Learning Multiple Tasks with Boosted Decision Trees

Jean Baptiste Faddoul, Boris Chidlovskii, Fabien Torre, Rémi Gilleron

CAp'12, May 2012

Multitask Learning

Multitask Learning MTL considers learning multiple "*related*" tasks jointly, in order to improve their predictive performance.

Related Tasks ?

Related Tasks ?

Table of contents

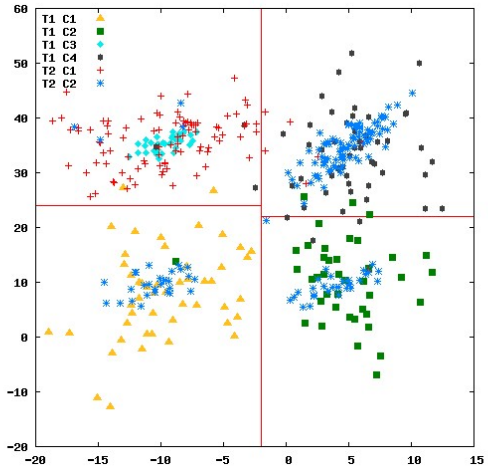
Label Correspondence Assumption

Learning multiple tasks becomes easier under the *label correspondence* assumption, where either:

- Tasks share the same labels sets.
- Tasks share the same training data points, each has a label for each task.

Global Relatedness Assumption

Related tasks might show different relatedness degrees / signs across the learning space.



Prior-Art

- [Quadrianto et al., 2010] formulates MTL as a maximization problem of mutual information among the label sets.
- Their approach assumes a global relatedness pattern between the tasks.
- In previous work [Faddoul et al., 2010] we proposed *MT-Adaboost*, an adaptation of Adaboost to MTL.
- The weak classifier proposed is called *MT-Stump*.

Prior-Art (Cont'd)

- No label correspondence or global relatedness assumptions.
- But, the sequential design of multi-task stump and its greedy algorithm can fail to capture task relatedness.

Contribution

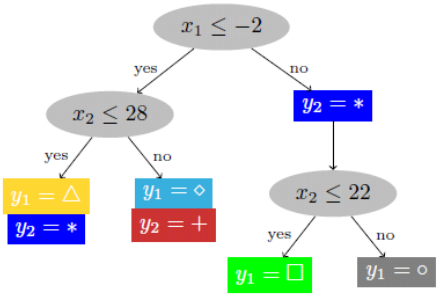
We propose a novel technique for the multi-task learning which addresses the limitations of previous approaches:

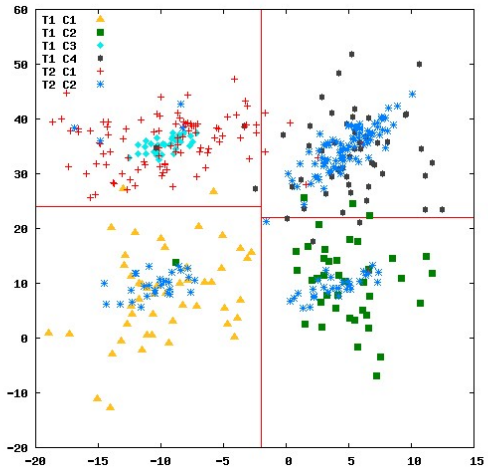
- We propose an adaptation to decision trees learning to multi-task setting.
- We derive an information-theoretic criterion and prove its superiority to baseline information gain.
- We integrate the proposed classifier in boosting framework.

A bit of Notation

- N classification tasks $T_1, \ldots, T_N$ over the instance space $\mathcal{X}$ and label sets $Y_1, \ldots, Y_N$
- Distribution D over $\mathcal{X} \times \{1, \ldots, N\}$.
- Training set $S = \{< x_i, y_i, j > | \; x_i \in \mathcal{X}, \; y_i \in Y_j, \; j \in \{1, \ldots, N\}, \; 1 \leq i \leq m\}$.
- Output $h : \mathcal{X} \rightarrow Y_1 \times \ldots \times Y_N$ which minimizes $\text{error}(h) = Pr_{<x,y,j> \sim D}[h_j(x) \neq y]$, where $h_j(x)$ is the j-th component of $h(x)$.

Multi-Task Decision Tree (MT-DT)





Information Gain

Decision tree learning is based *Information Gain (IG)* criteria.

- $IG(Y;X)$ on a random variable Y obtained after observing the value of X is the Kullback-Leibler divergent $D_{KL}(p(Y|X)||p(Y|I))$
- It is the reduction of Y's entropy obtained by observing the value of X.
- IG defines a preferred sequence of attributes to investigate to most rapidly narrow down the state of Y.

$$IG(Y;X) = H(Y) - H(Y|X),$$

Information Gain for MTL

- As a baseline, we can pool all the tasks as a single multi-class task. The IG in this case will be given by:

$$IG_J = IG(\oplus_{j=1}^{N} Y_j; X)$$

  $\oplus$ indicates the pooling of all tasks' labels

- Another baseline can be the sum of individual IGs:

$$IG_U = \sum_{j=1}^{T} IG(Y_j; X)$$

Information Gain for MTL (Cont'd)

- Evaluations show that IG_U fails to make better than IG_J.
- We prove that IG_J is equivalent to the weighted sum of individual task information gains.
- Then we derive IG_M a criterion superior to IG_J.

Information Gain for MTL (Cont'd)

Theorem

For N tasks with the class sets $Y_1, \ldots, Y_N$, let p_j denote the fraction of task j in the full dataset, $p_j = \frac{|S_j|}{\sum_{j=1}^{N} |S_j|}, j = 1, \ldots, N$, $\sum_{j=1}^{N} p_j = 1$. Then we have

$$IG(\oplus_{j=1}^{N} Y_j; X) = \sum_{j=1}^{N} p_j IG(Y_j; X) \leq max(IG(Y_1; X), \ldots, IG(Y_N; X))$$

Information Gain for MTL (Cont'd)

To prove our theorem we use the *Generalized Grouping* property of the entropy:

Lemma

For $q_{kj} \geq 0$, such that $\sum_{k=1}^{n} \sum_{j=1}^{m} q_{kj} = 1, p_k = \sum_{j=1}^{m} q_{kj}, \forall k = 1, \ldots, n$, the following holds

$$H(q_{11}, \ldots, q_{1m}, q_{21}, \ldots, q_{2m}, \ldots, q_{n1}, \ldots, q_{nm}) = \quad (1)$$

$$H(p_1, \ldots, p_n) + \sum p_k H\left(\frac{q_{k1}}{p_k}, \ldots, \frac{q_{km}}{p_k}\right), p_k > 0, \forall k. \quad (2)$$

Information Gain for MTL (Cont'd)

First, we use Lemma to develop the entropy term $H(\oplus_{j=1}^{N} Y_j)$ of the information gain

$$H(\oplus_{j=1}^{N} Y_j) = H(p_1, \ldots, p_N) + \sum_{j=1}^{N} p_j H(Y_j), \tag{3}$$

where $\sum_{j=1}^{N} p_j = 1$.

Information Gain for MTL (Cont'd)

Second, we develop the conditional entropy term as follows. We assume here that the tasks proportions are independent of the observation, i.e., $H(p_1, \ldots, p_N|x) = H(p_1, \ldots, p_N)$.

$$H(\oplus_{j=1}^{N} Y_j|X) = \sum_{x} p(x) H(\oplus_{j=1}^{N} Y_j|X = x)$$

$$= \sum_{x} p(x) \left(H(p_1, \ldots, p_N) + \sum_{j=1}^{N} p_j H(Y_j|X = x) \right)$$

$$= H(p_1, \ldots, p_N) + \sum_{j=1}^{N} p_j \sum_{x} p(x) H(Y_j|X = x)$$

$$= H(p_1, \ldots, p_N) + \sum_{j=1}^{N} p_j H(Y_j|X).$$

Information Gain for MTL (Cont'd)

Now we combine the entropy and the conditional entropy terms to evaluate the joint information gain $IG(\oplus_{j=1}^{N} Y_j; X)$. We obtain

$$
\begin{aligned}
IG(\oplus_{j=1}^{N} Y_j; X) &= H(\oplus_{j=1}^{N} Y_j) - H(\oplus_{j=1}^{N} Y_j|X) && (4)\\
&= \sum_{j=1}^{N} p_j IG(Y_j; X) && (5)\\
&\leq \sum_{j=1}^{N} p_j max(IG(Y_1; X), \ldots, IG(Y_N; X)) && (6)\\
&= max(IG(Y_1; X), \ldots, IG(Y_N; X)). && (7)
\end{aligned}
$$

This completes the proof of the theorem.

IGs Comparison

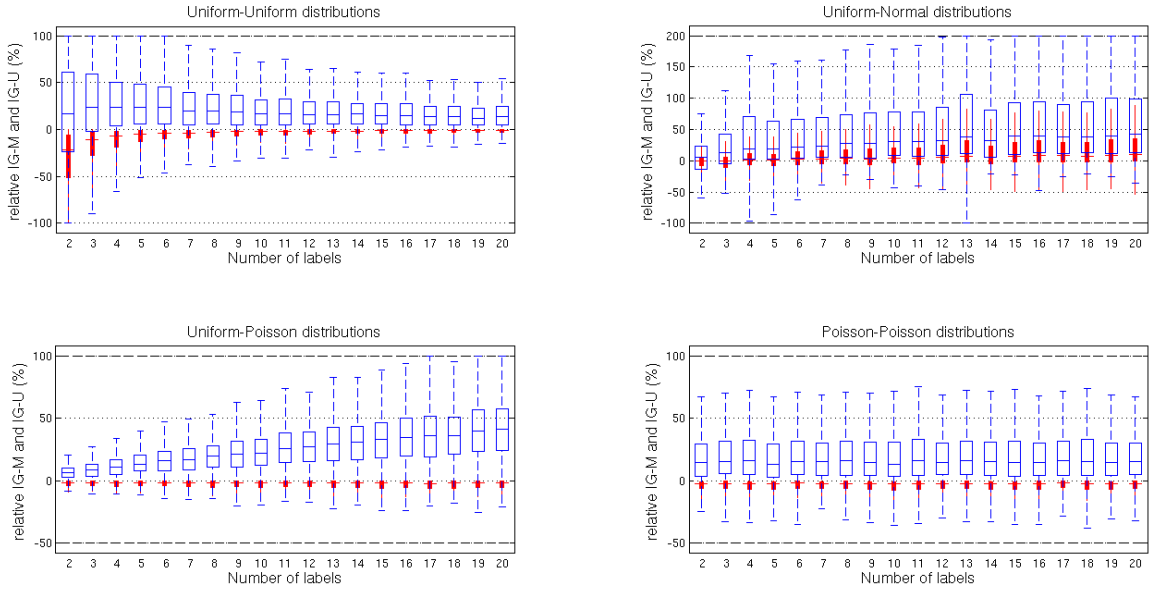


Figure: Information gain for randomly generated datasets.

Adaboost

- Adaboost is a meta-algorithm, comes from PAC framework [Valiant, 1984]
- A Weak-learner means that it is slightly better than random ($\epsilon < 0.5$)
- A Strong-learner allows to chose its error ϵ as small as wanted
- Adaboost transforms any weak learner into a strong learner!
- Head lines of the algorithm:
  1. Initialize examples weights D^0
  2. At each round $n = 1, \ldots, N$:
     - Call a weak learner on the distribution D^n, to learn a hypothesis h^n
     - Calculate ϵ^n the error of the h^n on the training set.
     - Weights of each incorrectly classified example are increased.
     - Weights of each correctly classified example are decreased.
  3. The final classifier is a weighted sum of the weak classifier's output

What is Good about Adaboost Framework ?

- Not prone to overfitting
- Can be used with many different classifiers
- Inherits the features of its weak classifier (semi-supervised, relational, statistical, etc ...)
- Simple to implement

Those properties motivate our choice of Adaboost as the framework of our multi-task algorithm

Requirements

We have to:

- Define Multi-Task-hypotheses and its learning algorithm: in our case it is MT-DT
- Modify AdaBoost for Multi-Task learning

MT-Adaboost

- We adapt *Adaboost.M1* which was introduced in [Schapire and Singer, 1999].
- Any other boosting algorithm can be used.
- In the case of multi-task, the distribution is defined over pairs of (example, task), i.e., $D \subset \mathcal{X} \times \{1, \ldots, N\}$
- The output of the algorithm is a function which takes an example as input and give a label per task as output:

$$H_j(x) = \arg\max_{y \in Y_j} \left(\sum_{i=1}^{i=T} (\ln 1/\beta_t) \right), \ 1 \leq j \leq N$$

MT-Adaboost

Require: $S = \cup_{j=1}^{N} \{e_i =< x_i, y_i, j > | \ x_i \in \mathcal{X}; \ y_i \in \mathcal{Y}_j\}$

$D_1 = \texttt{init}(S)$ *initialize distribution*
for $t = 1$ to T **do**
  $h^t = \texttt{WL}(S, D_t)$ {*train the weak learner and get an hypothesis MT-DT*}
  Calculate the error of h^t: $\epsilon_t = \sum_{j=1}^{N} \sum_{i:h_j^t(x_i) \neq y_i} D_j(x_i)$.
  if $\epsilon_t > 1/2$ **then**
    Set $T = t - 1$ and abort loop.
  end if
  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
  {*Update distribution:*}
  if $h_j^t(x_i) == y_i$ **then**
    $D_{t+1}(e_i) = \frac{D_t(e_i) \times \beta_t}{Z_t}$
  else
    $D_{t+1}(e_i) = \frac{D_t(e_i)}{Z_t}$
  end if
end for
{*Where Z_t is a normalization constant chosen so that D_{t+1} is a distribution*}
return Classifier H defined by:

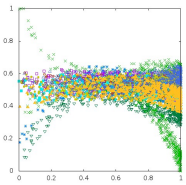
$$H_j(x) = \arg\max_{y \in \mathcal{Y}_j} \left(\sum_{i=1}^{i=T} (\ln 1/\beta_t) \right), \ 1 \leq j \leq N$$

Data Sets

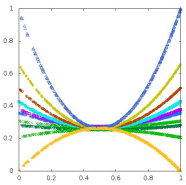
- **Synthetic**
- **Enron**
- **ECML'06 spam filtering challenge**

- In all experiments we use three random shuffles of 5-fold cross validation.
- Classification accuracy per task is reported.

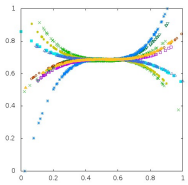
Synthetic Data Sets



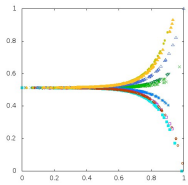
(a) A correlation pattern from beta-cubic distributions



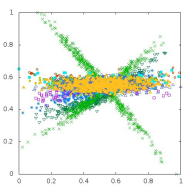
(b) A correlation pattern from beta-quadratic distributions



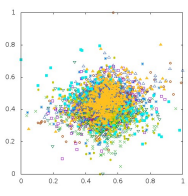
(c) A correlation pattern from gaussian-cubic distributions



(d) A correlation pattern from gaussian-exponential distributions



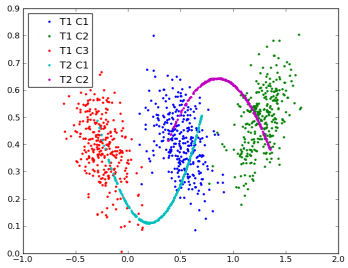
(e) A correlation pattern from gaussian-quadratic distributions



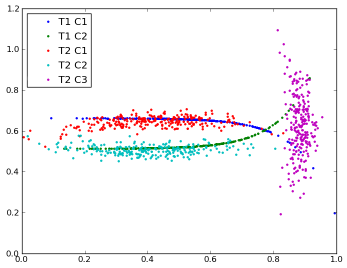
(f) A correlation pattern from laplace-linear distributions

Figure: Task Data Sets: Patterns for synthetic 2D data

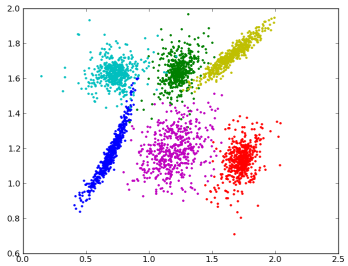
Synthetic Data Sets



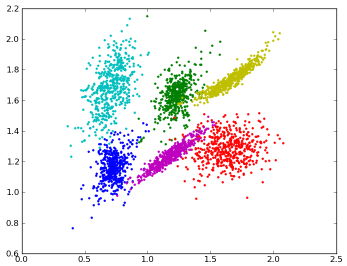
(a) Two related multi-class tasks



(b) Two related multi-class tasks



(c) A multi-class task



(d) A multi-class task

Results

Single Task Algorithms					
	AMH		M1C45	RF	
T1	71.86 ± 4.45		90.75 ± 0.08	87.88 ± 0.45	
T2	67.27 ± 5.96		83.74 ± 0.55	87.64 ± 0.23	
Avg	69.57		87.24	87.76	
Multi Task Learning with 2T-stumps and MT-DTs					
	MTMH NB	MTMHN NBPT	MTM1 IG_J	MTM1 IG_U	MT IG_M
T1	90.17 ± 0.17	90.51 ± 0.07	87.97 ± 0.80	89.88 ± 0.06	90.77 ± 0.07
T2	88.70 ± 0.77	88.57 ± 0.64	88.45 ± 1.56	88.58 ± 1.50	88.371 ± 0.26
Avg	89.44	89.54	88.21	89.23	89.57
MT-DTs with Random Forest					
	MTRF IG_J		MTRF IG_U	MTRF IG_M	
T1	88.33 ± 0.46		87.59 ± 0.61	87.75 ± 0.43	
T2	88.14 ± 0.53		88.61 ± 0.40	88.20 ± 0.37	
Avg	88.24		88.10	87.97	

Table: Comparison between all single task and multi-task algorithms on the first DS_1 synthetic dataset in previous slide. MH: AdaboostMH, M1C45: Adaboost.M1 /w C45 trees, RF: random forest, MTMH NB: MT-Adaboost.MH /w N-best 2T-stump, MTMH NBPT: MT-Adaboost.MH /w N-best per task, MTM1 IG_x: MT-Adaboost with MT-DT and IG_x as criterion.

ENRON [Cohen, 2004]

- contains all e-mails sent and received by some 150 accounts of the top management of Enron.
- Textual features we use are 2200, each of which represents a cluster of semantically similar words.
- Common **social features** are used.
- Two Tasks:
  - **Responsive/ Non Responsive**: Responsive emails published by the Department of Justice, they concern relevant emails to trials against two Enron's CEOs.
  - **5-Topics:** Topics extracted from Berkeley annotation.

Results

Tasks	Train (Test)	C4.5	IG_J	IG_U	IG_M
Responsive Vs. NonResponsive	299 (74)	80.32 $\pm$ 1.87	80.59 $\pm$ 2.23	80.01 $\pm$ 3.11	**81.81$\pm$1.16**
5 Topics	265 (66)	43.12 $\pm$ 1.03	43.65 $\pm$ 1.77	44.12 $\pm$ 0.42	**48.11$\pm$0.023**
Avg		61.72	62.12	62.066	**64.96**

Table: Average classification accuracy on Enron tasks.

Tasks	Train (Test)	Adaboost C4.5	MT-Adaboost IG_J	MT-Adaboost IG_U	MT-Adaboost IG_M
Responsive Vs. NonResponsive	299 (74)	85.10 $\pm$ 1.21	84.66 $\pm$ 2.15	84.52 $\pm$ 1.2	**86.01$\pm$1.53**
5 Topics	265 (66)	51.34 $\pm$ 0.43	52.89 $\pm$ 0.87	52.17 $\pm$ 0.74	**57.11$\pm$0.02**
Avg		68.22	68.78	68.35	**71.65**

Table: Average classification accuracy of boosted trees on Enron tasks.

ECML'06 Challenge

- It was used for the ECML/PKDD 2006 discovery challenge.
- It contains email in-boxes of 15 users. Each in-box has 400 spam/ham emails.
- They are encoded by standard bag-of-word vector representation.
- We consider each user as a task, the experiments are done on the first three users.

Results

Tasks	Train (Test)	C4.5	IG_J	IG_U	IG_M
User-1	320 (80)	86.45 ± 1.23	86.19 ± 1.14	86.00 ± 1.88	**87.65±3.42**
User-2	320 (80)	85.13 ± 2.16	85.53 ± 2.22	85.07 ± 3.16	**88.93±3.44**
User-3	320 (80)	88.03 ± 2.11	88.22 ± 2.56	**88.52±1.33**	88.19 ± 2.51
Avg		86.54	86.65	86.53	**88.26**

Table: Average classification accuracy on three ECML'06 user inboxes.

Results

- Results showed a superiority of IG_M over other MT-DT criteria in accuracy values.
- Learning tasks simultaneously does not bring the same improvement to all tasks.
- More difficult tasks (tasks with a lower accuracy) have a larger margin of improvement.

Conclusion

We proposed an adaptation of decision tree learning to the multi-task learning, with the important contribution is three-fold:

- We proposed `MT-DT` to deal with multi-class tasks, where tasks might have different number of classes.
- We derived a new information gain measure for decision trees in the multi-task setting.
- We modified MT-Adaboost to enable it for multi-class problems.

Thanks You !
Questions ?

Introduction Learning MT-DTs MT-Adaboost Experiments Conclusion


Cohen, W. (2004).
Enron.


Faddoul, J. B., Chidlovskii, B., Torre, F., and Gilleron, R. (2010).
Boosting multi-task weak learners with applications to textual and social data.
In *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA)*, pages 367–372.


Quadrianto, N., Smola, A., Caetano, T., Vishwanathan, S., and Petterson, J. (2010).
Multitask learning without label correspondences.
In *Proceedings of the Twenty-Fourth Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1957–1965.


Schapire, R. E. and Singer, Y. (1999).

Improved boosting algorithms using confidence-rated predictions.
Machine Learning, 37.


Valiant, L. G. (1984).
A theory of the learnable.
Communications of the ACM, 27:1134–1142.