Label tree structure learning in extreme multi-label classification

Kalina Jasinska   Marek Wydmuch   Krzysztof Dembczyński

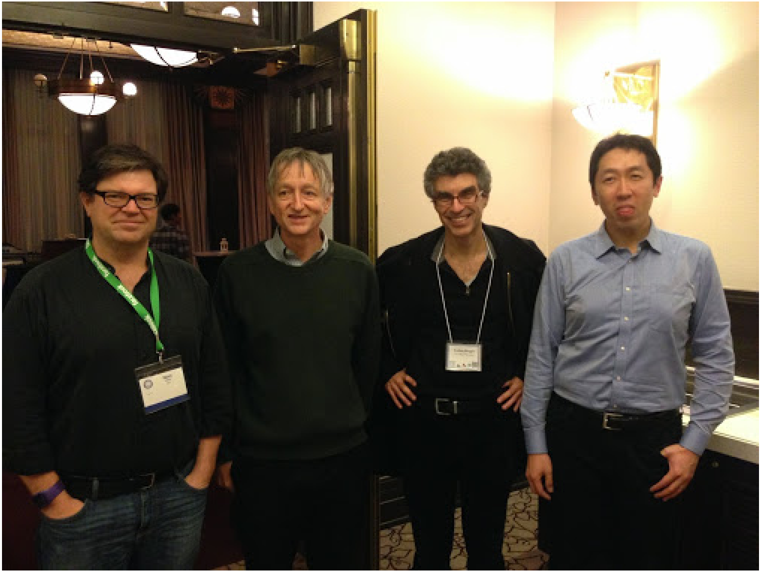
Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

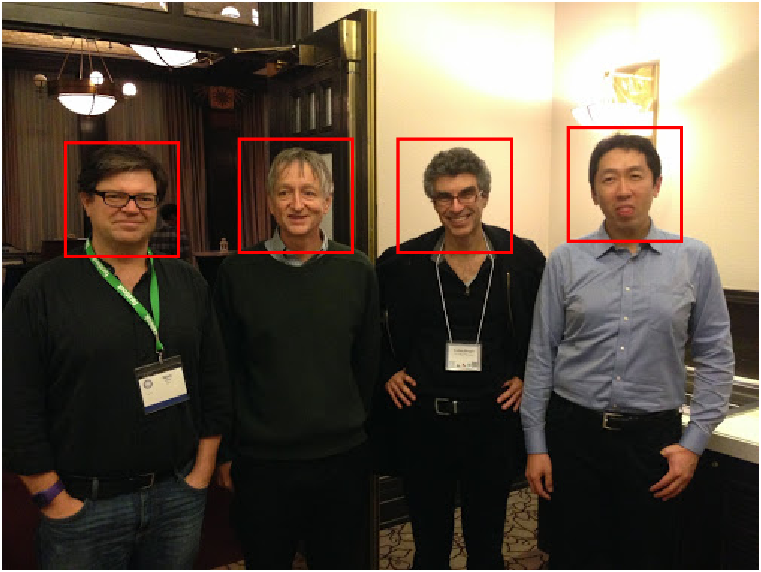
Poznań University of Technology, IDSS Seminar, December 19, 2017

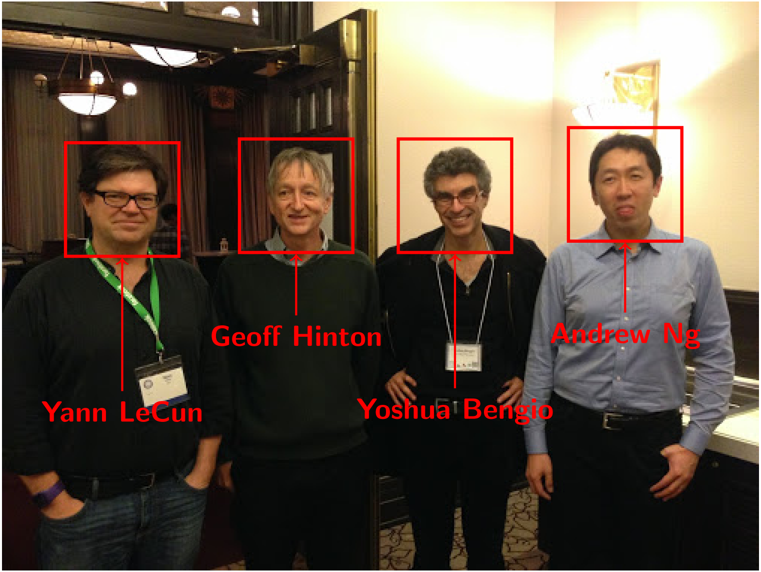
Outline

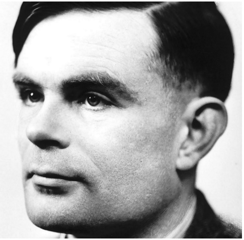
1. Extreme multi-label classification
2. Probabilistic label trees (PLT)
3. Online PLT
4. FastPLT: Greedy batch training
5. Summary

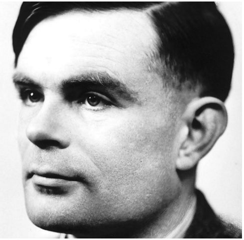
Outline






Yann LeCun
Geoff Hinton
Yoshua Bengio
Andrew Ng





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 long-distance runners, British male athletes, 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
Government Communications Headquarters people, History of artificial intelligence
Inventors who committed suicide, LGBT scientists
LGBT scientists from the United Kingdom, Male long-distance runners
Mathematicians who committed suicide, Officers of the Order of the British Empire
People associated with Bletchley Park, People educated at Sherborne School
People from Maida Vale, People from Wilmslow
People prosecuted under anti-homosexuality laws, Philosophers of mind
Philosophers who committed suicide, Princeton University alumni, 1930-39
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

Setting

- **Multi-class classification**:

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

	x_1	x_2	$\ldots$	x_d	y
$\boldsymbol{x}$	4.0	2.5		-1.5	5

- **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
$\boldsymbol{x}$	4.0	2.5		-1.5	1	1		0

Setting

- **Multi-class classification**:

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

The problem can be expressed as estimation of the distribution:

$$P(y \,|\, \boldsymbol{x}) \;\; \text{such that} \;\; \sum_y P(y \,|\, \boldsymbol{x}) = 1$$

- **Multi-label classification**:

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

The problem can be expressed as estimation of the distribution:

$$P(y_j = 1 \,|\, \boldsymbol{x}) \;\; \text{such that} \sum_{z \in \{0,1\}} P(y_j = z \,|\, \boldsymbol{x}) = 1, \quad j = 1, \ldots, m$$

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

Extreme classification

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

- **Predictive performance**:

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m $(\geq 10^5)$

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m $(\geq 10^5)$

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m $(\geq 10^5)$

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning
  - Long-tail label distributions and zero-shot learning

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m $(\geq 10^5)$

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning
  - Long-tail label distributions and zero-shot learning
- **Computational complexity**:

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning
  - Long-tail label distributions and zero-shot learning
- **Computational complexity**:
  - time vs. space

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning
  - Long-tail label distributions and zero-shot learning
- **Computational complexity**:
  - time vs. space
  - #examples vs. #features vs. #labels

Extreme classification

Extreme classification $\Rightarrow$ a **large** number of **labels** m ($\geq 10^5$)

- **Predictive performance**:
  - Performance measures: Hamming loss, prec@k, NDCG@k, Macro F
  - Learning theory for large m
  - Training and prediction under limited time and space budged
  - Learning with missing labels and positive-unlabeled learning
  - Long-tail label distributions and zero-shot learning
- **Computational complexity**:
  - time vs. space
  - #examples vs. #features vs. #labels
  - training vs. validation vs. prediction

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:
  - Past events: NIPS 2013, 2015, 2016 and ICML 2015,

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:
  - Past events: NIPS 2013, 2015, 2016 and ICML 2015,
  - NIPS 2017 Workshop (organizers: Manik Varma, Marius Kloft, and Krzysztof Dembczyński),

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:
  - Past events: NIPS 2013, 2015, 2016 and ICML 2015,
  - NIPS 2017 Workshop (organizers: Manik Varma, Marius Kloft, and Krzysztof Dembczyński),
  - WWW 2018 Workshop (organizers: Akshay Soni, Robert Busa-Fekete, Krzysztof Dembczyński, Aasish Pappu),

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:
  - Past events: NIPS 2013, 2015, 2016 and ICML 2015,
  - NIPS 2017 Workshop (organizers: Manik Varma, Marius Kloft, and Krzysztof Dembczyński),
  - WWW 2018 Workshop (organizers: Akshay Soni, Robert Busa-Fekete, Krzysztof Dembczyński, Aasish Pappu),
  - ECIR 2018 Tutorial (authors: Rohit Babbar, Krzysztof Dembczyski),

Extreme classification: Growing subfield of research

- Many papers published at main ML conference like **ICML** and **NIPS**.
- Workshops, seminars and tutorials:
  - Past events: NIPS 2013, 2015, 2016 and ICML 2015,
  - NIPS 2017 Workshop (organizers: Manik Varma, Marius Kloft, and Krzysztof Dembczyński),
  - WWW 2018 Workshop (organizers: Akshay Soni, Robert Busa-Fekete, Krzysztof Dembczyński, Aasish Pappu),
  - ECIR 2018 Tutorial (authors: Rohit Babbar, Krzysztof Dembczyski),
  - Dagstuhl Seminar 2018 (organizers: Manik Varma, Samy Bengio, Thorsten Joachims, Marius Kloft, Krzysztof Dembczyński).

- Smart 1-vs-all approaches,

Extreme classification: Algorithms

- Smart 1-vs-all approaches,
- Embeddings methods,

Extreme classification: Algorithms

- Smart 1-vs-all approaches,
- Embeddings methods,
- Label filtering,

Extreme classification: Algorithms

- Smart 1-vs-all approaches,
- Embeddings methods,
- Label filtering,
- Tree-based method: decision trees and **label trees**.

Outline

Decision trees vs. label trees

- Decision trees:

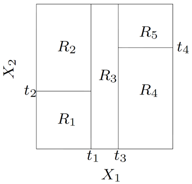
---

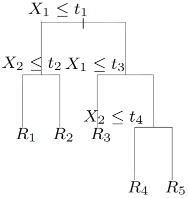
[1] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. In *NIPS 29*, 2015

[2] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

Decision trees vs. label trees

- Decision trees:
  - Partition of the feature space to small subregions:



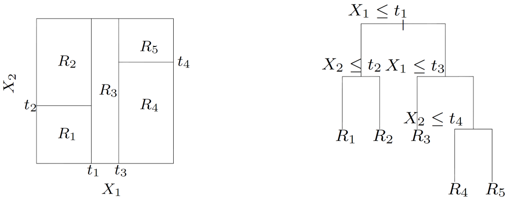


[1] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. In *NIPS 29*, 2015

[2] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

Decision trees vs. label trees

- Decision trees:
  - Partition of the feature space to small subregions:



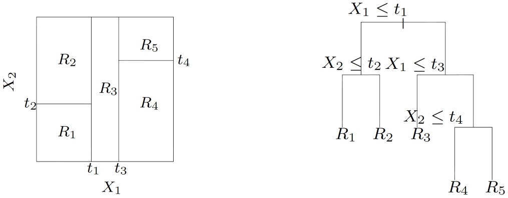
  - Fast prediction: logarithmic in n

[1] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. In *NIPS 29*, 2015

[2] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

Decision trees vs. label trees

- Decision trees:
  - Partition of the feature space to small subregions:



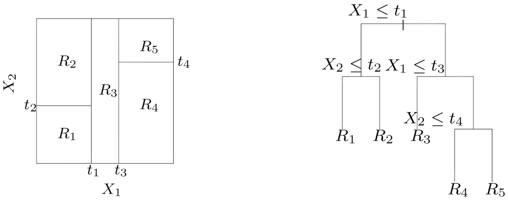
  - Fast prediction: logarithmic in n
  - Training can be expensive: computation of split criterion

[1] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. In *NIPS 29*, 2015

[2] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

Decision trees vs. label trees

- Decision trees:
  - Partition of the feature space to small subregions:



  - Fast prediction: logarithmic in n
  - Training can be expensive: computation of split criterion
  - Two new algorithms: LomTree[1] and FastXML[2]

[1] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. In *NIPS 29*, 2015

[2] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, pages 263–272. ACM, 2014

Decision trees vs. label trees

- Label trees:

[3] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[4] F Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

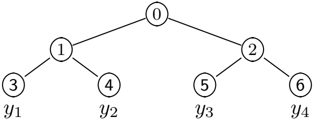
[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016

[7] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Decision trees vs. label trees

- Label trees:
  - Organize classifiers in a tree structure (one leaf ⇔ one label):



---

[3] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[4] F Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

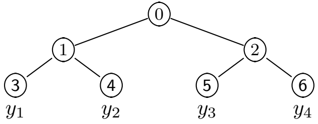
[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016

[7] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Decision trees vs. label trees

- Label trees:
  - Organize classifiers in a tree structure (one leaf $\Leftrightarrow$ one label):



  - Fast prediction: almost logarithmic in m

[3] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[4] F Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

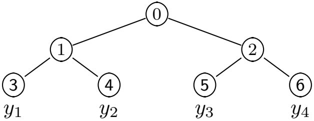
[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016

[7] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Decision trees vs. label trees

- Label trees:
  - Organize classifiers in a tree structure (one leaf $\Leftrightarrow$ one label):



  - Fast prediction: almost logarithmic in m
  - Different training and test procedures for multi-class and multi-label

---

[3] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[4] F Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

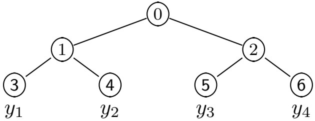
[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016

[7] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Decision trees vs. label trees

- Label trees:
  - Organize classifiers in a tree structure (one leaf $\Leftrightarrow$ one label):



  - Fast prediction: almost logarithmic in m
  - Different training and test procedures for multi-class and multi-label
  - Popular instances: Conditional probability trees[3], Hierarchical softmax[4], Label embedding trees [5], FastText[6], **Probabilistic label trees**[7]

---

[3] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[4] F Morin and Y. Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, 2005

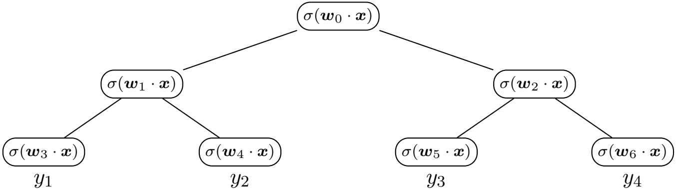
[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, pages 163–171. Curran Associates, Inc., 2010

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016

[7] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Probabilistic label trees (PLT)[8]

- PLT are based on b-ary **label trees**.

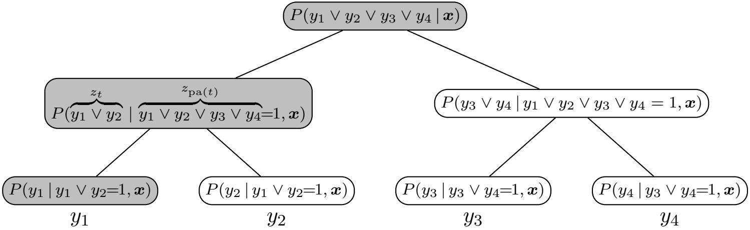


- **Probabilistic classifiers** in **all** nodes of the tree.
- **Internal** node classifier decides whether to **go down the tree**.
- A test example may follow **many paths** from the root to leaves.
- **Batch** and **online** learning possible.

[8] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *ICML*, 2016

Probabilistic label trees

- Class probability estimators in nodes for estimating $P(y_j = 1 \mid \boldsymbol{x})$.



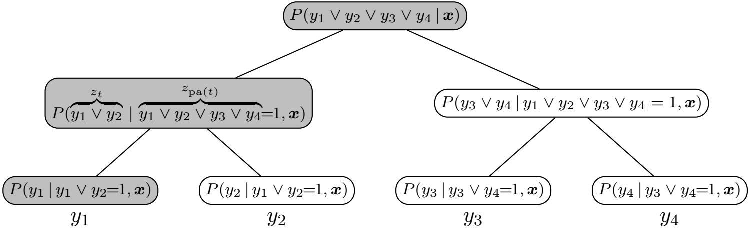
- Using the **chain rule** of probability

$$P(y_j = 1 \mid \boldsymbol{x}) = \eta_j(\boldsymbol{x}) = \prod_{t \in \mathrm{Path}(j)} \eta(\boldsymbol{x}, t)\,,$$

where $\eta(\boldsymbol{x}, t) = \begin{cases} P(z_t = 1 \mid \boldsymbol{x}) & \text{if } t \text{ is root,} \\ P(z_t = 1 \mid z_{\mathrm{pa}(t)} = 1, \boldsymbol{x}) & \text{otherwise.} \end{cases}$

Probabilistic label trees

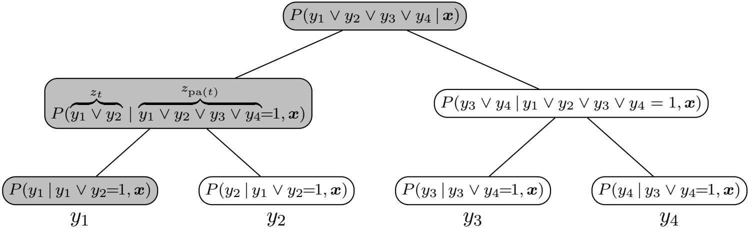
- Class probability estimators in nodes for estimating $P(y_j = 1 \mid \boldsymbol{x})$.



- Training: reduced complexity by the **conditions** used in the **nodes**.

Probabilistic label trees

- Class probability estimators in nodes for estimating $P(y_j = 1 \,|\, \boldsymbol{x})$.



- Training: reduced complexity by the **conditions** used in the **nodes**.
- Prediction: **priority queue search** or **branch and bound**.

PLT vs. HSM/CPET

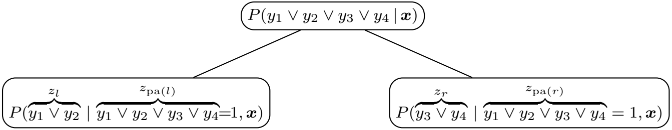
- Hierarchical softmax (HSM) and conditional probability estimation trees (CPET) are **only** for **multi-class** problems.

PLT vs. HSM/CPET

- Hierarchical softmax (HSM) and conditional probability estimation trees (CPET) are **only** for **multi-class** problems.
- FastText (also based on HSM) **randomly picks one of the labels** and treats the problem as multi-class.

PLT vs. HSM/CPET

- Hierarchical softmax (HSM) and conditional probability estimation trees (CPET) are **only** for **multi-class** problems.
- FastText (also based on HSM) **randomly picks one of the labels** and treats the problem as multi-class.
- PLT **generalizes** HSM: PLT trained on multi-class data gets the same model as HSM:



$$z_l = 1 - z_r \quad \text{and} \quad z_{\mathrm{pa}(l)} = z_{\mathrm{pa}(r)}$$

Experimental results

	#labels	#features	#test	#train	inst./lab.	lab./inst.
RCV1	2456	47236	155962	623847	1218.56	4.79
AmazonCat	13330	203882	306782	1186239	448.57	5.04
Wiki10	30938	101938	6616	14146	8.52	18.64
Delicious	205443	782585	100095	196606	72.29	75.54
WikiLSHTC	325056	1617899	587084	1778351	17.46	3.19
Amazon	670091	135909	153025	490449	3.99	5.45

Table: Datasets from the Extreme Classification repository.[9]

[9] http://manikvarma.org/downloads/XC/XMLRepository.html

Experimental results

	PLT			FastXML		
	P@1	P@3	P@5	P@1	P@3	P@5
RCV1	90.46	72.4	51.86	**91.13**	**73.35**	**52.67**
AmazonCat	91.47	75.84	61.02	**92.95**	**77.5**	**62.51**
Wiki10	**84.34**	**72.34**	**62.72**	81.71	66.67	56.70
Delicious	**45.37**	**38.94**	35.88	42.81	38.76	**36.34**
WikiLSHTC	45.67	29.13	21.95	**49.35**	**32.69**	**24.03**
Amazon	**36.65**	**32.12**	**28.85**	34.24	29.3	26.12

Tree-structure learning in label trees

- Clustering,
- Huffman trees,
- Online tree learning (CPET).

Tree-structure learning in PLT

- Till now we used random and Huffman trees.
- Two new ideas:
  - Online PLT,
  - Greedy Batch PLT.

Outline

Learning of Probabilistic Label Tree

- How to train a PLT with **no prior knowledge** of the label set?

Learning of Probabilistic Label Tree

- How to train a PLT with **no prior knowledge** of the label set in **fully online fashion**?

Online learning tree building

- To allow expansion of the tree structure, additional temporary classifiers t are maintained for certain classifiers h.

Online learning tree building

- To allow expansion of the tree structure, additional temporary classifiers t are maintained for certain classifiers h.
- When the algorithm observes an example (x, y) with a new unseen label, it uses tree expansion method.

Online learning tree building

- To allow expansion of the tree structure, additional temporary classifiers t are maintained for certain classifiers h.
- When the algorithm observes an example (x, y) with a new unseen label, it uses tree expansion method.
- Method ensures that proper conditional probabilities are learned by the estimators in the tree structure.

Online learning tree building methods

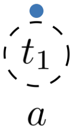
- Online tree structure learning in HSM/CPET[10],
- OnlinePLT with Leaf Expansion[11],
- OnlinePLT with Root Expansion.

[10] A. Beygelzimer, J. Langford, Y. Lifshits, G. B. Sorkin, and A. L. Strehl. Conditional probability tree estimation analysis and algorithms. In *UAI*, pages 51–58, 2009

[11] Kalina Jasinska and Krzysztof Dembczyński. Probabilistic label tree classifiers for extreme multi-label classification, 2016. Poster

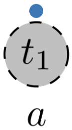
CPET – online label tree building

example: (x_1, a)



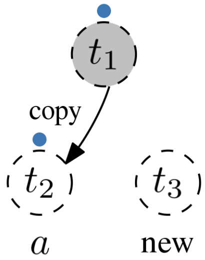
CPET – online label tree building

example: (x_2, b)



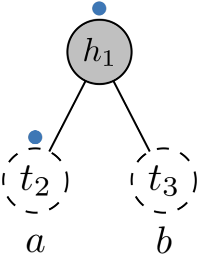
CPET – online label tree building

example: (x_2, b)



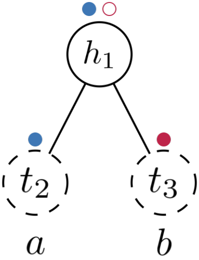
CPET – online label tree building

example: (x_2, b)



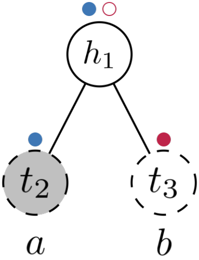
CPET – online label tree building

example: (x_2, b)



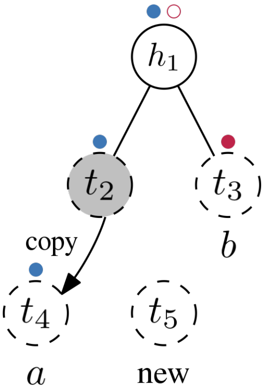
CPET – online label tree building

example: (x_3, c)



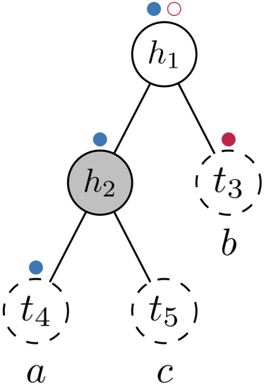
CPET – online label tree building

example: (x_3, c)



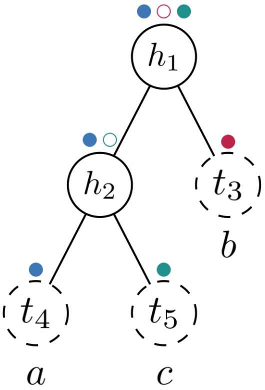
CPET – online label tree building

example: (x_3, c)



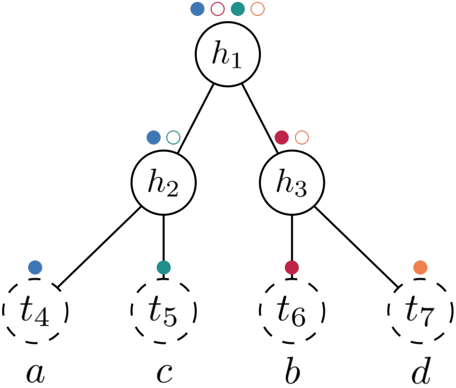
CPET – online label tree building

example: (x_3, c)



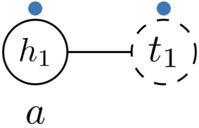
CPET – online label tree building

example: (x_4, d)



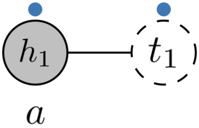
OnlinePLT with Leaf Expansion

example: (x_1, a)



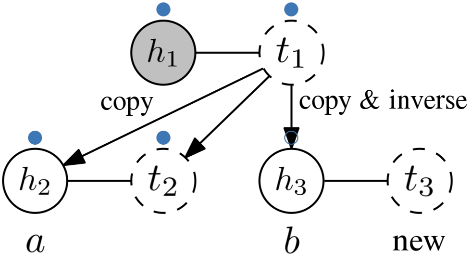
OnlinePLT with Leaf Expansion

example: (x_2, b)



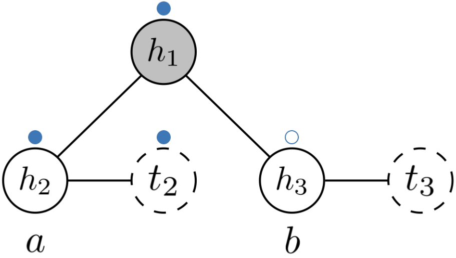
OnlinePLT with Leaf Expansion

example: (x_2, b)



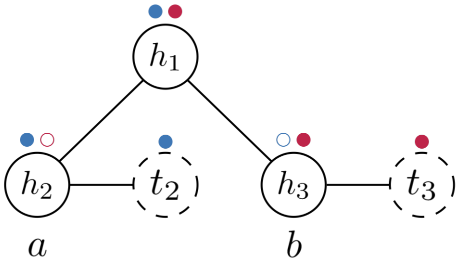
OnlinePLT with Leaf Expansion

example: (x_2, b)



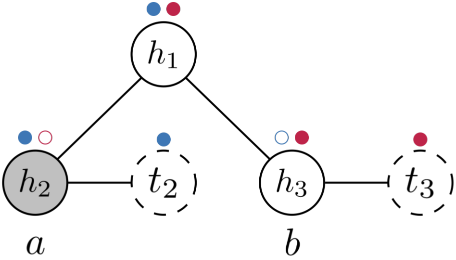
OnlinePLT with Leaf Expansion

example: (x_2, b)



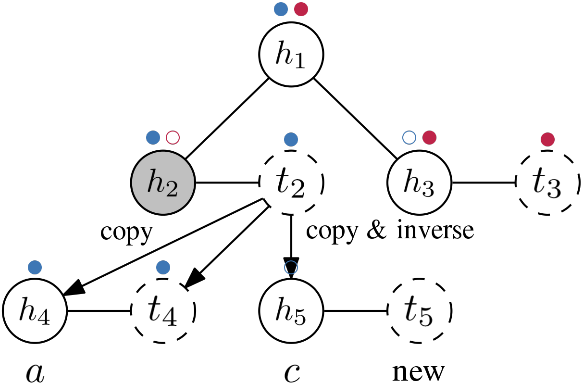
OnlinePLT with Leaf Expansion

example: (x_3, c)



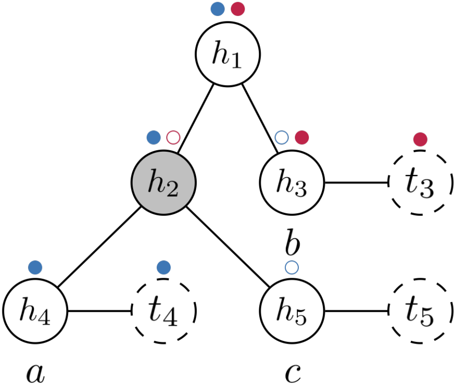
OnlinePLT with Leaf Expansion

example: (x_3, c)



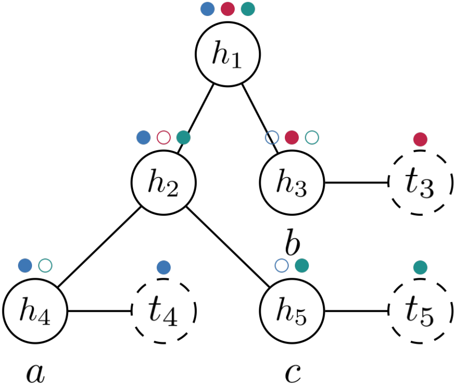
OnlinePLT with Leaf Expansion

example: (x_3, c)



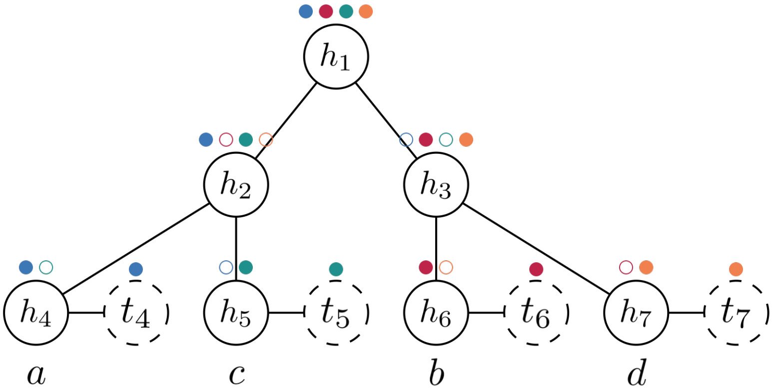
OnlinePLT with Leaf Expansion

example: (x_3, c)



OnlinePLT with Leaf Expansion

example: (x_4, d)



OnlinePLT with Root Expansion

- Additional temporary classifiers required:
  OPLT-LE: from m to tree size
  OPLT-RE: from 1 to $\lceil \log_b(m) \rceil$
- Label placement:
  OPLT-LE: Labels that came early end positioned at the opposite sides of the tree structure.
  OPLT-RE: Placing the labels in order of their expected prior probability.

OnlinePLT – real world datasets

Dataset	$P@k$	PLT	OPLT-LE	OPLT-RE
AmazonCat	$P@1$	91.47	91.24	**91.71**
	$P@3$	75.84	74.81	**76.14**
	$P@5$	61.02	58.79	**61.41**
Wiki10	$P@1$	**84.34**	83.57	84.07
	$P@3$	72.34	72.00	**72.59**
	$P@5$	62.72	62.80	**62.94**
Delicious	$P@1$	45.37	44.60	**45.50**
	$P@3$	38.94	39.22	**39.69**
	$P@5$	35.88	36.51	**36.88**
WikiLSHTC	$P@1$	**45.67**	42.93	44.49
	$P@3$	29.13	26.39	**29.21**
	$P@5$	21.95	18.55	**22.21**
Amazon	$P@1$	**36.65**	29.77	33.05
	$P@3$	**32.12**	26.44	29.44
	$P@5$	**28.85**	23.82	26.82

Outline

Learning of Probabilistic Label Tree

How to learn the tree structure of a PLT in a **batch** setting?

Learning of Probabilistic Label Tree

How to train a PLT in a **top-down** manner?

FastXML[12] – multi-label decision tree

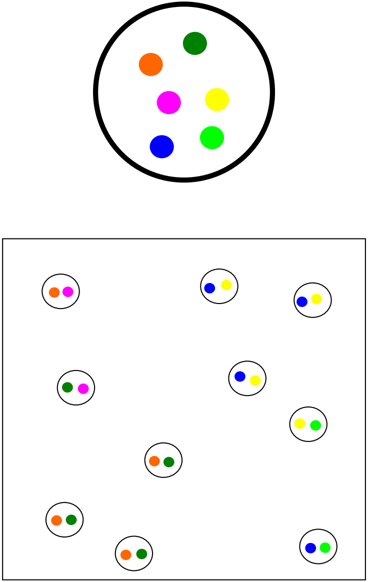
- Problems to address:
  - How to train a **decision tree** in **extreme multi-label** setting?
  - How to **divide examples** among the node's children?
  - How to **optimize precision@k** in decision tree learning?
- Optimize in each node:

$$\min_{\boldsymbol{w},\boldsymbol{\delta},\boldsymbol{r}} \|\boldsymbol{w}\|_1 + F(\boldsymbol{\delta}, L_{log}(\boldsymbol{w}, X)) + G(\boldsymbol{\delta}, L_{\mathcal{NDCG@K}}(\boldsymbol{r}, Y))$$

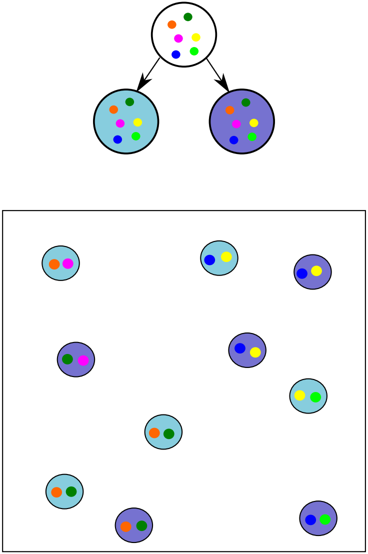
- Efficient optimization via alternate optimization with respect to model weights $\boldsymbol{w}$, left and right label rankings $\boldsymbol{r}$ and example assignment $\boldsymbol{\delta}$.

[12] `https://www.youtube.com/watch?v=1X71fTx1LKA`

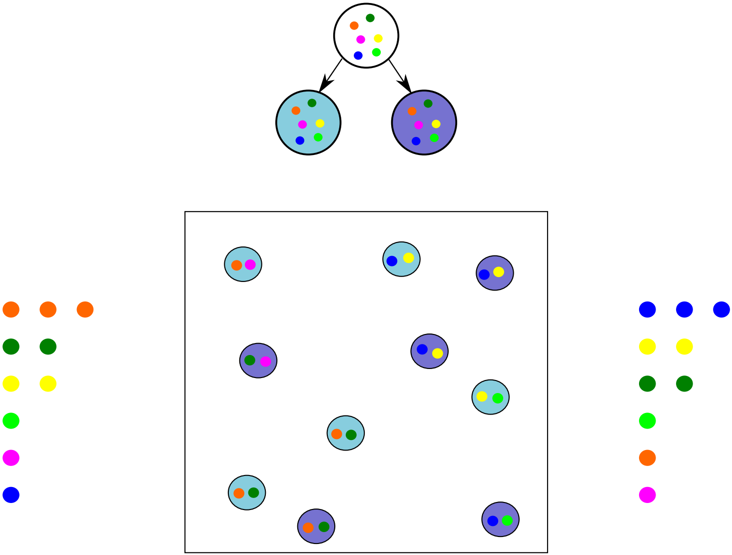
FastXML – multi-label decision tree



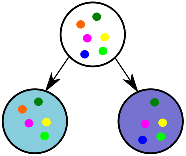
FastXML – multi-label decision tree

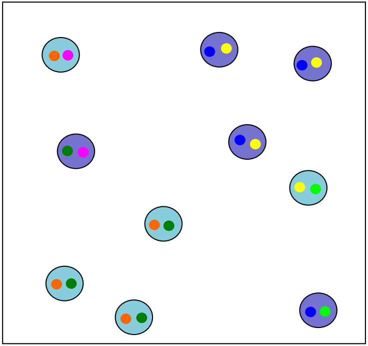


FastXML – multi-label decision tree

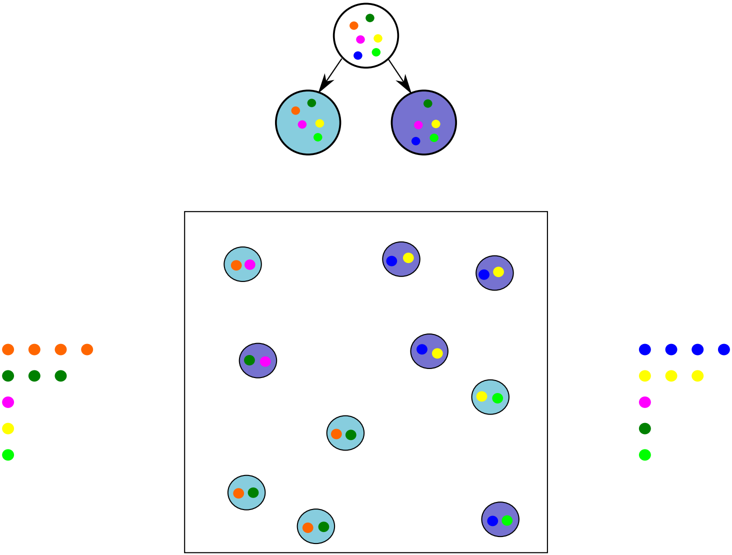


FastXML – multi-label decision tree

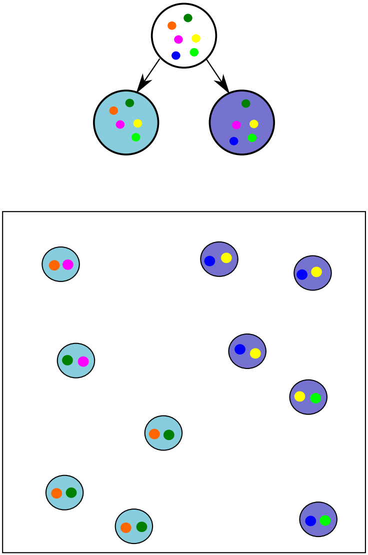




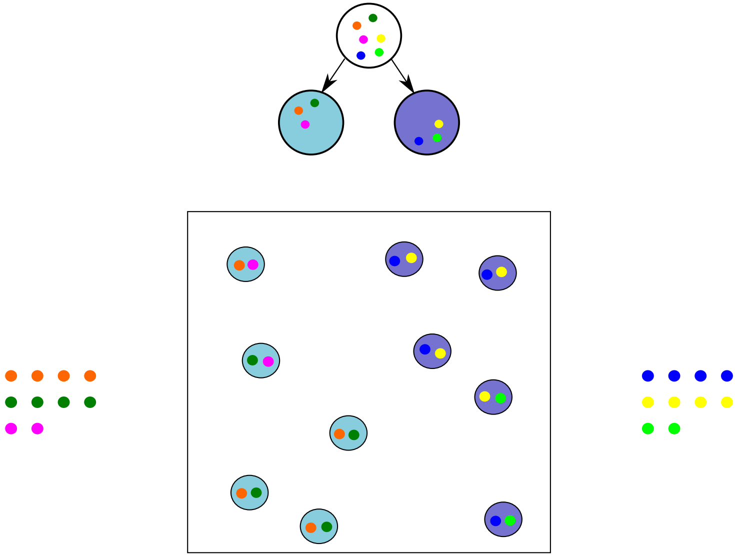
FastXML – multi-label decision tree



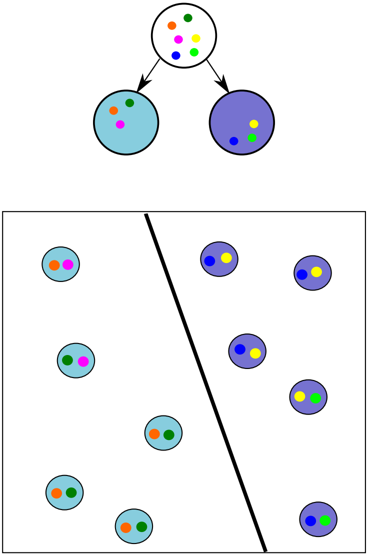
FastXML – multi-label decision tree



FastXML – multi-label decision tree



FastXML – multi-label decision tree



FastPLT

How to apply this idea to PLT?

FastPLT

- Main differences:
  - a **label tree** instead of a decision tree,
  - **assign labels** instead of examples,
  - train **two models** instead of one.

FastPLT

- Build tree **top-down**.
- In each (**parent**) node optimize:

$$\min_{\boldsymbol{w}_{\mathrm{l}},\boldsymbol{w}_{\mathrm{r}},\boldsymbol{\delta}} L_{log}(\boldsymbol{z}_{\mathrm{l}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{l}}(\boldsymbol{w}_{\mathrm{l}})) + L_{log}(\boldsymbol{z}_{\mathrm{r}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{r}}(\boldsymbol{w}_{\mathrm{r}}))$$

- Optimize the **child nodes** model weights (**prediction**) and label assignment (**ground truth**).

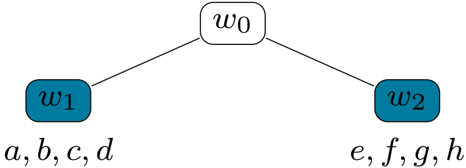
FastPLT – top-down tree building

- Build tree **top-down**.

w_0

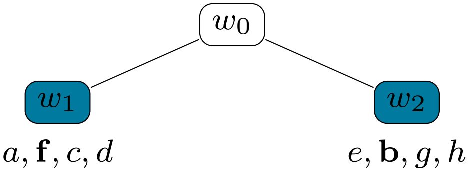
a, b, c, d, e, f, g, h

- Build tree **top-down**.



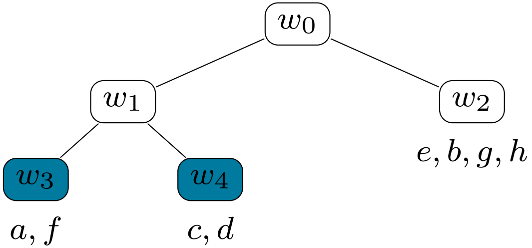
FastPLT – top-down tree building

- Build tree **top-down**.



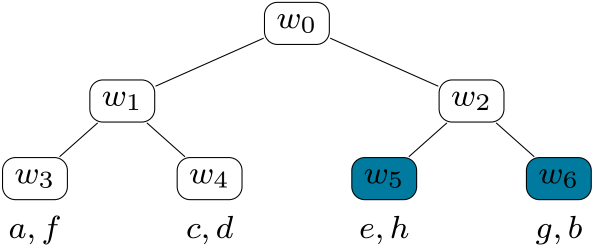
FastPLT – top-down tree building

- Build tree **top-down**.



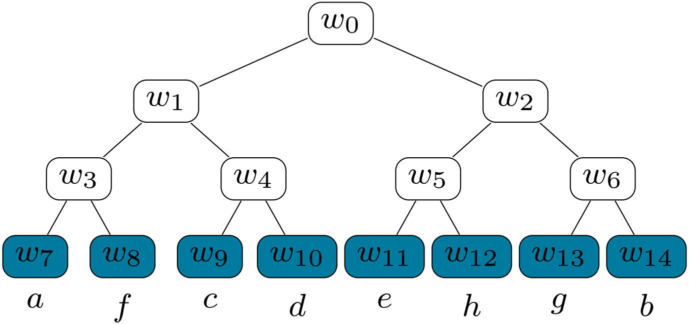
FastPLT – top-down tree building

- Build tree **top-down**.



FastPLT – top-down tree building

- Build tree **top-down**.



FastPLT

- In each node optimize:

$$\min_{\boldsymbol{w}_{\mathrm{l}},\boldsymbol{w}_{\mathrm{r}},\boldsymbol{\delta}} L_{log}(\boldsymbol{z}_{\mathrm{l}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{l}}(\boldsymbol{w}_{\mathrm{l}})) + L_{log}(\boldsymbol{z}_{\mathrm{r}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{r}}(\boldsymbol{w}_{\mathrm{r}}))$$

- Optimization with respect to
  - **model weights** $\boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{r}}$ – solving two logistic regression problems with ground truth determined by the label assignment $\boldsymbol{\delta}$,
  - **label assignment** $\boldsymbol{\delta}$ – move labels left/right until there is a label which results in lower loss when moved.
- The optimization algorithm can be shown to **guarantee convergence to a local minimum**.

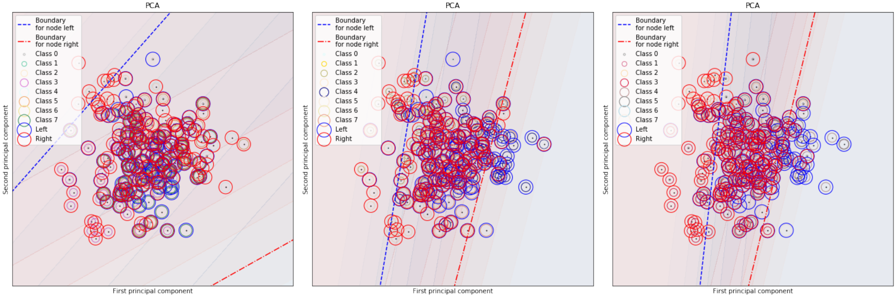
FastPLT

- In each node optimize:

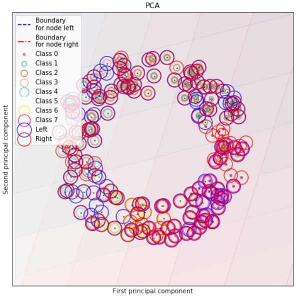
$$\min_{\boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{r}}, \boldsymbol{\delta}} L_{log}(\boldsymbol{z}_{\mathrm{l}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{l}}(\boldsymbol{w}_{\mathrm{l}})) + L_{log}(\boldsymbol{z}_{\mathrm{r}}(\boldsymbol{\delta}), \hat{\boldsymbol{z}}_{\mathrm{r}}(\boldsymbol{w}_{\mathrm{r}}))$$

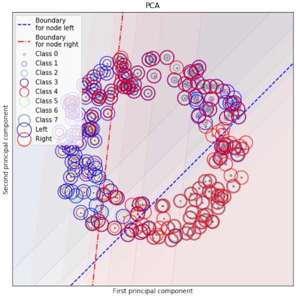
- Optimization with respect to
  - **model weights** $w_{\mathrm{l}}, w_{\mathrm{r}}$ – solving two logistic regression problems with ground truth determined by the label assignment $\boldsymbol{\delta}$,
  - **label assignment** δ – move labels left/right until there is a label which results in lower loss when moved.
- The optimization algorithm can be shown to **guarantee convergence to a local minimum**.

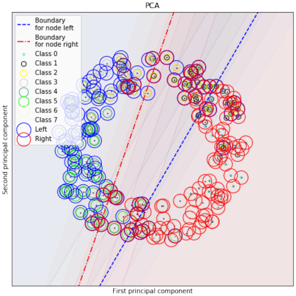
FastPLT – a node prototype



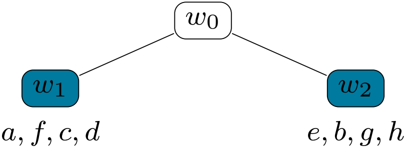
FastPLT – a node prototype

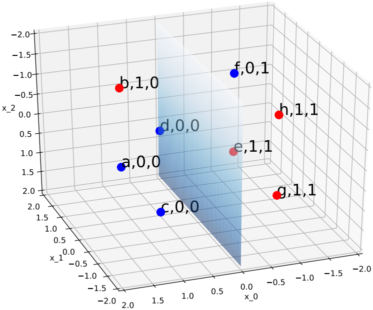




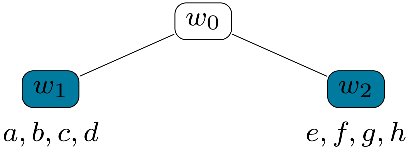


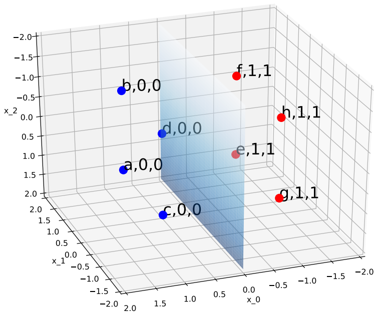
FastPLT – a multi-class example



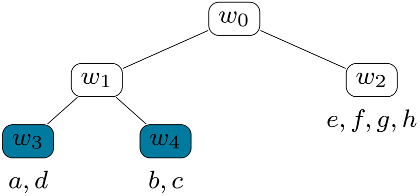


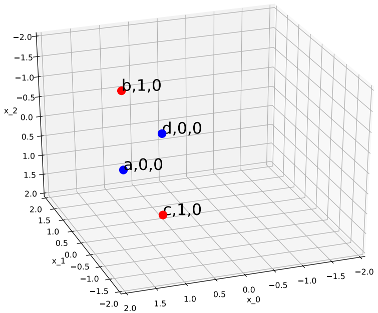
FastPLT – a multi-class example



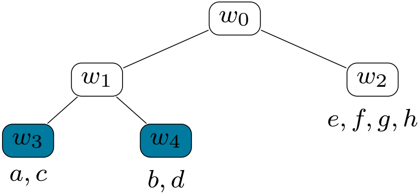


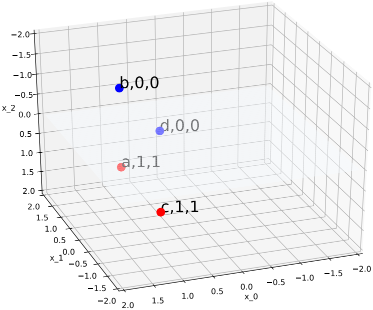
FastPLT – a multi-class example



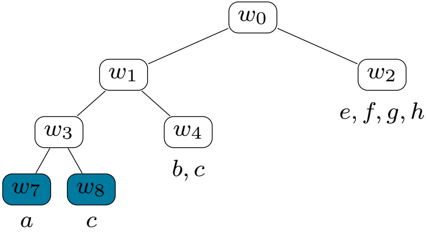


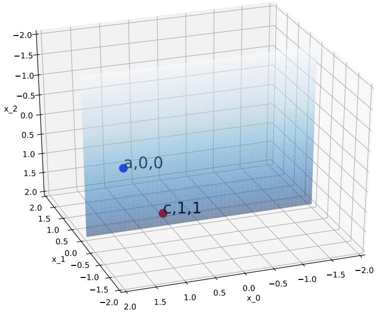
FastPLT – a multi-class example





FastPLT – a multi-class example





FastPLT – real world datasets

- FastPLT is a PLT implementation supporting batch learning,
- Implemented based on FastXML and LIBLINEAR,
- FastPLT was tested on benchmark datasets[13]
- Compared FastPLT tree building policies:
  - in order,
  - random,
  - **fastplt** with in order initialization,
  - **fastplt** with random initialization.

[13] `http://manikvarma.org/downloads/XC/XMLRepository.html`

FastPLT – real world datasets

Dataset	PLT	FastPLT random	FastPLT fastplt random	FastPLT sorted	FastPLT fastplt sorted
RCV1x-2K	**90.46**	88.37	88.57	88.35	**88.76**
AmazonCat-13K	**91.47**	89.83	89.85	90.06	**90.17**
AmazonCat-14K	84.83	85.47	84.88	**85.53**	–
Wiki10-31K	**84.34**	83.71	83.57	**83.80**	–
Delicious-200K	45.37			**45.52**	
WikiLSHTC-325K	**45.67**			42.52	
Amazon-670K	**36.65**			32.38	

Table: Precision@1 FastPLT with L1 regularization

Outline

Summary

- PLT generalizes HSM/CPET to multi-label problems.
- Tree structure learning:
  - Online PLT,
  - FastPLT.
- Promising results on benchmark datasets.