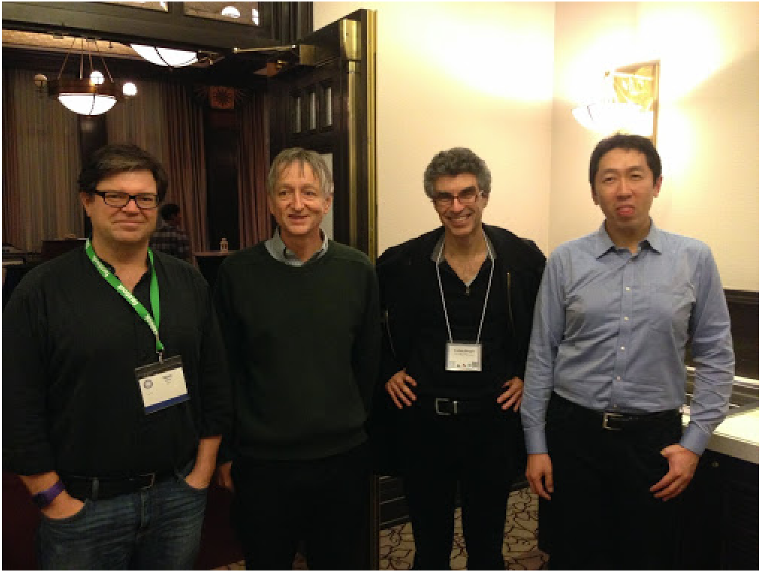
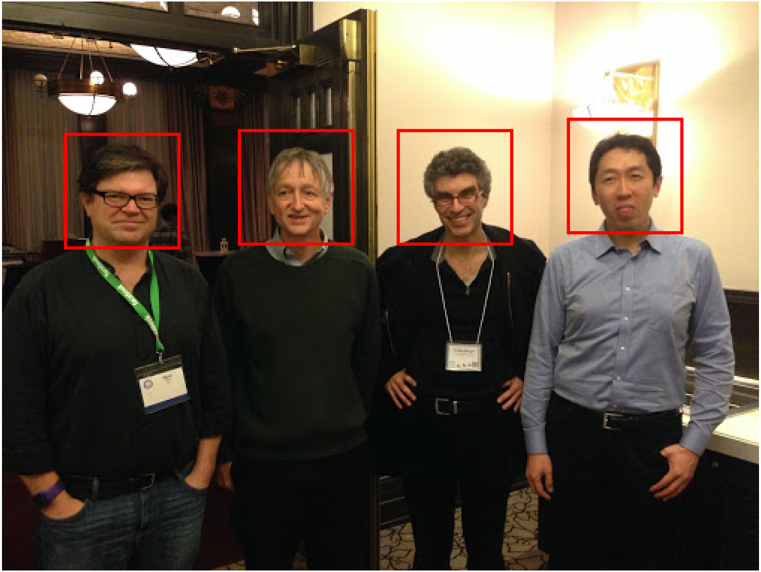
Extreme Zero-Shot Learning (XZSL)

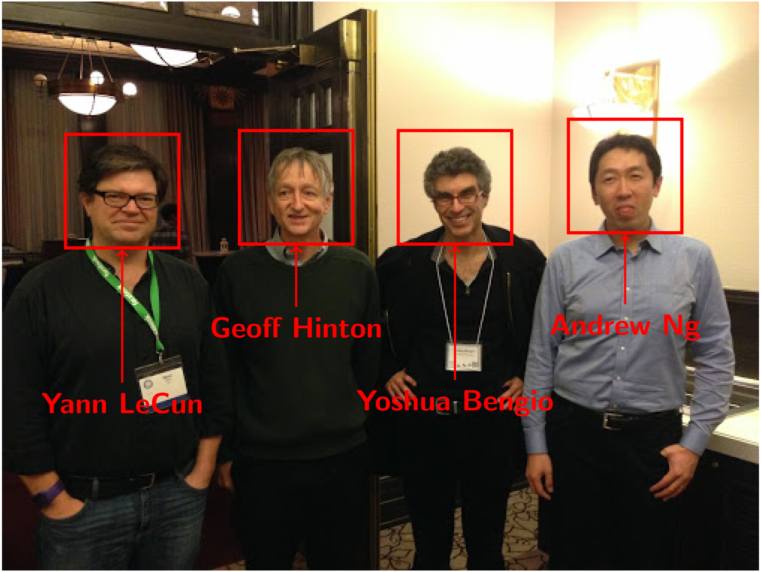
Krzysztof Dembczyński

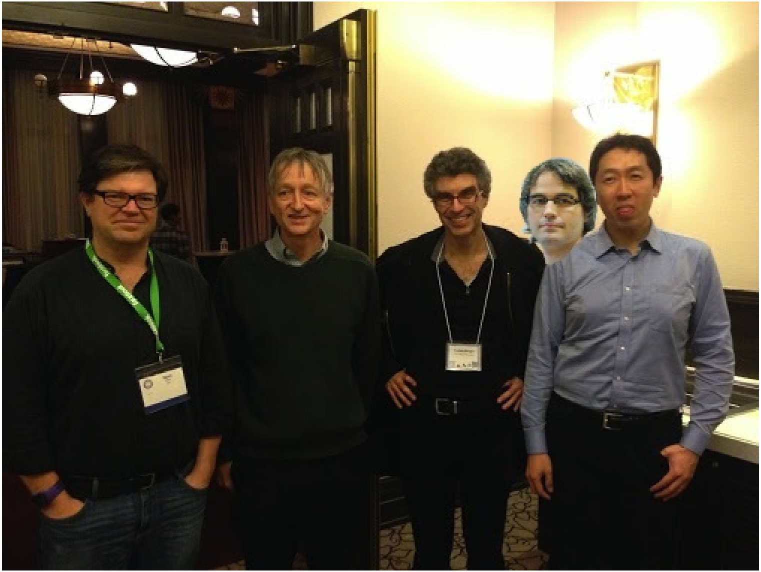
Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland

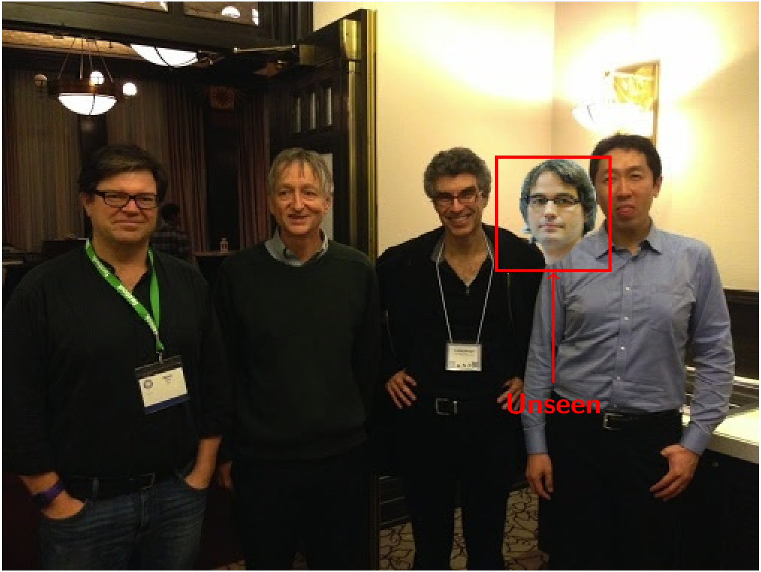
Theoretical Foundations of Machine Learning
Kraków, February 13, 2017

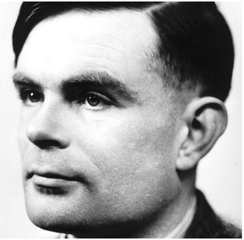


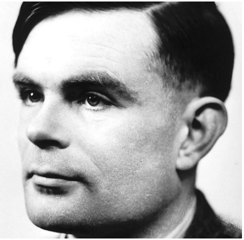



Yann LeCun
Geoff Hinton
Yoshua Bengio
Andrew Ng

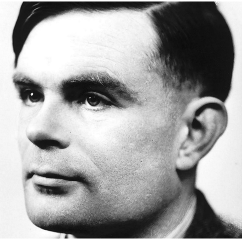



Unseen





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



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

new tags: English machine learners, Polish machine learners, ...

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

Setting

- **Multi-class classification**:

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

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

- **Multi-label classification**:

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

	x_1	x_2	$\dots$	x_d	y_1	y_2	$\dots$	y_m
$\boldsymbol{x}$	4.0	2.5		-1.5	1	1		0

Extreme classification

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

Extreme classification

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

- **Predictive performance**:

Extreme classification

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

- **Predictive performance**:
  - Learning theory for large m

Extreme classification

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

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

Extreme classification

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

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

Extreme classification

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

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

Statistical challenges

- Learning theory for an extremely large number of labels:

Statistical challenges

- Learning theory for an extremely large number of labels:
  - **Statistical guarantees** for the **error** rate that **do not depend**, or depend very weakly (sublinearly), on the **total number of labels**.

Statistical challenges

- Learning theory for an extremely large number of labels:
  - **Statistical guarantees** for the **error** rate that **do not depend**, or depend very weakly (sublinearly), on the **total number of labels**.
  - The **bound** on the error rate could be expressed in terms of the average number of **positive labels** (which is certainly much less than the total number of labels).

Statistical challenges

- Learning theory for an extremely large number of labels:
  - **Statistical guarantees** for the **error** rate that **do not depend**, or depend very weakly (sublinearly), on the **total number of labels**.
  - The **bound** on the error rate could be expressed in terms of the average number of **positive labels** (which is certainly much less than the total number of labels).
  - Particular performance guarantees depend on the considered **loss function**.

Statistical challenges

- Learning theory for an extremely large number of labels:
  - **Statistical guarantees** for the **error** rate that **do not depend**, or depend very weakly (sublinearly), on the **total number of labels**.
  - The **bound** on the error rate could be expressed in terms of the average number of **positive labels** (which is certainly much less than the total number of labels).
  - Particular performance guarantees depend on the considered **loss function**.
  - **Different theoretical settings**: statistical learning theory, learning reductions, online learning.

Statistical challenges

- Training and prediction under limited time and space budget:

Statistical challenges

- Training and prediction under limited time and space budget:
  - **Restricted** computational **resources** (time and space) for both **training** and **prediction**.

Statistical challenges

- Training and prediction under limited time and space budget:
  - **Restricted** computational **resources** (time and space) for both **training** and **prediction**.
  - A **trade-off** between computational (time and space) **complexity** and the **predictive performance**.

Statistical challenges

- Training and prediction under limited time and space budget:
  - **Restricted** computational **resources** (time and space) for both **training** and **prediction**.
  - A **trade-off** between computational (time and space) **complexity** and the **predictive performance**.
  - By imposing hard constraints on time and space budget, the challenge is then to **optimize** the **predictive performance** of an algorithm under these **constraints**.

Statistical challenges

- Unreliable learning information:

Statistical challenges

- Unreliable learning information:
  - We **cannot** expect that all labels will be properly **checked** and **assigned** to training examples.

Statistical challenges

- Unreliable learning information:
  - We **cannot** expect that all labels will be properly **checked** and **assigned** to training examples.
  - Therefore we often deal with a problem of learning with **missing labels** or learning from **positive and unlabeled examples**.

Statistical challenges

- Performance measures:

Statistical challenges

- Performance measures:
  - Typical performance measures such as **0/1** or **Hamming** loss do **not fit** well to the extreme setting.

Statistical challenges

- Performance measures:
  - Typical performance measures such as **0/1** or **Hamming** loss do **not fit** well to the extreme setting.
  - Other measures are often used such as **precision@k** or the **F-measure**.

Statistical challenges

- Performance measures:
  - Typical performance measures such as **0/1** or **Hamming** loss do **not fit** well to the extreme setting.
  - Other measures are often used such as **precision@k** or the **F-measure**.
  - However, it remains an **open question** how to **design loss functions** suitable for extreme classification.

Statistical challenges

- Long-tail label distributions and zero-shot learning:

Statistical challenges

- Long-tail label distributions and zero-shot learning:
  - A close relation to the problem of **estimating distributions over large alphabets**.

Statistical challenges

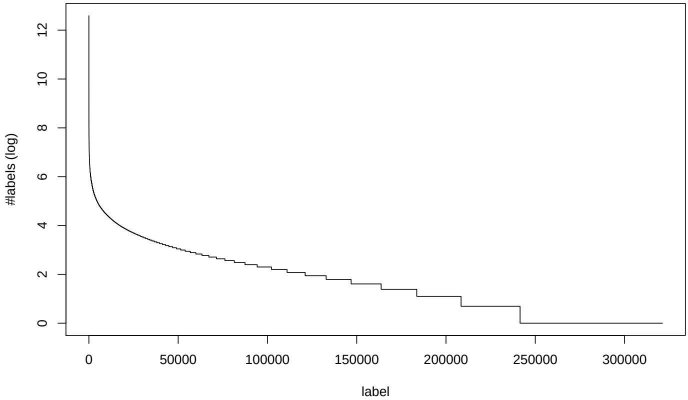
- Long-tail label distributions and zero-shot learning:
  - A close relation to the problem of **estimating distributions over large alphabets**.
  - The distribution of label frequencies is often characterized by a **long-tail** for which proper **smoothing** (like add-constant or Good-Turing estimates) or **calibration** techniques (like isotonic regression or domain adaptation) have to be used.

Statistical challenges

- Long-tail label distributions and zero-shot learning:
  - A close relation to the problem of **estimating distributions over large alphabets**.
  - The distribution of label frequencies is often characterized by a **long-tail** for which proper **smoothing** (like add-constant or Good-Turing estimates) or **calibration** techniques (like isotonic regression or domain adaptation) have to be used.
  - In practical applications, learning algorithms run in **rapidly changing environments**: **new labels** may appear during testing/prediction phase ($\Rightarrow$ **zero-shot learning**)

Statistical challenges

- Long-tail label distributions and zero-shot learning:
  - Frequency of labels in the WikiLSHTC dataset:[1]

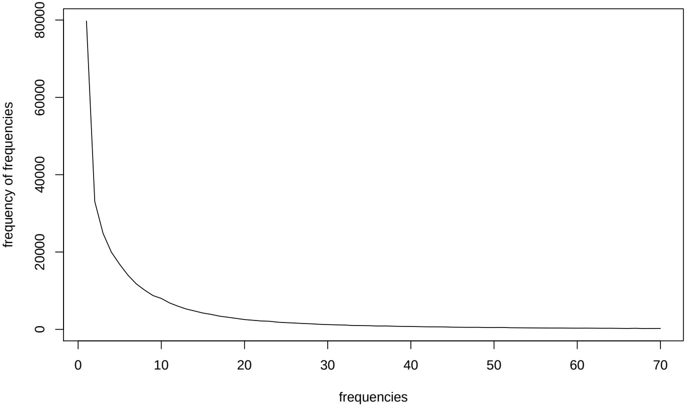


  - Many labels with only few examples ($\Rightarrow$ one-shot learning).

[1] http://research.microsoft.com/en-us/um/people/manik/downloads/XC/XMLRepository.html

Statistical challenges

- Long-tail label distributions and zero-shot learning:
  - Frequency of frequencies for the WikiLSHTC dataset:



  - The missing mass obtained by the Good-Turing estimate: 0.014.

Computational challenges: naive solution

- Size of the problem:

Computational challenges: naive solution

- Size of the problem:
  - $\#$ examples: $n > 10^6$

Computational challenges: naive solution

- Size of the problem:
  - $\#$ examples: $n > 10^6$
  - $\#$ features: $d > 10^6$

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity:

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity: $> 10^{17}$

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity: $> 10^{17}$
  - Space complexity:

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity: $> 10^{17}$
  - Space complexity: $> 10^{11}$

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T \mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity: $> 10^{17}$
  - Space complexity: $> 10^{11}$
  - Test time complexity:

Computational challenges: naive solution

- Size of the problem:
  - # examples: $n > 10^6$
  - # features: $d > 10^6$
  - # labels: $m > 10^5$
- Naive solution: A dense linear model for each label (1-vs-All):

$$\mathbf{X}^T\mathbf{W} = \hat{\mathbf{Y}}$$

  - Train time complexity: $> 10^{17}$
  - Space complexity: $> 10^{11}$
  - Test time complexity: $> 10^{11}$

Computational challenges: naive solution

- **It does not have to be so hard**:

Computational challenges: naive solution

- **It does not have to be so hard**:
  - Large data $\longrightarrow$ sparse data (sparse features and labels)

Computational challenges: naive solution

- **It does not have to be so hard**:
  - Large data $\longrightarrow$ sparse data (sparse features and labels)
  - Fast learning algorithms for standard learning problems exist!

Computational challenges: naive solution

- **It does not have to be so hard**:
  - Large data $\longrightarrow$ sparse data (sparse features and labels)
  - Fast learning algorithms for standard learning problems exist!
  - High performance computing resources available!

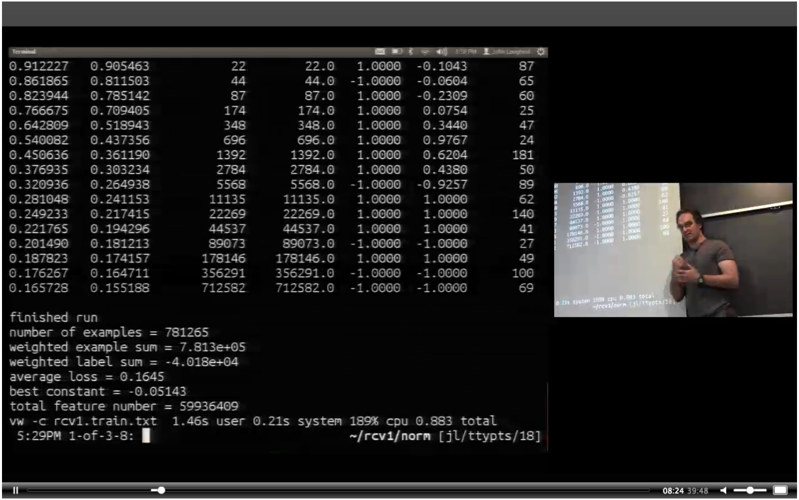


Figure: Vowpal Wabbit[2] at a lecture of John Langford[3]

[2] Vowpal Wabbit, `http://hunch.net/~vw`

[3] `http://cilvr.cs.nyu.edu/doku.php?id=courses:bigdata:slides:start`

Fast binary classification[4]

- Data set: **RCV1**
- Predicted category: CCAT
- # training examples: 781 265
- # features: 60M
- Size: 1.1 GB
- Command line: `time vw -sgd rcv1.train.txt -c`
- Learning time: 1-3 secs on a laptop.

[4] `http://cilvr.cs.nyu.edu/doku.php?id=courses:bigdata:slides:start`

Computational challenges

How can we reduce computational (time and space) costs of the naive solution?

Linear models

- Fast training by least squares:[5]

[5] T. Hastie, R. Tibshirani, and J.H. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009

Linear models

- Fast training by least squares:[5]

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

[5] T. Hastie, R. Tibshirani, and J.H. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009

Linear models

- Fast training by least squares:[5]

$$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

- Works well in low dimensional feature spaces.
- Does not realy improve space and test time complexity.

[5] T. Hastie, R. Tibshirani, and J.H. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009

Linear models

- Training time complexity:

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])
  - Negative sampling.[9]

---

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])
  - Negative sampling.[9]
- Space complexity:

---

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])
  - Negative sampling.[9]
- Space complexity:
  - Proper regularization: L_1 vs L_2.

---

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])
  - Negative sampling.[9]
- Space complexity:
  - Proper regularization: L_1 vs L_2.
  - Feature hashing.[10]

---

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

[8] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

[10] K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

[11] Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Training time complexity:
  - Stochastic gradient descent[6] or coordinate gradient descent[7]
  - Sparse feature vectors (e.g., sparse updates in SGD[8])
  - Negative sampling.[9]
- Space complexity:
  - Proper regularization: L_1 vs L_2.
  - Feature hashing.[10]
  - Removing small weights.[11]

---

6 L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *COMPSTAT*, pages 177–187, Paris, France, August 2010. Springer

7 R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008

8 John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934, 2009

9 Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008

10 K.Q. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120. ACM, 2009

11 Rohit Babbar and Bernhard Schölkopf. Dismec - distributed sparse machines for extreme multi-label classification. *CoRR*, 2016

Linear models

- Low-dimensional representation of $\mathbf{X}$, $\mathbf{W}$, $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{U}^{\dagger}\mathbf{V}\mathbf{X}$$

  - feature space: PCA on $\mathbf{X}$.
  - label space: PCA on $\mathbf{Y}$,[12] compressed sensing,[13] etc.
  - both spaces: CCA on both $\mathbf{X}$ and $\mathbf{Y}$,[14] etc.
  - matrix factorization of $\mathbf{W}$.[15]
  - A kind of **lossy compression**/**embedding** methods.

---

[12] F. Tai and H.-T. Lin. Multi-label classification with principal label space transformation. In *Neural Computat.*, volume 9, pages 2508–2542, 2012

[13] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009

[14] Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, pages 1529–1537. Curran Associates, Inc., 2012

[15] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit S. Dhillon. Large-scale Multi-label Learning with Missing Labels. In *ICML*, 2014

Computational challenges

- Prediction time is still **linear** in the number of labels!

Computational challenges

- Prediction time is still **linear** in the number of labels!

How to reduce the test time complexity?

Test time complexity for linear models

- Classification of a test example in case of linear models can be formulated as:

$$i^* = \underset{i \in \{1,\dots,m\}}{\arg\max}\, \boldsymbol{w}_i \boldsymbol{x} \,,$$

  i.e., the problem of **maximum inner product search** (**MIPS**).
- Exact solution: the threshold algorithm[16]

[16] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In *PODS '01*, pages 102–113. ACM, New York, NY, USA, 2001

MIPS vs. nearest neighbors

- MIPS is similar, but not the same, to nearest neighbor search under the square or cosine distance:

$$
\begin{aligned}
i^* &= \operatorname*{arg\,min}_{i \in \{1,\ldots,m\}} \|\boldsymbol{w}_i - \boldsymbol{x}\|_2^2 = \operatorname*{arg\,max}_{i \in \{1,\ldots,m\}} \boldsymbol{w}_i \boldsymbol{x} - \frac{\|\boldsymbol{w}_i\|_2^2}{2} \\
i^* &= \operatorname*{arg\,max}_{i \in \{1,\ldots,m\}} \frac{\boldsymbol{w}_i \boldsymbol{x}}{\|\boldsymbol{w}_i\|\|\boldsymbol{x}\|} = \operatorname*{arg\,max}_{i \in \{1,\ldots,m\}} \frac{\boldsymbol{w}_i \boldsymbol{x}}{\|\boldsymbol{w}_i\|}
\end{aligned}
$$

- Some tricks are used to treat MIPS as nearest neighbor search.[17]

[17] A. Shrivastava and P. Li. Improved asymmetric locality sensitive hashing (ALSH) for maximum inner product search (mips). In *UAI*, 2015

Sudheendra Vijayanarasimhan, Jonathon Shlens, Rajat Monga, and Jay Yagnik. Deep networks with large output spaces. *CoRR*, abs/1412.7479, 2014

J. Yagnik, D. Strelow, D. A. Ross, and R. s. Lin. The power of comparative reasoning. In *International Conference on Computer Vision*, pages 2431–2438, Nov 2011

Decision trees

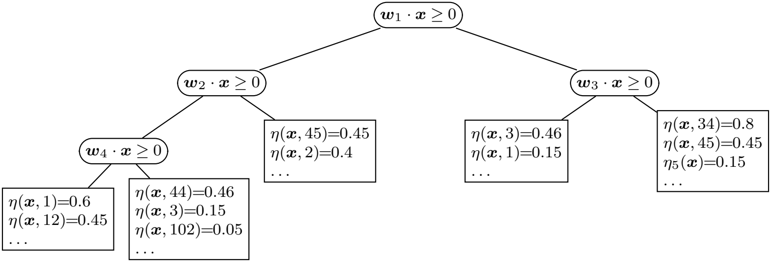
- Fast prediction: logarithmic in n
- Training can be expensive: computation of split criterion
- Two new algorithms: LomTree[18] and FastXML[19]

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

[19] 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

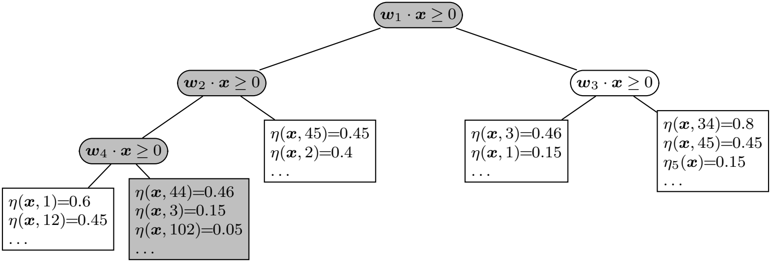
FastXML

- Uses an **ensemble** of standard decision trees.
- **Sparse linear** classifiers trained in internal nodes.
- Very **efficient** training procedure.
- **Empirical distributions** in leaves.
- A test example passes **one path** from the root to a leaf.



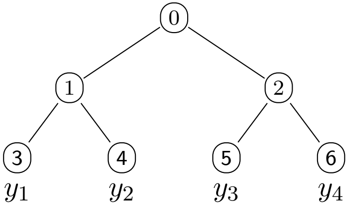
FastXML

- Uses an **ensemble** of standard decision trees.
- **Sparse linear** classifiers trained in internal nodes.
- Very **efficient** training procedure.
- **Empirical distributions** in leaves.
- A test example passes **one path** from the root to a leaf.



Label trees

- Organize classifiers in a tree structure (one leaf ⇔ one label).[20]

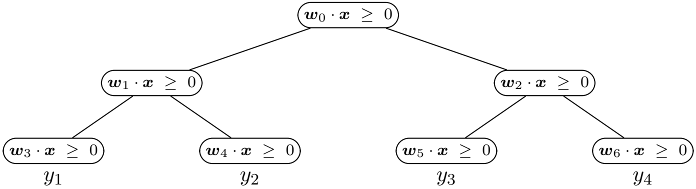


- Structure of the tree can be given or trained.
- Different training and test procedures for multi-class and multi-label classification.

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

Probabilistic label trees (PLT)[21]

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

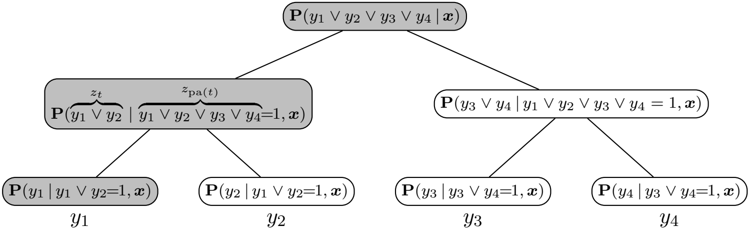


- **Probabilistic classifiers** in **all** nodes of the tree.
- **Internal** node classifier decides whether to **go down the tree**.
- A test example may follow **many paths** from the root to leaves.

[21] 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 $\mathbf{P}(y_i = 1 \,|\, \boldsymbol{x})$.



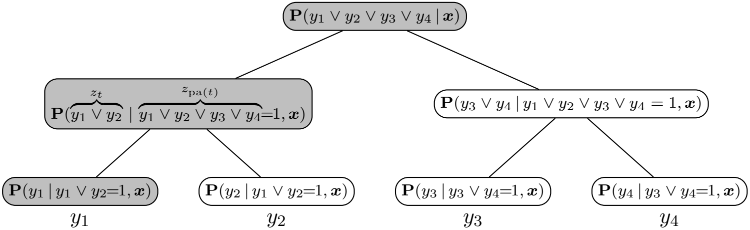
- Using the **chain rule** of probability

$$\mathbf{P}(y_i = 1 \,|\, \boldsymbol{x}) = \eta(\boldsymbol{x}, i) = \prod_{t \in \mathrm{Path}(i)} \eta_T(\boldsymbol{x}, t)\,,$$

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

Probabilistic label trees

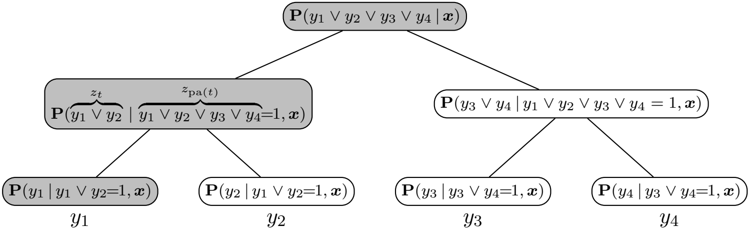
- Class probability estimators in nodes for estimating $\mathbf{P}(y_i = 1 \,|\, \boldsymbol{x})$.



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

Probabilistic label trees

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



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

Probabilistic label trees

- The same idea under different names:
  - **Conditional probability trees**[22]
  - **Probabilistic classifier chains**[23]
  - **Hierarchical softmax**[24]
  - **Homer**[25]
  - **Nested dichotomies**[26]
  - **Multi-stage classification**[27]

---

[22] 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

[23] K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, pages 279–286. Omnipress, 2010

[24] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS*, pages 246–252, 2005

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

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

[27] Marek Kurzynski. On the multistage bayes classifier. *Pattern Recognition*, 21(4):355–365, 1988

FastXML vs. PLT

	FastXML	PLT
tree structure	✓	✓
structure learning	✓	$\times$
number of trees	≥ 1	1
number of leaves	linear in # examples	m
internal nodes models	linear	linear
leaves models	empirical distribution	linear
visited paths during prediction	1 per tree	several
sparse probability estimation	✓	✓

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.[28]

[28] `http://research.microsoft.com/en-us/um/people/manik/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

Experimental results

	PLT					FastXML			
	train [min]	test [ms]	b	depth	#calls	train [min]	test [ms]	depth	#calls
RCV1	**64**	0.22	32	2,25	**43,46**	78	0.96	14.95	747
AmazonCat	**96**	0.17	16	3,43	**54,39**	561	1.14	17.44	871
Wiki10	290	2.66	32	2,98	**121,98**	**16**	3.00	10.83	541
Delicious	1327	32.97	2	17,69	11779,65	**458**	4.01	14.79	**739**
WikiLSHTC	**653**	3.00	32	3,66	**622,27**	724	1.17	18.01	900
Amazon	**54**	0.99	8	6,45	**374,30**	422	1.39	15.92	796

Reduction to structured output prediction[29]

- Reduce extreme classification to structured output prediction:

[29] Joint work with Kalina Jasinska and Nikos Karampatziakis

Reduction to structured output prediction[29]

- Reduce extreme classification to structured output prediction:
  - encode labels by **sequences of bits**,

[29] Joint work with Kalina Jasinska and Nikos Karampatziakis

Reduction to structured output prediction[29]

- Reduce extreme classification to structured output prediction:
  - encode labels by **sequences of bits**,
  - choose a proper **dependence structure** between bits,

[29] Joint work with Kalina Jasinska and Nikos Karampatziakis

Reduction to structured output prediction[29]

- Reduce extreme classification to structured output prediction:
  - encode labels by **sequences of bits**,
  - choose a proper **dependence structure** between bits,
  - use appropriate **training** and **inference** methods.

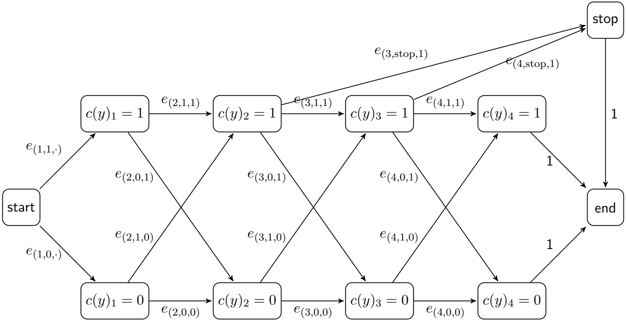
[29] Joint work with Kalina Jasinska and Nikos Karampatziakis

Reduction to structured output prediction[29]

- Reduce extreme classification to structured output prediction:
  - encode labels by **sequences of bits**,
  - choose a proper **dependence structure** between bits,
  - use appropriate **training** and **inference** methods.
- Enables to work under **limited time** and **space budget**.

[29] Joint work with Kalina Jasinska and Nikos Karampatziakis

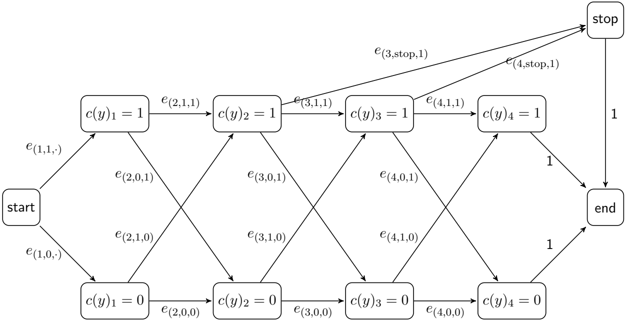
LTLS



- LTLS[30] encodes labels as paths in a **trellis** of width 2.

[30] Kalina Jasinska and Nikos Karampatziakis. Log-time and log-space extreme classication. In *Extreme Classification workshop at NIPS*, 2016

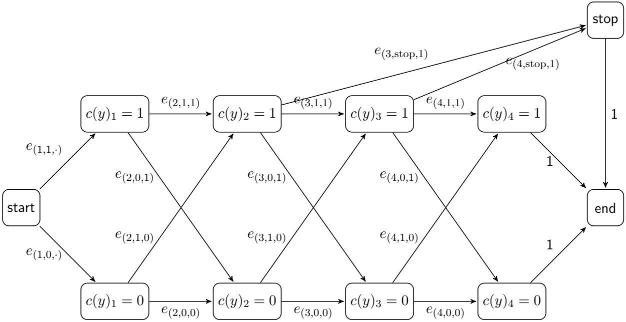
LTLS



- LTLS[30] encodes labels as paths in a **trellis** of width 2.
- Each **path** corresponds to one and only one label.

[30] Kalina Jasinska and Nikos Karampatziakis. Log-time and log-space extreme classication. In *Extreme Classification workshop at NIPS*, 2016

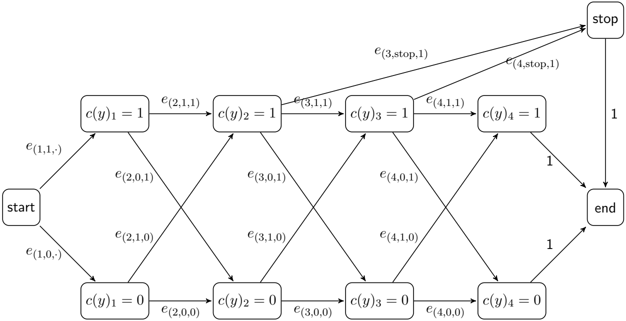
LTLS



- LTLS[30] encodes labels as paths in a **trellis** of width 2.
- Each **path** corresponds to one and only one label.
- Training concerns models on **edges**.

[30] Kalina Jasinska and Nikos Karampatziakis. Log-time and log-space extreme classication. In *Extreme Classification workshop at NIPS*, 2016

LTLS



- LTLS[30] encodes labels as paths in a **trellis** of width 2.
- Each **path** corresponds to one and only one label.
- Training concerns models on **edges**.
- Prediction of the most probable labels corresponds to **finding** the most probable paths.

---

[30] Kalina Jasinska and Nikos Karampatziakis. Log-time and log-space extreme classication. In *Extreme Classification workshop at NIPS*, 2016

LTLS

- Can be trained with logistic loss or a variant of structured hinge loss[31]

[31] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, 2016

LTLS

- Can be trained with logistic loss or a variant of structured hinge loss[31]
- Can work with any number of labels (not only the powers of 2).

[31] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, 2016

LTLS

- Can be trained with logistic loss or a variant of structured hinge loss[31]
- Can work with any number of labels (not only the powers of 2).
- The number of edges is upperbounded by $5\lceil \log_2 m \rceil + 1$

[31] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, 2016

LTLS

- Can be trained with logistic loss or a variant of structured hinge loss[31]
- Can work with any number of labels (not only the powers of 2).
- The number of edges is upperbounded by $5\lceil\log_2 m\rceil + 1$
- The inference can be efficiently performed by the Viterbi algorithm.

[31] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, 2016

LTLS

- Can be trained with logistic loss or a variant of structured hinge loss[31]
- Can work with any number of labels (not only the powers of 2).
- The number of edges is upperbounded by $5\lceil \log_2 m \rceil + 1$
- The inference can be efficiently performed by the Viterbi algorithm.
- Therefore, the space and time complexity of training and testing is **logarithmic** in the number of labels.

[31] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, 2016

Experimental results

		LTLS-LR	PLT	LOMtree	FastXML
sector	P@1	0.8616	0.8730	0.8210	0.8490
	model size	12.06	16.00	17.00	7.00
aloi.bin	P@1	0.8128	0.9088	0.8947	0.9550
	model size	209	128	106	992
Dmoz	P@1	0.2082	0.3263	0.2127	0.3840
	model size	397	2048	1800	1500
LSHTC1	precision@1	0.0950	0.1524	0.1056	0.2166
	model size	525	1024	744	308

Extreme Zero-Shot Learning[32]

- In traditional setting the target variable is a binary indicator.
- We replace the binary indicator by a richer representation.
- We use a textual description of a label.

[32] Joint work with Marcin Elantkowski and Moustapha Cisse

Extreme Zero-Shot Learning

- Consider a multi-label dataset with n training examples and m labels of the form

$$\{(\boldsymbol{x}_i, \mathcal{Y}_i)\}_{i=1}^{n}$$

Extreme Zero-Shot Learning

- Consider a multi-label dataset with n training examples and m labels of the form

$$\{(\boldsymbol{x}_i, \mathcal{Y}_i)\}_{i=1}^n$$

- $\mathcal{Y}_i = \{\ell_1^i, \ell_2^i, \ldots, \ell_{k_i}^i\}$ is a set of labels relevant for the i-th example, with $0 \leq k_i \leq m$ and $\ell_j^i \in \mathcal{Y}_s$, where $\mathcal{Y}_s = \{1, 2, \ldots, m\}$ is a set of all labels in the training data.

Extreme Zero-Shot Learning

- Consider a multi-label dataset with n training examples and m labels of the form

$$\{(\boldsymbol{x}_i, \mathcal{Y}_i)\}_{i=1}^{n}$$

- $\mathcal{Y}_i = \{\ell_1^i, \ell_2^i, \ldots, \ell_{k_i}^i\}$ is a set of labels relevant for the i-th example, with $0 \leq k_i \leq m$ and $\ell_j^i \in \mathcal{Y}_s$, where $\mathcal{Y}_s = \{1, 2, \ldots, m\}$ is a set of all labels in the training data.
- $\bar{\mathcal{Y}}_i = \mathcal{Y}_s \setminus \ \mathcal{Y}_i$ is a set of labels irrelevant for the i-th example.

Extreme Zero-Shot Learning

- Consider a multi-label dataset with n training examples and m labels of the form
$$\{(\boldsymbol{x}_i, \mathcal{Y}_i)\}_{i=1}^{n}$$
- $\mathcal{Y}_i = \{\ell_1^i, \ell_2^i, \ldots, \ell_{k_i}^i\}$ is a set of labels relevant for the i-th example, with $0 \leq k_i \leq m$ and $\ell_j^i \in \mathcal{Y}_s$, where $\mathcal{Y}_s = \{1, 2, \ldots, m\}$ is a set of all labels in the training data.
- $\bar{\mathcal{Y}}_i = \mathcal{Y}_s \setminus \ \mathcal{Y}_i$ is a set of labels irrelevant for the i-th example.
- $\boldsymbol{x}_i$ is a sparse vector of features (e.g., word indices) from a set $\mathcal{V}_{\mathcal{X}}$, represented as a sequence of features
$$\boldsymbol{x}_i = \{x_1^i, x_2^i, \ldots, x_{d_i}^i\}, \quad x_j^i \in \mathcal{V}_{\mathcal{X}} \,,$$
where d_i denotes the number of non-zero features for the i-th example (e.g., a number of words in a document).

Extreme Zero-Shot Learning

- At test time, we are given a set of m' additional labels

$$\mathcal{Y}_u = \{m+1, m+2, \ldots, m+m'\}$$

that can be assigned to test examples.

Extreme Zero-Shot Learning

- At test time, we are given a set of m' additional labels

$$\mathcal{Y}_u = \{m+1, m+2, \ldots, m+m'\}$$

  that can be assigned to test examples.
- These labels have not been observed during training, i.e. $\mathcal{Y}_s \cap \mathcal{Y}_u = \emptyset$.

Extreme Zero-Shot Learning

- At test time, we are given a set of m' additional labels

$$\mathcal{Y}_u = \{m+1, m+2, \ldots, m+m'\}$$

  that can be assigned to test examples.
- These labels have not been observed during training, i.e. $\mathcal{Y}_s \cap \mathcal{Y}_u = \emptyset$.
- Additionally, for each label $\ell \in \{1, 2, \ldots, m+m'\}$ we are given its description (e.g., textual description from a dictionary) of the form

$$\boldsymbol{t}_\ell = \{t_1^\ell, t_2^\ell, \ldots, t_{m_\ell}^\ell\}, t_j^\ell \in \mathcal{V}_\mathcal{Y}\,,$$

  where $\mathcal{V}_\mathcal{Y}$ is a separate feature space describing labels, possibly different from $\mathcal{V}_\mathcal{X}$.

Extreme Zero-Shot Learning

- We are interested in finding a scoring function

$$f : (\boldsymbol{x}_i, \ell) \to \mathbb{R}\,,$$

such that $f(\boldsymbol{x}_i, \ell) > f(\boldsymbol{x}_i, \hat{\ell})$ for all $\ell \in \mathcal{Y}_i$, $\hat{\ell} \in \bar{\mathcal{Y}}_i$.

Extreme Zero-Shot Learning

- We are interested in finding a scoring function

$$f : (\boldsymbol{x}_i, \ell) \to \mathbb{R}\,,$$

  such that $f(\boldsymbol{x}_i, \ell) > f(\boldsymbol{x}_i, \hat{\ell})$ for all $\ell \in \mathcal{Y}_i$, $\hat{\ell} \in \bar{\mathcal{Y}}_i$.
- To leverage the provided label description we use:

$$f : (\boldsymbol{x}_i, \boldsymbol{t}_\ell) \to \mathbb{R}\,,$$

  i.e., the label descriptions instead of raw labels.

Extreme Zero-Shot Learning

- To obtain the scoring function we learn a mapping of both:

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

[34] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011

Extreme Zero-Shot Learning

- To obtain the scoring function we learn a mapping of both:
  - feature spaces and

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

[34] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011

Extreme Zero-Shot Learning

- To obtain the scoring function we learn a mapping of both:
  - feature spaces and
  - label descriptions

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

[34] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011

Extreme Zero-Shot Learning

- To obtain the scoring function we learn a mapping of both:
  - feature spaces and
  - label descriptions

  into a common embedding space, in which the dot product between both representations is maximized.

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

[34] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011

Extreme Zero-Shot Learning

- To obtain the scoring function we learn a mapping of both:
  - feature spaces and
  - label descriptions

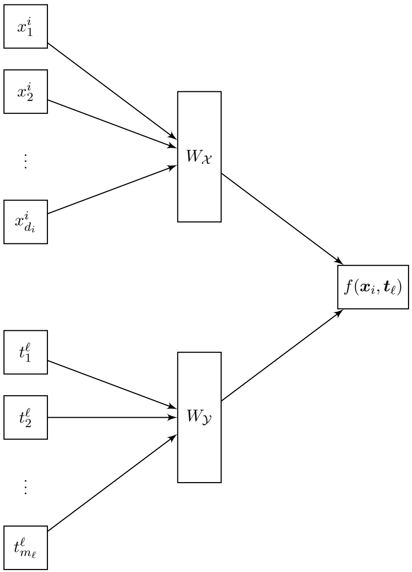
  into a common embedding space, in which the dot product between both representations is maximized.
- While these mappings are different, they are learned jointly to optimize the loss function of choice:
  - logistic loss (like in FastText[33]) or
  - WARP loss (like in Wsabie[34])

---

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

[34] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011

Extreme Zero-Shot Learning



- More formally, we define

$$\phi_{\mathcal{X}}(\boldsymbol{x}_i) : \mathbb{R}^{|\mathcal{V}_{\mathcal{X}}|} \to \mathbb{R}^p \quad \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) : \mathbb{R}^{|\mathcal{V}_{\mathcal{Y}}|} \to \mathbb{R}^p\,,$$

and consider scoring function $f(\cdot)$ of the form:

$$f(\boldsymbol{x}_i, \boldsymbol{t}_\ell) = \phi_{\mathcal{X}}(\boldsymbol{x}_i)^\top \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell)\,.$$

- More formally, we define

$$\phi_{\mathcal{X}}(\boldsymbol{x}_i) : \mathbb{R}^{|\mathcal{V}_{\mathcal{X}}|} \to \mathbb{R}^p \quad \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) : \mathbb{R}^{|\mathcal{V}_{\mathcal{Y}}|} \to \mathbb{R}^p \,,$$

  and consider scoring function $f(\cdot)$ of the form:

$$f(\boldsymbol{x}_i, \boldsymbol{t}_\ell) = \phi_{\mathcal{X}}(\boldsymbol{x}_i)^\top \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) \,.$$

- To obtain representation $\phi(\cdot)$ we use a weight matrix $W \in \mathbb{R}^{|\mathcal{V}| \times p}$ which acts as a look-up table over the features (e.g., words).

Extreme Zero-Shot Learning

- More formally, we define

$$\phi_{\mathcal{X}}(\boldsymbol{x}_i) : \mathbb{R}^{|\mathcal{V}_{\mathcal{X}}|} \to \mathbb{R}^p \quad \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) : \mathbb{R}^{|\mathcal{V}_{\mathcal{Y}}|} \to \mathbb{R}^p \,,$$

  and consider scoring function $f(\cdot)$ of the form:

$$f(\boldsymbol{x}_i, \boldsymbol{t}_\ell) = \phi_{\mathcal{X}}(\boldsymbol{x}_i)^\top \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) \,.$$

- To obtain representation $\phi(\cdot)$ we use a weight matrix $W \in \mathbb{R}^{|\mathcal{V}| \times p}$ which acts as a look-up table over the features (e.g., words).
- Embedding of an example or a label is then obtained by averaging representations of individual features within it.

Extreme Zero-Shot Learning

- More formally, we define

$$\phi_{\mathcal{X}}(\boldsymbol{x}_i) : \mathbb{R}^{|\mathcal{V}_{\mathcal{X}}|} \to \mathbb{R}^p \quad \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) : \mathbb{R}^{|\mathcal{V}_{\mathcal{Y}}|} \to \mathbb{R}^p \,,$$

  and consider scoring function $f(\cdot)$ of the form:

$$f(\boldsymbol{x}_i, \boldsymbol{t}_\ell) = \phi_{\mathcal{X}}(\boldsymbol{x}_i)^\top \phi_{\mathcal{Y}}(\boldsymbol{t}_\ell) \,.$$

- To obtain representation $\phi(\cdot)$ we use a weight matrix $W \in \mathbb{R}^{|\mathcal{V}| \times p}$ which acts as a look-up table over the features (e.g., words).
- Embedding of an example or a label is then obtained by averaging representations of individual features within it.
- We maintain two separate weight matrices, $W_{\mathcal{X}}$ and $W_{\mathcal{Y}}$, for $\phi_{\mathcal{X}}(\cdot)$ and $\phi_{\mathcal{Y}}(\cdot)$, respectively.

Extreme Zero-Shot Learning

- Learning in the extreme setting:
  - Negative sampling
- Testing in the extreme setting:
  - Fast nearest neighbor search in the embedding space
- Problems with unseen labels:
  - Labels can be described by features not seen during training.

XZSL – Experimental results

- We compare the above approach against:
  - k-NN in TF-IDF space
  - Label name in example.
  - fastText[35]
  - All-in-Text[36]
  - FastXML[37]

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

[36] Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. All-in text: Learning document, label, and word representations jointly. In *AAAI Conference on Artificial Intelligence*, pages 1948–1954, 2016

[37] 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

XZSL – Experimental results

- Statistics of the BioASQ dataset

# of train instances	6,792,815
# of test instances	4,912,719
# of seen labels	23,669
# of unseen label	2,435
# of avg. cardinality	10.83
# of words in documents	528,156
# of words in descriptions	39,958

XZSL – Experimental results

	TF-IDF	Label in Exp	FastText	All in Text	XZSL
P@1	0.143	0.050	**0.875**	0.74	0.812
rank loss	0.278	0.50	**0.012**	0.035	0.013
avg. prec.	0.040	0.017	**0.461**	0.327	0.37
P@1 (zsl)	**0.084**	0.048	-	0.013	0.046
rank loss (zsl)	0.217	0.299	-	0.216	**0.095**
avg. prec. (zsl)	**0.142**	0.083	-	0.026	0.1

Conclusions

- **Take-away message**:

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,
    - Unreliable learning information,

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,
    - Unreliable learning information,
    - Long-tail label distributions,

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,
    - Unreliable learning information,
    - Long-tail label distributions,
    - **Zero-shot learning**.

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,
    - Unreliable learning information,
    - Long-tail label distributions,
    - **Zero-shot learning**.
  - For more check:

Conclusions

- **Take-away message**:
  - **Extreme classification**: #examples, #features, **#labels**
  - **Complexity**: time vs. space, training vs. validation vs. **prediction**
  - **Statistical challenges**:
    - Is learning possible in the extreme setting?
    - Training and prediction under limited time and space budged.
    - Performance measures,
    - Unreliable learning information,
    - Long-tail label distributions,
    - **Zero-shot learning**.
  - For more check:
    - `http://www.cs.put.poznan.pl/kdembczynski`