Supplementary material: Curriculum Learning of Multiple Tasks

Anastasia Pentina, Viktoriia Sharmanska, Christoph H. Lampert
IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria
{apentina, vsharman, chl}@ist.ac.at

1. Proof of Theorem 1

We apply PAC-Bayesian theory to prove a generalization bound for the case of sequential task solving. For more details on it see [1, 6, 9].

Assume that the learner observes a sequence of tasks in a fixed order, $t_1, ..., t_n$, with corresponding training sets, $S_1, ..., S_n$, where $S_i = \{(x_1^i, y_1^i), ..., (x_{m_i}^i, y_{m_i}^i)\}$ consists of m_i i.i.d. samples from a task-specific data distribution D_i. We assume that all tasks share the same input space $\mathcal{X}$ and output space $\mathcal{Y}$ and that the learner uses the same loss function $l : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ and hypothesis set $H \subset \{h : \mathcal{X} \to \mathcal{Y}\}$ for solving these tasks. The learner solves only one task at a time by using some arbitrary but fixed deterministic algorithm $\mathcal{A}$ that produces a posterior distribution Q_i over H based on training data S_i and some prior knowledge P_i, which is also expressed in form of probability distribution over the hypothesis set. Moreover, we assume that the solution Q_i plays the role of a prior for the next task, i.e. $P_{i+1} = Q_i$ (P_1 is just some fixed distribution, Q_0). For making predictions for task t_i the learner uses the Gibbs predictor, associated with the corresponding posterior distribution Q_i. For an input $x \in \mathcal{X}$ this randomized predictor samples $h \in H$ according to Q_i and returns $h(x)$. The goal of the learner is to perform well on all tasks, $t_1, ..., t_n$, i.e. to minimize the average expected error of the Gibbs classifiers defined by $Q_1, \dots, Q_n$:

$$\mathrm{er} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{er}_i(Q_i(Q_{i-1}, S_i)) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}_{(x,y)\sim D_i}\mathbf{E}_{h\sim Q_i} l(h(x), y). \tag{1}$$

Since the data distributions of the tasks $t_1, ..., t_n$ are unknown, one can not directly compute (1). However, it can be approximated by the empirical error based on the observed data:

$$\widehat{\mathrm{er}} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\mathrm{er}}_i(Q_i(Q_{i-1}, S_i)) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\sum_{j=1}^{m_i} \mathbf{E}_{h\sim Q_i} l(h(x_j^i), y_j^i). \tag{2}$$

The following theorem provides an upper bound on the difference between the two quantities (1) and (2):

Theorem 2. *For any fixed distribution Q_0, learning algorithm $\mathcal{A}$ and any $\delta > 0$ the following inequality holds with probability at least $1 - \delta$ (over sampling the training sets $S_1, ..., S_n$):*

$$\mathrm{er} \leq \widehat{\mathrm{er}} + \frac{1}{n\sqrt{\bar{m}}} \mathrm{KL}\left(Q_1 \times \cdots \times Q_n || Q_0 \times \cdots \times Q_{n-1}\right) + \frac{1}{8\sqrt{\bar{m}}} - \frac{\log \delta}{n\sqrt{\bar{m}}}, \tag{3}$$

where $Q_i = \mathcal{A}(Q_{i-1}, S_i)$ is a posterior distribution for the task t_i learned by $\mathcal{A}$ based on Q_{i-1} and S_i, $\bar{m} = \left(\frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\right)^{-1}$ is the harmonic mean of the sample sizes and KL *denotes Kullback-Leibler divergence.*

Proof. First we use Donsker-Varadhan's variational formula [10] to change the expectation over posteriors $(Q_1, ..., Q_n)$ to the expectation over priors $(Q_0, Q_1, ..., Q_{n-1})$:

$$\mathrm{er} - \widehat{\mathrm{er}} \leq \frac{1}{\lambda}\Big(\mathrm{KL}\left(Q_1 \times \cdots \times Q_n || Q_0 \times \cdots \times Q_{n-1}\right) + \log \mathop{\mathbf{E}}_{h_1\sim Q_0} \cdots \mathop{\mathbf{E}}_{h_n\sim Q_{n-1}} \exp\Big(\frac{\lambda}{n}\sum_{i=1}^{n}(\mathrm{er}_i(h_i) - \widehat{\mathrm{er}}_i(h_i))\Big)\Big), \tag{4}$$

where $\mathrm{er}_i(h)$ is the expected loss of a hypothesis h computed with respect to the data distribution of task t_i and $\widehat{\mathrm{er}}_i(h)$ is the corresponding empirical loss, computed on S_i. This inequality holds for any $\lambda > 0$.

Note, that Q_i may depend on $S_1, ..., S_i$, but does not depend on $S_{i+1}, ..., S_n$. Therefore:

$$\mathop{\mathbf{E}}_{S_1 \cdots S_n} \mathop{\mathbf{E}}_{h_1 \sim Q_0} \cdots \mathop{\mathbf{E}}_{h_n \sim Q_{n-1}} \exp\left(\frac{\lambda}{n}\sum_{i=1}^{n}(\mathrm{er}_i(h_i) - \hat{\mathrm{er}}_i(h_i))\right) = \mathop{\mathbf{E}}_{h_1 \sim Q_0} \mathop{\mathbf{E}}_{S_1} \exp\left(\frac{\lambda}{n}(\mathrm{er}_1(h_1) - \hat{\mathrm{er}}_1(h_1))\right) \cdots \mathop{\mathbf{E}}_{h_n \sim Q_{n-1}} \mathop{\mathbf{E}}_{S_n} \exp\left(\frac{\lambda}{n}(\mathrm{er}_n(h_n) - \hat{\mathrm{er}}_n(h_n))\right). \quad (5)$$

We fix $h_n \in H$. Then we can rewrite the last term of (5) in the following way:

$$\exp\left(\frac{\lambda}{n}(\mathrm{er}_n(h_n) - \hat{\mathrm{er}}_n(h_n))\right) = \prod_{j=1}^{m_n} \exp\left(\frac{\lambda}{nm_n}\left(\mathrm{er}_n(h_n) - l(h_n(x_j^n), y_j^n)\right)\right). \quad (6)$$

Since the data points in S_n are i.i.d., all terms in this product are independent and take values between $\frac{\lambda(\mathrm{er}_n(h_n)-1)}{nm_n}$ and $\frac{\lambda\,\mathrm{er}_n(h_n)}{nm_n}$. Therefore, by Hoeffding's lemma [4], we obtain that the last term of (5) is bounded by a constant:

$$\mathop{\mathbf{E}}_{h_n \sim Q_{n-1}} \mathop{\mathbf{E}}_{S_n} \exp\left(\frac{\lambda}{n}(\mathrm{er}_n(h_n) - \hat{\mathrm{er}}_n(h_n))\right) \le \exp\left(\frac{\lambda^2}{8n^2 m_n}\right)$$

We repeat the same procedure for all other tasks and obtain that:

$$\mathop{\mathbf{E}}_{S_1 \dots S_n} \mathop{\mathbf{E}}_{h_1 \sim Q_0} \cdots \mathop{\mathbf{E}}_{h_n \sim Q_{n-1}} \exp\left(\frac{\lambda}{n}\sum_{i=1}^{n}(er_i(h_i) - \hat{\mathrm{er}}_i(h_i))\right) \le \exp\left(\frac{\lambda^2}{8n\bar{m}}\right), \quad (7)$$

where $\bar{m} = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\right)^{-1}$. Therefore, by Markov's inequality, with probability at least $1-\delta$:

$$\mathop{\mathbf{E}}_{h_1 \sim Q_0} \cdots \mathop{\mathbf{E}}_{h_n \sim Q_{n-1}} \exp\left(\frac{\lambda}{n}\sum_{i=1}^{n}(er_i(h_i) - \hat{\mathrm{er}}_i(h_i))\right) \le \frac{1}{\delta}\exp\left(\frac{\lambda^2}{8n\bar{m}}\right). \quad (8)$$

By combining (8) with (4) we get:

$$\mathrm{er} \le \hat{\mathrm{er}} + \frac{1}{\lambda}\mathrm{KL}\left(Q_1 \times \cdots \times Q_n || Q_0 \times \cdots \times Q_{n-1}\right) + \frac{\lambda}{8n\bar{m}} - \frac{1}{\lambda}\log\delta. \quad (9)$$

By setting $\lambda = n\sqrt{\bar{m}}$ we obtain the final result. □

Theorem 2 holds only for tasks that are given to the learner in an arbitrary but fixed order, which must be chosen before observing the sample sets $S_1, \dots, S_n$. We can, however, extend it to hold uniformly for all orders of tasks: for each possible task order, $\pi \in \mathcal{S}_n$, where $\mathcal{S}_n$ is the *symmetric group*, we use (3) with confidence parameter $\delta/n!$. We then combine all inequalities (of which there are $n!$ many) using the union bound, thereby obtaining the following generalization:

Theorem 3. *For any fixed distribution Q_0, any learning algorithm $\mathcal{A}$ and any $\delta > 0$ with probability at least $1-\delta$ (over sampling the training sets $S_1, ..., S_n$) the following inequality holds uniformly for any order $\pi \in \mathcal{S}_n$:*

$$\mathrm{er} \le \hat{\mathrm{er}} + \frac{1}{8\sqrt{\bar{m}}} + \frac{\log n}{\sqrt{\bar{m}}} - \frac{\log\delta}{n\sqrt{\bar{m}}} + \frac{1}{n\sqrt{\bar{m}}}\mathrm{KL}\left(Q_{\pi(1)} \times \cdots \times Q_{\pi(n)} || Q_0 \times \cdots \times Q_{\pi(n-1)}\right), \quad (10)$$

where $Q_{\pi(i)} = \mathcal{A}(Q_{\pi(i-1)}, S_{\pi(i)})$, $\bar{m} = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\right)^{-1}$ and $\pi(0) = 0$.

Theorem 1 is an instantiation of Theorem 3 for the special case of binary classification using linear predictors. Assume $\mathcal{Y} = \{+1, -1\}$, $\mathcal{X} \in \mathbb{R}^d$, and let H be a set of linear predictors $\{\mathrm{sign}\langle w, x\rangle\}$, where $w \in \mathbb{R}^d$ is a weight vector. We also assume that the learner uses $0/1$ loss, $l(y_1, y_2) = [\![y_1 \neq y_2]\!]$. In this case the expected error of the Gibbs predictor is at least half the expected error of the corresponding majority vote predictor [8]. Therefore, by multiplying the right hand side of (10) by a factor of 2, one obtains a generalization bound for deterministic majority vote classifier.

The case of linear predictors can be captured by the PAC-Bayesian setting if prior and posterior distributions are Gaussian [3]. More formally, assume that $Q_i = \mathcal{N}(w_i, Id)$ for $i = 0, ..., n$, i.e. Gaussian distributions with unit variance that differ only by the value of their mean vectors. Due to the symmetry of the Gaussian distribution, the predictor defined by w_i is equivalent to the majority vote predictor corresponding to distribution Q_i. Hence one can use the result of Theorem 3 in the case of deterministic linear predictors. We also assume that the learner uses an algorithm, $\mathcal{A}$, that for every task t_i returns w_i based on the mean vector of the used prior distribution and training data S_i.

By computing the complexity term from (10) we obtain:

$$\mathrm{KL}(Q_{\pi(1)} \times \cdots \times Q_{\pi(n)} || Q_0 \times \cdots \times Q_{\pi(n-1)}) = \sum_{i=1}^{n}\mathrm{KL}(Q_{\pi(i)} || Q_{\pi(i-1)}) = \sum_{i=1}^{n}\frac{||w_{\pi(i)} - w_{\pi(i-1)}||^2}{2}, \quad (11)$$

where $\pi(0) = 0$, $w_0 = \mathbf{0}$ and $w_{\pi(i)} = \mathcal{A}(w_{\pi(i-1)}, S_{\pi(i)})$. Note that the loss of the Gibbs classifier defined by Q_i

on a point (x,y) is given by $\bar{\Phi}\left(\frac{yx^T w_i}{||x||}\right)$, where $\bar{\Phi}(z) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)\right)$ and $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ is the Gauss error function [2, 7]. Together with (11) it gives us the result of Theorem 1.

2. Learning with multiple subsequences

Assume that, as in the case of learning in a fixed order described in Theorem 2, n tasks $t_1, ..., t_n$ are processed one after another from t_1 till t_n. We extend the sequential learning scenario by allowing the learner to not transfer information between some of the subsequent task. Specifically, if the posterior distribution Q_i obtained for task t_i is not informative with respect to the next task, t_{i+1}, the learner may use original, fixed distribution Q_0 as a prior for t_{i+1} instead of Q_i. Such scenario can be described by introducing the set of flags $b_i \in \{0,1\}$ for $i = 2, ..., n$, where $b_i = 1$ means that information from task t_{i-1} is transferred to the task t_i, in other words Q_{i-1} is used as a prior for solving t_i, while $b_i = 0$ denotes that there is no transfer from t_{i-1} to t_i and Q_0 is used as a prior P_i.

In the same manner, as we proved Theorem 2, we can prove the following generalization bound for the case of sequential learning with ability to not transfer information between subsequent tasks:

Theorem 4. *For any fixed distribution Q_0, set of flags $b_i \in \{0,1\}$ for $i = 2, ..., n$, learning algorithm $\mathcal{A}$ and any $\delta > 0$ the following inequality holds with probability at least $1-\delta$ (over sampling the training sets $S_1, ..., S_n$):*

$$\begin{aligned}\text{er} \leq \widehat{\text{er}} &+ \frac{1}{n\sqrt{\bar{m}}}\text{KL}\left(Q_1 \times \cdots \times Q_n || P_1 \times \cdots \times P_n\right) \\ &+ \frac{1}{8\sqrt{\bar{m}}} - \frac{\log\delta}{n\sqrt{\bar{m}}},\end{aligned} \tag{12}$$

where:

$$\begin{aligned} P_i &= \begin{cases} Q_0 & \text{if } i = 1 \text{ or } b_i = 0 \\ Q_{i-1} & \text{if } b_i = 1 \end{cases} \\ Q_i &= \mathcal{A}(P_i, S_i) \\ \bar{m} &= \left(\frac{1}{n}\sum_{i=1}^n \frac{1}{m_i}\right)^{-1}. \end{aligned}$$

The result of Theorem 4 holds for any, but fixed in advance order of tasks and set of flags b_i. Now, we can extend it to hold uniformly for all possible partitions of tasks in subsequences and orders of tasks in each group. First, note that there are $n! \leq n^n$ possible full orderings of n tasks. Second, there are 2^{n-1} possible ways to define flags b_i for each task. Therefore there are less than $n^n 2^{n-1}$ possible partitions of tasks into groups and orderings inside each group. We now let the confidence parameter to be $\delta/((2n)^n)$ and combine inequalities for all possible partitions and orderings (of which there are less than $(2n)^n$ many) using the union bound argument. Thereby we obtain the following result:

Theorem 5. *For any fixed distribution Q_0, learning algorithm $\mathcal{A}$ and any $\delta > 0$ with probability at least $1-\delta$ (over sampling the training sets $S_1, ..., S_n$) the following inequality holds uniformly for all orders $\pi \in \mathcal{S}$ and all set of flags $\{b_2, ..., b_n\} \in \{0,1\}^{n-1}$:*

$$\begin{aligned}\text{er} \leq \widehat{\text{er}} + \frac{1}{8\sqrt{\bar{m}}} + \frac{\log 2n}{\sqrt{\bar{m}}} - \frac{\log\delta}{n\sqrt{\bar{m}}} + \\ \frac{1}{n\sqrt{\bar{m}}}\text{KL}\left(Q_{\pi(1)} \times \cdots \times Q_{\pi(n)} || P_{\pi(1)} \times \cdots \times P_{\pi(n)}\right),\end{aligned} \tag{13}$$

where:

$$\begin{aligned} P_{\pi(i)} &= \begin{cases} Q_0 & \text{if } i = 1 \text{ or } b_i = 0 \\ Q_{\pi(i-1)} & \text{if } b_i = 1 \end{cases} \\ Q_{\pi(i)} &= \mathcal{A}(P_{\pi(i)}, S_{\pi(i)}) \\ \bar{m} &= \left(\frac{1}{n}\sum_{i=1}^n \frac{1}{m_i}\right)^{-1}. \end{aligned}$$

We can formulate the instantiation of Theorem 5 for the case of linear predictors and 0/1 loss using Gaussian distributions as we did for proving Theorem 1 based on Theorem 3. As a result, we obtain the following generalization bound:

Theorem 6. *For any deterministic learning algorithm $\mathcal{A}$ and any $\delta > 0$, the following holds with probability at least $1-\delta$ over sampling the training sets $S_1, ..., S_n$ uniformly for any order π in the symmetric group $\mathcal{S}_n$ and any set of flags $\{b_2, ..., b_n\} \in \{0,1\}^{n-1}$:*

$$\begin{aligned}&\frac{1}{2n}\sum_{i=1}^n \mathop{\mathbf{E}}_{(x,y)\sim D_i} [\![y \neq \text{sign}\langle w_i, x\rangle]\!] \leq \\ &\frac{1}{n}\sum_{i=1}^n \left[\frac{1}{m_{\pi(i)}}\sum_{j=1}^{m_{\pi(i)}} \bar{\Phi}\left(\frac{y_j^{\pi(i)}\langle w_{\pi(i)}, x_j^{\pi(i)}\rangle}{||x_j^{\pi(i)}||}\right) + \right. \\ &\left. \frac{||w_{\pi(i)} - b_i w_{\pi(i-1)}||^2}{2\sqrt{\bar{m}}}\right] + \frac{1}{8\sqrt{\bar{m}}} - \frac{\log\delta}{n\sqrt{\bar{m}}} + \frac{\log 2n}{\sqrt{\bar{m}}},\end{aligned} \tag{14}$$

where:

$$
\begin{aligned}
w_{\pi(i)} &= \begin{cases} \mathcal{A}(\mathbf{0}, S_{\pi(i)}) & \textit{if } i=1 \ \textit{ or } \ b_i = 0 \\ \mathcal{A}(w_{\pi(i-1)}, S_{\pi(i)}) & \textit{otherwise} \end{cases} \\
\bar{\Phi}(z) &= \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)\right) \\
\operatorname{erf}(z) &= \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \\
\bar{m} &= \left(\frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\right)^{-1}.
\end{aligned}
$$

Algorithm 2 MultiSeqMT: Sequential Learning with Multiple Subsequences

---

```
Input S_1, ..., S_n {training sets}
T ← {1, 2, ..., n} {indices of yet unused tasks}
P ← {0} {ws of the last tasks in the existing subseq.}
for i = 1 to n do
    for all w̃ ∈ P do
        k(w̃) ← steps 5-8 of Algorithm 1 with
                substituting w_π(i−1) by w̃ in (4)
    end for
    w* ← minimizer of (4) w.r.t. w̃ with substituting
            w_π(i−1) by w̃ and k by k(w̃)
    w_k(w*) ← solution of (2) using S_k(w*) and
                w* instead of w̃
    T ← T \ {k(w*)}
    P ← P ∪ {w_k(w*)}
    if w* ≠ 0 then
        P ← P \ {w*}
    end if
end for
Return w_1, ..., w_n
```

References

[1] O. Catoni. *PAC-Bayesian Supervised Classification (The Thermodynamics of Statistical Learning)*. Institute of Mathematical Statistics, 2007. 1

[2] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learing (ICML)*, 2009. 3

[3] R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In *Conference on Neural Information Processing Systems (NIPS)*, 2001. 2

[4] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963. 2

[5] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image Search with Relative Attribute Feedback. In *Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[6] J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research (JMLR)*, pages 273–306, 2005. 1

[7] J. Langford and J. Shawe-Taylor. PAC-Bayes and margins. In *Conference on Neural Information Processing Systems (NIPS)*, 2002. 3

[8] D. McAllester. Simplified PAC-Bayesian margin bounds. In *Learning Theory and Kernel Machines*. 2003. 2

[9] M. W. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalization Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003. 1

[10] Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 2012. 1

	Chimpanzee	Giant panda	Leopard	Persian cat	Hippopotamus	Raccoon	Rat	Seal
IndSVM	26.34 ± 0.31	24.12 ± 0.58	20.60 ± 0.27	25.90 ± 0.45	29.60 ± 0.49	31.07 ± 0.36	39.66 ± 0.66	28.98 ± 0.30
MergedSVM	22.81 ± 0.31	19.08 ± 0.47	20.65 ± 0.35	24.02 ± 0.44	28.31 ± 0.48	29.40 ± 0.57	36.66 ± 0.62	27.60 ± 0.33
MT	24.16 ± 0.35	20.12 ± 0.45	19.71 ± 0.33	23.99 ± 0.40	27.94 ± 0.52	29.25 ± 0.43	37.41 ± 0.60	27.65 ± 0.29
SeqMT(ours)	23.86 ± 0.33	19.33 ± 0.52	19.36 ± 0.29	23.81 ± 0.40	27.83 ± 0.46	29.04 ± 0.37	36.34 ± 0.57	27.04 ± 0.22
Max	24.56 ± 0.37	20.66 ± 0.49	20.63 ± 0.32	25.28 ± 0.34	29.59 ± 0.49	30.38 ± 0.51	38.03 ± 0.58	28.27 ± 0.37
Error	24.47 ± 0.42	20.02 ± 0.58	19.97 ± 0.27	24.84 ± 0.46	29.07 ± 0.55	29.75 ± 0.31	38.00 ± 0.54	28.27 ± 0.38
Reg	23.94 ± 0.32	19.44 ± 0.50	19.36 ± 0.29	23.81 ± 0.40	27.83 ± 0.46	29.04 ± 0.37	36.34 ± 0.57	27.04 ± 0.22
Random	24.18 ± 0.37	20.44 ± 0.46	20.06 ± 0.33	24.41 ± 0.37	28.66 ± 0.55	29.95 ± 0.48	37.40 ± 0.66	27.84 ± 0.27
Semantic	23.62 ± 0.32	19.07 ± 0.51	19.67 ± 0.30	24.03 ± 0.37	28.67 ± 0.47	29.00 ± 0.43	37.23 ± 0.54	28.09 ± 0.36
Best	23.35 ± 0.38	19.07 ± 0.51	19.22 ± 0.30	23.69 ± 0.46	27.79 ± 0.33	28.82 ± 0.46	36.57 ± 0.63	27.46 ± 0.37
Worst	24.89 ± 0.40	21.18 ± 0.48	20.58 ± 0.32	25.20 ± 0.39	29.19 ± 0.47	30.32 ± 0.51	38.74 ± 0.65	28.16 ± 0.28

Table 1. Sequential learning of tasks from easiest to hardest in the AwA dataset. For each class and method, the numbers are average error rate and standard error of the mean over 20 repeats.

Attribute/Class	Athletic	Boots	Clogs	Flats	Heels	Pumps	Rain Boots	Sneakers	Stiletto	Wedding
Pointy at the front	2	6	3	5	10	9	4	1	8	7
Open	3	2	8	5	7	6	1	4	9	10
Bright in color	6	1	2	8	4	3	10	7	9	5
Covered with ornaments	4	9	6	5	8	7	1	3	10	2
Shiny	2	9	4	3	6	5	8	1	10	7
High at the heel	4	6	5	1	9	8	3	2	10	7
Long on the leg	7	9	2	3	6	5	10	8	4	1
Formal	3	6	4	7	9	8	1	2	5	10
Sporty	10	5	6	7	4	3	8	9	1	2
Feminine	1	6	4	5	10	9	3	2	8	7

Table 2. Ordering of classes with respect to attributes in the *Shoes* dataset [5]. Cells, coloured in blue, represent classes that were used as negative examples and the ones coloured in yellow represent the ones used as positive examples for the corresponding attribute.