit

Kalina Jasinska-Kobus[1] Marek Wydmuch[1] Mikhail Kuznetsov[2]
Róbert Busa-Fekete[3] Krzysztof Dembczyński[1,2]

[1] *Institute of Computing Science, Poznan University of Technology, Poland*
[2] *Yahoo! Research, New York, USA*
[3] *Google Research, New York, USA*

The NeurIPS 2020 Vowpal Wabbit Workshop

Agenda

1. Extreme multi-label classification (XMLC)
2. Formal setting
3. Probabilistic label trees (PLTs)
4. PLT in Vowpal Wabbit

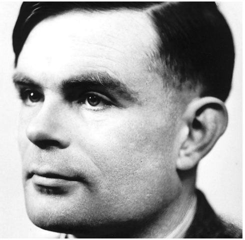
Agenda

1. Extreme multi-label classification (XMLC)
2. Formal setting
3. Probabilistic label trees (PLTs)
4. PLT in Vowpal Wabbit

Extreme multi-label classification (XMLC) is a problem of **labeling** an item with a **small** set of labels out of an **extremely large** number of potential labels.

Document tagging

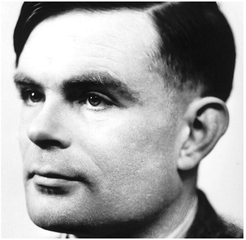
WIKIPEDIA
The Free Encyclopedia



Alan Turing

Document tagging

Wikipedia
The Free Encyclopedia

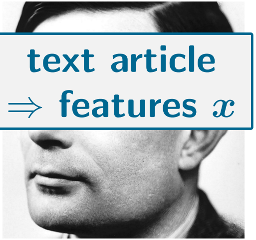


Alan Turing

Alan Turing | 1912 births | 1954 deaths
20th-century mathematicians | 20th-century philosophers
Academics of the University of Manchester Institute of Science and Technology
Alumni of King's College | Cambridge Artificial intelligence researchers
Atheist philosophers | Bayesian statisticians | British cryptographers | British logicians
British anti-fascists | British long-distance runners | British people of World War II
Computability theorists | Computer designers | English atheists
English computer scientists | English inventors | English logicians
English long-distance runners | English mathematicians
English people of Scottish descent | English philosophers | Former Protestants
Fellows of the Royal Society | Gay men | Gay academics | GCHQ people
Government Communications Headquarters people | History of artificial intelligence
Inventors who committed suicide | Male long-distance runners
Mathematicians who committed suicide | Officers of the Order of the British Empire
Bletchley Park people | People educated at Sherborne School
People from Maida Vale | People from Wilmslow
People prosecuted under anti-homosexuality laws | Princeton University alumni
Programmers who committed suicide | People who have received posthumous pardons
Recipients of British royal pardons | Academics of the University of Manchester
Suicides by cyanide poisoning | Suicides in England | Theoretical computer scientists

Document tagging

WIKIPEDIA
The Free Encyclopedia



text article
$\Rightarrow$ features x

Alan Turing

Alan Turing | 1912 births | 1954 deaths
20th-century mathematicians | 20th-century philosophers
Academics of the University of Manchester Institute of Science and Technology
Alumni of King's College | Cambridge Artificial intelligence researchers
Atheist philosophers | Bayesian statisticians | British cryptographers | British logicians
British anti-fascists | British long-distance runners | British people of World War II
Computability theorists | Computer designers | English atheists
English comp … cians
English long- …

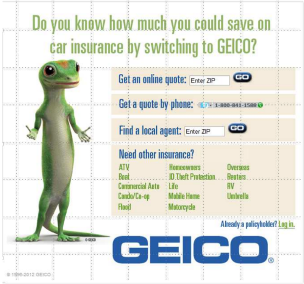
categories $\Rightarrow$ labels y

English people of Scottish descent | English philosophers | Former Protestants
Fellows of the Royal Society | Gay men | Gay academics | GCHQ people
Government Communications Headquarters people | History of artificial intelligence
Inventors who committed suicide | Male long-distance runners
Mathematicians who committed suicide | Officers of the Order of the British Empire
Bletchley Park people | People educated at Sherborne School
People from Maida Vale | People from Wilmslow
People prosecuted under anti-homosexuality laws | Princeton University alumni
Programmers who committed suicide | People who have received posthumous pardons
Recipients of British royal pardons | Academics of the University of Manchester
Suicides by cyanide poisoning | Suicides in England | Theoretical computer scientists

Information retrieval for search and advertising

Tagging ads with relevant queries:[1]

Ad:

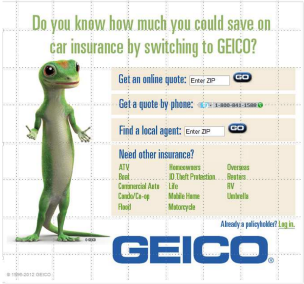


[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, 2013

Information retrieval for search and advertising

Tagging ads with relevant queries:[1]

Ad:



Queries:

"geico car insurance"
"geico auto insurance"
"geico insurance"
"www geico com"
"care geicos"
"geico com"
"need cheep auto insurance"
"wisconsin car insurence quotes"
"cheap auto insurence florida"
"all state auto insurence coupon code"

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, 2013

Information retrieval for search and advertising

Tagging ads with relevant queries:[1]

Ad:

ad $\Rightarrow$ features x

Queries:

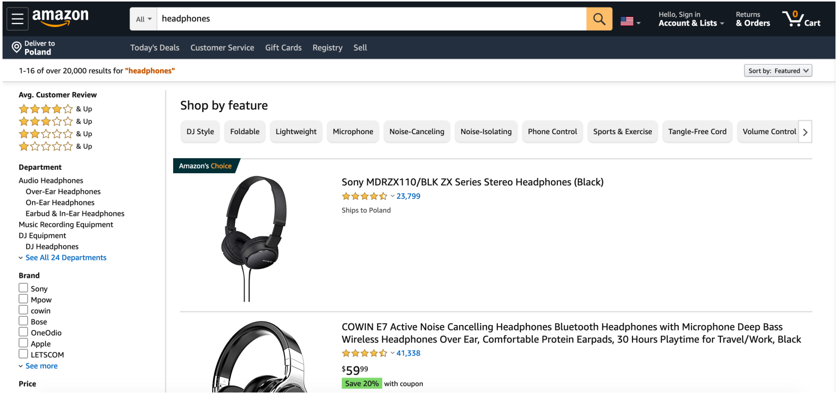
```
"geico car insurance"
"geico auto insurance"
"geico insurance"
"www geico com"
"care geicos"
"geico com"
"need cheep auto insurance"
"wisconsin car insurence quotes"
"cheap auto insurence florida"
"all state auto insurence coupon code"
```

queries $\Rightarrow$ labels y

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, 2013

Information retrieval for search and advertising

Predicting relevant products for queries:[2]

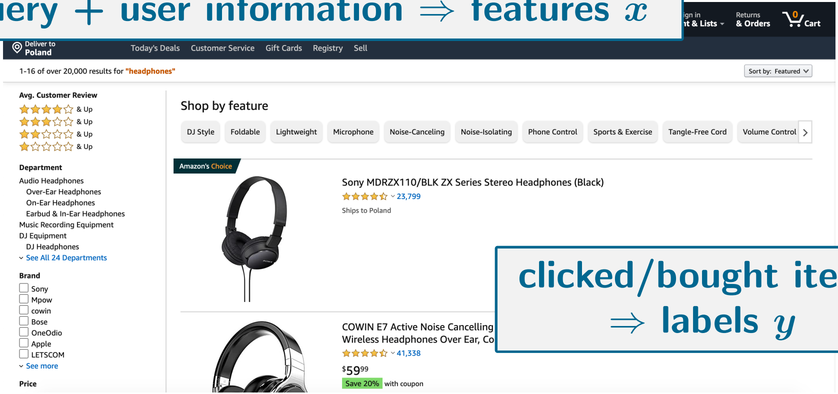


[2] Tharun Kumar Reddy Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, and Anshumali Shrivastava. Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products. In *NeurIPS*, 2019

Information retrieval for search and advertising

Predicting relevant products for queries:[2]

query + user information $\Rightarrow$ features x



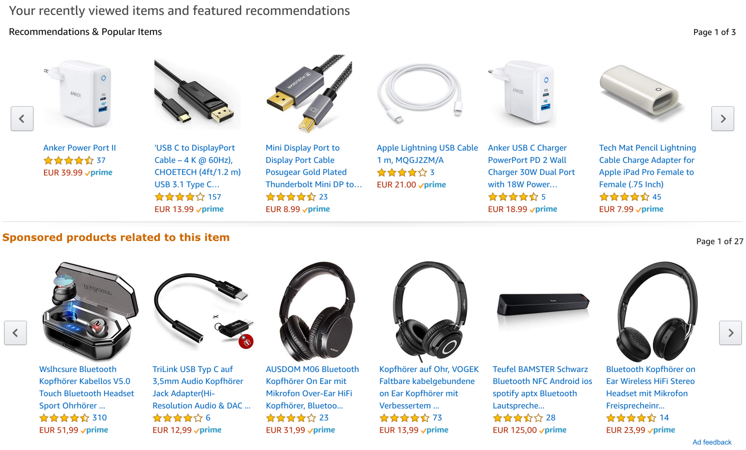
clicked/bought items $\Rightarrow$ labels y

---

[2] Tharun Kumar Reddy Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, and Anshumali Shrivastava. Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products. In *NeurIPS*, 2019

Recommendation systems

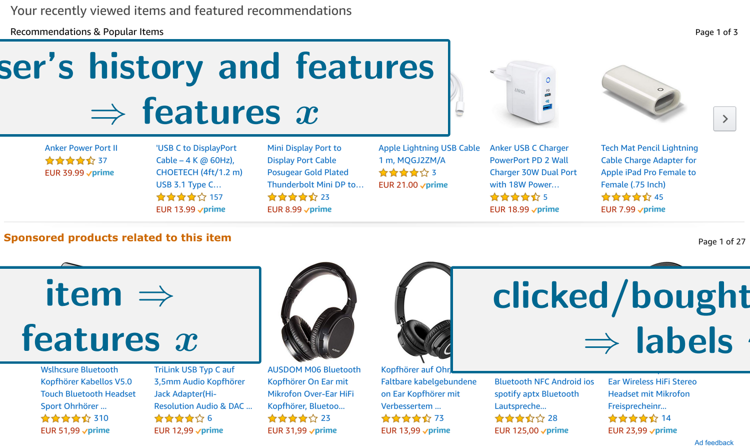
User-to-item and item-to-item recommendations:[3]



[3] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *KDD*, 2018

Recommendation systems

User-to-item and item-to-item recommendations:[3]



user's history and features $\Rightarrow$ features x

item $\Rightarrow$ features x

clicked/bought items $\Rightarrow$ labels y

[3] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *KDD*, 2018

Agenda

1. Extreme multi-label classification (XMLC)
2. Formal setting
3. Probabilistic label trees (PLTs)
4. PLT in Vowpal Wabbit

Formal setting

- **Multi-label classification**:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \{0,1\}^m$$

x_1	x_2	$\ldots$	x_d	y_1	y_2	$\ldots$	y_m
4.0	2.5		-1.5	1	1		0

Formal setting

- **Multi-label classification**:

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d \xrightarrow{\boldsymbol{h}(\boldsymbol{x})} \boldsymbol{y} = (y_1, y_2, \ldots, y_m) \in \{0,1\}^m$$

x_1	x_2	$\ldots$	x_d	y_1	y_2	$\ldots$	y_m
4.0	2.5		-1.5	1	1		0

- **Goal**: find a classifier $\boldsymbol{h}(\boldsymbol{x}) : \mathcal{X} \rightarrow \mathcal{R}^m$ with small **expected loss**:

$$L_\ell(\boldsymbol{h}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P(\boldsymbol{x},\boldsymbol{y})}(\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})))$$

Formal setting

- The **regret** of a classifier $\boldsymbol{h}$ with respect to ℓ

$$\mathrm{reg}_\ell(\boldsymbol{h}) = L_\ell(\boldsymbol{h}) - L_\ell(\boldsymbol{h}^*_\ell) = L_\ell(\boldsymbol{h}) - L^*_\ell$$

  quantifies the suboptimality of $\boldsymbol{h}$ compared to the optimal (**Bayes**) classifier:

$$\boldsymbol{h}^*_\ell = \operatorname*{arg\,min}_{\boldsymbol{h}} L_\ell(\boldsymbol{h})$$

Conditional marginal probability

- **Conditional marginal probability** of a label:

$$\eta_j(\boldsymbol{x}) = P(y_j = 1|\boldsymbol{x}) = \sum_{\boldsymbol{y}:y_j=1} P(\boldsymbol{y}|\boldsymbol{x})$$

Conditional marginal probability

- **Conditional marginal probability** of a label:

$$\eta_j(\boldsymbol{x}) = P(y_j = 1|\boldsymbol{x}) = \sum_{\boldsymbol{y}:y_j=1} P(\boldsymbol{y}|\boldsymbol{x})$$

- Bayes classifiers for popular MLC losses, such as:
  - Hamming loss,
  - micro and macro-F1 measure,
  - precision@k,
  - DCG@k.

  are directly expressed via conditional label probabilities.

Conditional marginal probability

- **Conditional marginal probability** of a label:

$$\eta_j(\boldsymbol{x}) = P(y_j = 1|\boldsymbol{x}) = \sum_{\boldsymbol{y}:y_j=1} P(\boldsymbol{y}|\boldsymbol{x})$$

- Bayes classifiers for popular MLC losses, such as:
  - Hamming loss,
  - micro and macro-F1 measure,
  - precision@k,
  - DCG@k.

  are directly expressed via conditional label probabilities.
- Hence accurate estimation of $\eta_j(\boldsymbol{x})$ is crucial for solving XMLC problems.

MLC under hamming loss

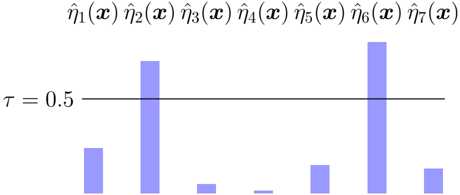
- **Hamming loss**:

$$\ell_H(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{1}{m} \sum_{j=1}^{m} [\![y_j \neq h_j(\boldsymbol{x})]\!]$$

- Sparse labels $\Rightarrow$ Hamming loss of an **all-zero** classifier close to **0**

- **The optimal strategy**:[4]

$$h_j^*(\boldsymbol{x}) = [\![\eta_j(\boldsymbol{x}) > 0.5]\!]$$



[4] Krzysztof Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88, 2012

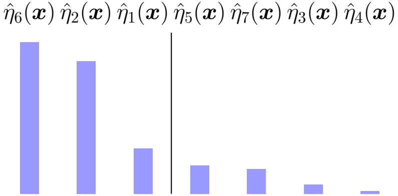
MLC under precision@k

- **Precision at position k (p@k)**:

$$\mathrm{p}@k(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}) = \frac{1}{k} \sum_{j \in \hat{\mathcal{Y}}_k} [\![y_j = 1]\!] \,,$$

where $\hat{\mathcal{Y}}_k$ is a set of k labels predicted by $\boldsymbol{h}$.

- **The optimal strategy**:[5]
select top k labels according to $\eta_j(\boldsymbol{x})$.



[5] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *NeurIPS*, 2018

Computational challenge

$\boxed{\textbf{Extreme classification} \Rightarrow \text{a } \textbf{large} \text{ number of } \textbf{labels } m\ (\geq 10^5)}$

$\Rightarrow$ a **large** number of **features** d $(\geq 10^6)$

$\Rightarrow$ a **large** number of **examples** n $(\geq 10^6)$

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**
  - High performance computing resources available

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**
  - High performance computing resources available
  - Large data $\longrightarrow$ sparse data (sparse features and labels)

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**
  - High performance computing resources available
  - Large data $\longrightarrow$ sparse data (sparse features and labels)
  - Vowpal Wabbit – library for fast online learning exists

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**
  - High performance computing resources available
  - Large data $\longrightarrow$ sparse data (sparse features and labels)
  - Vowpal Wabbit – library for fast online learning exists
  - Learning time for binary model and $n = 10^6$: $\sim 2 - 3s$

Computational challenge

- Naive approach: One-vs-All with linear models, $\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x}$

Problem size:		**Complexity:**
$n \geq 10^6$, $d \geq 10^6$, $m \geq 10^5$	$\Rightarrow$	training time $\geq 10^{17}$ space $\geq 10^{11}$ predict time $\geq 10^{11}$

- **Is it really that hard?**
  - High performance computing resources available
  - Large data $\longrightarrow$ sparse data (sparse features and labels)
  - Vowpal Wabbit – library for fast online learning exists
  - Learning time for binary model and $n = 10^6$: $\sim 2 - 3s$
  - Learning time $m = 10^5$: $10^5 \times 2s > 2$ day – **not great, not terrible**

Agenda

1. Extreme multi-label classification (XMLC)
2. Formal setting
3. Probabilistic label trees (PLTs)
4. PLT in Vowpal Wabbit

Probabilistic label trees (PLTs)[6]

- **PLT**s follows the learning reductions framework: the original problem is decomposed to a **set of binary problems** organized in a tree structure.

[6] Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Probabilistic label trees (PLTs)[6]

- **PLT**s follows the learning reductions framework: the original problem is decomposed to a **set of binary problems** organized in a tree structure.
- Path from the root to a leaf corresponds to one and only one label.

[6] Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

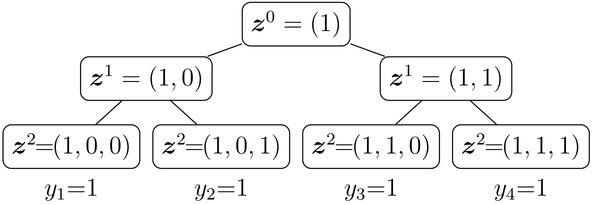
Probabilistic label trees (PLTs)[6]

- **PLT**s follows the learning reductions framework: the original problem is decomposed to a **set of binary problems** organized in a tree structure.
- Path from the root to a leaf corresponds to one and only one label.
- Each label/path is **coded** by vector $\boldsymbol{z} = (1, z_1, \ldots, z_l) \in \mathcal{C}$.

[6] Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Probabilistic label trees (PLTs)[6]

- **PLT**s follows the learning reductions framework: the original problem is decomposed to a **set of binary problems** organized in a tree structure.
- Path from the root to a leaf corresponds to one and only one label.
- Each label/path is **coded** by vector $\boldsymbol{z} = (1, z_1, \dots, z_l) \in \mathcal{C}$.
- An internal node is identified by a **partial** code $\boldsymbol{z}^j = (z_1, \dots, z_j)$.
- The code does **not** have to be binary.



6 Kalina Jasinska, Krzysztof Dembczynski, Róbert Busa-Fekete, Karlson Pfannschmidt, Timo Klerx, and Eyke Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Probabilistic label trees (PLTs)

- **Factorization** of the conditional marginal probability:

$$\eta_j(\boldsymbol{x}) = P(\boldsymbol{z} \,|\, \boldsymbol{x}) = \prod_{i=0}^{l} P(z_i \,|\, \boldsymbol{z}^{i-1}, \boldsymbol{x})\,.$$

$$P(z_0 = 1 \,|\, \boldsymbol{x}) = 1$$

$$P(z_1 = 0 \,|\, \boldsymbol{z}^0 = (1), \boldsymbol{x}) = 0.5 \qquad P(z_1 = 1 \,|\, \boldsymbol{z}^0 = (1), \boldsymbol{x}) = 0.6$$

$$P(z_2{=}0 \,|\, \boldsymbol{z}^1{=}(1,0), \boldsymbol{x}){=}1.0 \qquad P(z_2{=}1 \,|\, \boldsymbol{z}^1{=}(1,0), \boldsymbol{x}){=}0.1 \qquad P(z_2{=}0 \,|\, \boldsymbol{z}^1{=}(1,1), \boldsymbol{x}){=}0.5 \qquad P(z_2{=}1 \,|\, \boldsymbol{z}^1{=}(1,1), \boldsymbol{x}){=}0.7$$

$$P(\boldsymbol{z}{=}(1,0,0) \,|\, \boldsymbol{x}){=}0.5 \qquad P(\boldsymbol{z}{=}(1,0,1) \,|\, x){=}0.05 \qquad P(\boldsymbol{z}{=}(1,1,0) \,|\, \boldsymbol{x}){=}0.3 \qquad P(\boldsymbol{z}{=}(1,1,1) \,|\, \boldsymbol{x}){=}0.42$$

Probabilistic label trees (PLTs)

- Efficient learning, requires updating at most $||\boldsymbol{y}||_1$ paths per training example.

Probabilistic label trees (PLTs)

- Efficient learning, requires updating at most $||\boldsymbol{y}||_1$ paths per training example.
- Different tree-search algorithms can by applied for prediction:

Probabilistic label trees (PLTs)

- Efficient learning, requires updating at most $||\boldsymbol{y}||_1$ paths per training example.
- Different tree-search algorithms can by applied for prediction:
  - beam search (approx. top-k predition, $\log m$ complexity for balanced trees)

Probabilistic label trees (PLTs)

- Efficient learning, requires updating at most $||\boldsymbol{y}||_1$ paths per training example.
- Different tree-search algorithms can by applied for prediction:
  - beam search (approx. top-k predition, $\log m$ complexity for balanced trees)
  - uniform-cost search (exact top-k prediction, $\log m$ complexity under additional assumptions)

Probabilistic label trees (PLTs)

- Efficient learning, requires updating at most $||\boldsymbol{y}||_1$ paths per training example.
- Different tree-search algorithms can by applied for prediction:
  - beam search (approx. top-k predition, $\log m$ complexity for balanced trees)
  - uniform-cost search (exact top-k prediction, $\log m$ complexity under additional assumptions)
  - threshold-based search (threshold-based prediction, $\log m$ complexity under additional assumptions)

PLTs and other label tree approaches

- **PLT**s can be treated as multi-label generalization of:
  - Hierarchical softmax[7]
  - Conditional probabilistic estimation trees[8]
  - Nested dichotomies[9]
- Other (non-probabilistic) label tree approaches:
  - Label embedding trees[10]
  - Filter tree[11]
  - HOMER[12]

---

[7] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

[8] Alina Beygelzimer, John Langford, Yury Lifshits, Gregory B. Sorkin, and Alexander L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, 2009

[9] John Fox. *Applied regression analysis, linear models, and related methods*. Sage, 1997

[10] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, 2010

[11] Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error-correcting tournaments. In *ALT*, 2009

[12] Grigorios Tsoumakas and Ioannis Katakis I Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, 2008

PLTs and other label tree approaches

- The **PLT** model has been used in several XMLC packages:
  - Parabel[13]
  - extremeText[14]
  - Bonsai[15]
  - AttentionXML[16]
  - napkinXC[17]
- **It is now also available in Vowpal Wabbit!**

---

[13] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *WWW*, 2018

[14] Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *NeurIPS*, 2018

[15] Sujay Khandagale, Han Xiao, and Rohit Babbar. Bonsai - diverse and shallow trees for extreme multi-label classification, 2019

[16] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *NeurIPS*. 2019

[17] Kalina Jasinska-Kobus, Marek Wydmuch, Krzysztof Dembczynski, Mikhail Kuznetsov, and Robert Busa-Fekete. Probabilistic label trees for extreme multi-label classification, 2020

Theoretical guarantees

- **Theorem**:[18] For any distribution P and internal node classifiers $f_{\boldsymbol{z}^i}$ the following holds:

$$|\eta_j(\boldsymbol{x}) - \hat{\eta}_j(\boldsymbol{x})| \leq \sum_{i=0}^{l} \sqrt{\frac{2}{\lambda}} \sqrt{\mathrm{reg}_\ell(f_{\boldsymbol{z}^i} \mid \boldsymbol{x})} \,,$$

  where $\mathrm{reg}_\ell(f_{\boldsymbol{z}^i} \mid \boldsymbol{x})$ is binary classification regret for a strongly proper composite loss ℓ (e.g., logistic loss) and λ is a constant specific for loss ℓ.

[18] Kalina Jasinska-Kobus, Marek Wydmuch, Krzysztof Dembczynski, Mikhail Kuznetsov, and Robert Busa-Fekete. Probabilistic label trees for extreme multi-label classification, 2020

Theoretical guarantees

- **Theorem**:[18] For any distribution P and internal node classifiers $f_{\boldsymbol{z}^i}$ the following holds:

$$|\eta_j(\boldsymbol{x}) - \hat{\eta}_j(\boldsymbol{x})| \leq \sum_{i=0}^{l} \sqrt{\frac{2}{\lambda}} \sqrt{\mathrm{reg}_\ell(f_{\boldsymbol{z}^i} \,|\, \boldsymbol{x})}\,,$$

  where $\mathrm{reg}_\ell(f_{\boldsymbol{z}^i} \,|\, \boldsymbol{x})$ is binary classification regret for a strongly proper composite loss ℓ (e.g., logistic loss) and λ is a constant specific for loss ℓ.
- This theorem leads to guarantees for such metrics as Hamming loss, generalized performance metrics, and precision@k.[18]

[18] Kalina Jasinska-Kobus, Marek Wydmuch, Krzysztof Dembczynski, Mikhail Kuznetsov, and Robert Busa-Fekete. Probabilistic label trees for extreme multi-label classification, 2020

Agenda

1. Extreme multi-label classification (XMLC)
2. Formal setting
3. Probabilistic label trees (PLTs)
4. PLT in Vowpal Wabbit

PLT in Vowpal Wabbit

```
$ vw <dataset> --plt <m> --kary_tree <k>
```

- The option `--plt <m>`, where `<m>` is the number of distinct labels, directs `vw` to perform multi-label classification using a PLT.

PLT in Vowpal Wabbit

```
$ vw <dataset> --plt <m> --kary_tree <k>
```

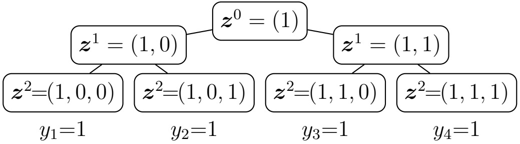
- The option `--plt <m>`, where `<m>` is the number of distinct labels, directs `vw` to perform multi-label classification using a PLT.
- Labels must be natural numbers from set $\{1, \ldots,$ `<m>`$\}$, with `<m>` being the maximum label value.

PLT in Vowpal Wabbit

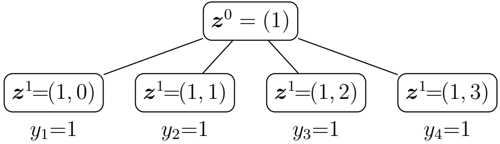
```
$ vw <dataset> --plt <m> --kary_tree <k>
```

- The option `--plt <m>`, where `<m>` is the number of distinct labels, directs `vw` to perform multi-label classification using a PLT.
- Labels must be natural numbers from set $\{1, \ldots,$ `<m>`$\}$, with `<m>` being the maximum label value.
- The option `--kary_tree <k>` controls tree node arity.

`--plt 4 --kary_tree 2`



`--plt 4 --kary_tree 4`



PLT in Vowpal Wabbit

```
$ vw <dataset> -i <plt_model> --threshold <thr>/--top_k <k>
```

- The `--threshold <thr>` option indicates the use of the threshold-based prediction, i.e., labels with $\hat{\eta}_j(\boldsymbol{x})$ above threshold `<thr>` are predicted.
- The `--top_k <k>` option indicates the use of the top-k prediction instead.

Demo[20]

	d (dim $\mathcal{X}$)	m (dim $\mathcal{Y}$)	n_{train}	n_{test}	avg. $\|\|\boldsymbol{y}\|\|_1$
AmazonCat-13K[19]	203882	13330	1186239	306782	5.04

```
$ wget http://www.cs.put.poznan.pl/mwydmuch/data/amazonCat_train.vw
```

```
$ wget http://www.cs.put.poznan.pl/mwydmuch/data/amazonCat_test.vw
```

[19] K. Bhatia, K. Dahiya, H. Jain, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016

[20] https://github.com/VowpalWabbit/vowpal_wabbit/tree/master/demo/plt

Demo

```
$ time vw amazonCat_train.vw -c --multilabel_oaa 13330 \
  --passes 3 -b 31 -f ovr_weights --holdout_off ⇒ 339.11 mins
```

```
$ time vw amazonCat_test.vw -i ovr_weights ⇒ 23.82 mins
```

Demo

```
$ time vw amazonCat_train.vw -c --multilabel_oaa 13330 \
  --passes 3 -b 31 -f ovr_weights --holdout_off ⇒ 339.11 mins
```

```
$ time vw amazonCat_test.vw -i ovr_weights ⇒ 23.82 mins
```

```
$ time vw amazonCat_train.vw -c --plt 13330 --kary_tree 16 \
  --passes 3 -b 31 -f plt_weights --holdout_off ⇒ 24.07 mins
```

```
$ time vw amazonCat_test.vw -i plt_weights --top_k 5 ⇒ 38.24 secs
```

Demo

```
$ time vw amazonCat_train.vw -c --multilabel_oaa 13330 \
  --passes 3 -b 31 -f ovr_weights --holdout_off ⇒ 339.11 mins
```

```
$ time vw amazonCat_test.vw -i ovr_weights ⇒ 23.82 mins
```

```
$ time vw amazonCat_train.vw -c --plt 13330 --kary_tree 16 \
  --passes 3 -b 31 -f plt_weights --holdout_off ⇒ 24.07 mins
```

```
$ time vw amazonCat_test.vw -i plt_weights --top_k 5 ⇒ 38.24 secs
```

	$p@1$	$p@3$	$p@5$
`vw --multilabel_oaa`	91.11	76.40	61.41
`vw --plt`	91.46	75.01	59.68

Thank you for your attention

Try:

```
$ vw <dataset> --plt <m>
```

Read more:

Probabilistic Label Trees for Extreme Multi-label Classification

https://arxiv.org/pdf/2009.11218.pdf