Easy Questions First?
A Case Study on Curriculum Learning for Question Answering

Mrinmaya Sachan **Eric P. Xing**
School of Computer Science
Carnegie Mellon University
{mrinmays, epxing}@cs.cmu.edu

Abstract

Cognitive science researchers have emphasized the importance of ordering a complex task into a sequence of easy to hard problems. Such an ordering provides an easier path to learning and increases the speed of acquisition of the task compared to conventional learning. Recent works in machine learning have explored a curriculum learning approach called self-paced learning which orders data samples on the easiness scale so that easy samples can be introduced to the learning algorithm first and harder samples can be introduced successively. We introduce a number of heuristics that improve upon self-paced learning. Then, we argue that incorporating easy, yet, a diverse set of samples can further improve learning. We compare these curriculum learning proposals in the context of four non-convex models for QA and show that they lead to real improvements in each of them.

1 Introduction

A key challenge in building an intelligent agent is in modeling the incrementality and the cumulative nature of human learning (Skinner, 1958; Peterson, 2004; Krueger and Dayan, 2009). Children typically learn grade by grade, progressing from simple concepts to more complex ones. Given a complex set of concepts, it is often the case that some concepts are easier than others. Some concepts can even be prerequisite to learning other concepts. Hence, evolving a useful curriculum where easy concepts are presented first and more complex concepts are gradually introduced can be beneficial for learning.

We explore methods for learning a curriculum in the context of non-convex models for question answering. Curriculum learning (CL) (Bengio et al., 2009) and self-paced learning (SPL) (Kumar et al., 2010) have been recently introduced in machine learning literature. However, their usefulness in the context of NLP tasks such as QA has not been studied so far. The main challenge in learning a curriculum is that it requires the identification of easy and hard concepts in the given training dataset. However, in real-world applications, such a ranking of training samples is difficult to obtain. Furthermore, a human judgement of 'easiness' of a task might not correlate with what is easy for the algorithm in the feature and hypothesis space employed for the given application. SPL combines the selection of the curriculum and the learning task in a single objective. The easiness of a question in self-paced learning is defined by its local loss. We propose and study other heuristics that define a measure of easiness and learn the curriculum by selecting samples using this measure. These heuristics are similar to those used in active learning, but with one key difference. In curriculum learning, all the training examples and labels are already known, which is not the case in active learning. Our experiments show that these heuristics work well in practice.

While the strategy of learning from easy questions first and then gradually handling harder questions is supported by many cognitive scientists, others (Cantor, 1946) argue that it is also important to expose the learning to diverse (even if sometimes harder) examples. We argue that the right curriculum should not only be arranged in the increasing order of difficulty but also introduce the learner to sufficient number of diverse examples that are sufficiently dissimilar from what has already been introduced to the learning process. We showed that the above heuristics when coupled with diversity lead to significant improvements.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 453–463,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

We provide empirical evaluation on four QA models: (a) an alignment-based approach (Sachan et al., 2015) for machine comprehension – a reading comprehension task (Richardson et al., 2013) with a set of questions and associated texts, (b) an alignment-based approach (Sachan et al., 2016) for a multiple-choice elementary science test (Clark and Etzioni, 2016), (c) *QANTA* (Iyyer et al., 2014) – a recursive neural network for answering quiz bowl questions, and (d) memory networks (Weston et al., 2014) – a recurrent neural network with a long-term memory component for answering 20 pre-defined tasks for machine comprehension. We show value in our approaches for curriculum learning on all these settings.

Our paper has the following contributions:

1. In our knowledge, this is the first application of curriculum learning to the task of QA and one of the first in NLP. We hope to make the NLP and ML communities aware of the benefits of CL for non-convex optimization.
2. We perform an in-depth analysis of SPL, and propose heuristics which offer significant improvements over SPL; the state-of-the-art in curriculum learning.
3. We stress on diversity of questions in the curriculum during learning and propose a method that learns a curriculum while capturing diversity to gain more improvements.

2 Problem Setting for QA

For each question $q_i \in Q$, let $A_i = \{a_{i1}, \ldots, a_{im}\}$ be the set of candidate answers to the question. Let a_i^* be the correct answer. The candidate answers may be pre-defined, as in multiple-choice QA, or may be undefined but easy to extract with a high degree of confidence (e.g., by using a pre-existing system). We want to learn a function $f : (q, \mathcal{K}) \to a$ that, given a question q_i and background knowledge $\mathcal{K}$ (texts/resources required to answer the question), outputs an answer $\hat{a}_i \in A_i$. We consider a scoring function $S_{\mathbf{w}}(q, a; \mathcal{K})$ (with model parameters $\mathbf{w}$) and a prediction rule $f_{\mathbf{w}}(q_i) = \hat{a}_i = \underset{a_{ij} \in A_i}{\arg\max}\, S_{\mathbf{w}}(q_i, a_{ij}; \mathcal{K})$. Let $\Delta(\hat{a}_i, a_i^*)$ be the cost of giving a wrong answer. We consider the empirical risk minimization (ERM) framework given a loss function L and a regularizer Ω:

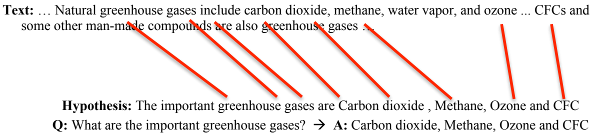
$$\min_{\mathbf{w}} \sum_{q_i \in Q} L_{\mathbf{w}}(a_i^*, f_{\mathbf{w}}(q_i); \mathcal{K}) + \Omega(\mathbf{w}) \qquad (1)$$

3 QA Models

The field of QA is quite rich. Solutions proposed have ranged from various IR based approaches that treat this as a problem of retrieval from existing knowledge bases or perform inference using a large corpus of unstructured texts by learning a similarity between the question and a set of candidate answers (Yih et al., 2013). A comprehensive review of QA is out of scope of this paper. So we point the interested readers to Jurafsky and Martin (2000), chapter 28 for a more comprehensive review. In this paper, we will explore curriculum learning in the context of non-convex models for QA. The models will be (1) latent structural SVM (Yu and Joachims, 2009) based solutions for standardized question-answering tests and (2) deep learning models (Iyyer et al., 2014; Weston et al., 2014) for QA.

Recently, researchers have proposed standardized tests as 'drivers for progress in AI' (Clark and Etzioni, 2016). Some example standardized tests are reading comprehensions (Richardson et al., 2013), algebra word problems (Kushman et al., 2014), geometry problems (Seo et al., 2014), entrance exams (Fujita et al., 2014; Arai and Matsuzaki, 2014), etc. These tests are usually in the form of question-answers and focus on elementary learning. The idea of learning the curriculum could be especially useful in the context of standardized tests. Standardized tests (Clark and Etzioni, 2016) are implicitly incremental in nature, covering various levels of difficulty. Thus they are rich sources of data for building systems that learn incrementally. These datasets can also help us understand the *shaping* hypothesis as we can use them to verify if easier questions are indeed getting picked by our incremental learning algorithm before harder questions.

On the other hand, deep learning models (LeCun et al., 2015) have recently shown good performance in many standard NLP and vision tasks, including QA. These models usually learn representations of data and the QA model jointly. The models use a cascade of many layers of nonlinear processing units, leading to a highly non-convex model and a large parameter space. This renders these models susceptible to local-minima. Hence, the idea of learning the curricula is also very useful in the context of deep-learning models, as the technique of processing questions in the increasing order of difficulty often leads to better minima



Text: … Natural greenhouse gases include carbon dioxide, methane, water vapor, and ozone ... CFCs and some other man-made compounds are also greenhouse gases …

Hypothesis: The important greenhouse gases are Carbon dioxide , Methane, Ozone and CFC

Q: What are the important greenhouse gases? → **A:** Carbon dioxide, Methane, Ozone and CFC

Figure 1: Alignment structure for an example question from the science QA dataset. The question and answer candidate are combined to generate a hypothesis sentence. Then alignments (shown by red lines) are found between the hypothesis and the appropriate snippet in the texts.

(as shown in our results).

3.1 Alignment Based Models

Alignment based models for QA (Yih et al., 2013; Sachan et al., 2015; Sachan et al., 2016) cast QA as a textual entailment problem by converting each question-answer candidate pair (q_i, a_{ij}) into a hypothesis statement h_{ij}. For example, the question "What are the important greenhouse gases?" and answer candidate "Carbon dioxide, Methane, Ozone and CFC" in Figure 1 can be combined to achieve a hypothesis "The important greenhouse gases are Carbon dioxide , Methane, Ozone and CFC.". A set of question matching/rewriting rules are used to achieve this transformation. These rules match the question into one of a large set of pre-defined templates and apply a unique transformation to the question and answer candidate to achieve the hypothesis statement. For each question q_i, the QA task thereby reduces to picking the hypothesis $\hat{h}_i$ that has the highest likelihood among the set of hypotheses $\mathbf{h}_i = \{h_{i1}, \ldots, h_{im}\}$ generated for that question of being entailed by a body of relevant texts. The body of relevant texts can vary for each instance of the QA task. For example, it could be just the passage in a reading comprehension task, or a set of science textbooks in the science QA task. Let $h_i^* \in \mathbf{h}_i$ be the correct hypothesis. The model considers the quality of word alignment from a hypothesis h_{ij} (formed by combining question-answer candidates (q_i, a_{ij})) to snippets in the textbooks as a proxy for the evidence. The alignment depends on: (a) snippet from the relevant texts chosen to be aligned to the hypothesis and (b) word alignment from the hypothesis to the snippet. The snippet from the texts to be aligned to the hypothesis is determined by picking a subset of sentences in the texts. Then each hypothesis word is aligned to a unique word in the snippet. See Figure 1 for an illustration. The choice of snippets composed with the word alignment is latent. Let $\mathbf{z}_{ij}$ represent the latent structure for the question-answer candidate pair $(q_i, a_{i,j})$. A natural solution is to treat QA as a problem of ranking the hypothesis set $\mathbf{h}_i$ such that the correct hypothesis is at the top of this ranking. Hence, a scoring function $S_{\mathbf{w}}(h, \mathbf{z})$ is learnt such that the score given to the correct hypothesis h_i^* and the corresponding latent structure $\mathbf{z}_i^*$ is higher than the score given to any other hypothesis and its corresponding latent structure. In fact, in a max-margin fashion, the model learns the scoring function such that $S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) > S_{\mathbf{w}}(h_{ij}, \mathbf{z}_{ij}) + \Delta(h_i^*, h_{ij}) - \xi_i$ for all $h_j \in \mathbf{h} \setminus h^*$ for some slack ξ_i. This can be formulated as the following optimization problem:

$$\min_{||\mathbf{w}||} \quad \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_i \xi_i$$

$$s.t. \quad S_{\mathbf{w}}(h_i^*, \mathbf{z}_i^*) \geq \max_{\mathbf{z}_{ij}} S_{\mathbf{w}}(h_{ij}, \mathbf{z}_{ij}) + \Delta(h_i^*, h_{ij}) - \xi_i$$

It is intuitive to use 0-1 cost, i.e. $\Delta(h_i^*, h_{ij}) = \mathbb{1}(h_i^* \neq h_{ij})$ If the scoring function is convex then this objective is in concave-convex form and can be minimized by the concave-convex programming procedure (CCCP) (Yuille and Rangarajan, 2003). The scoring function is assumed to be linear: $S_{\mathbf{w}}(h, \mathbf{z}) = \mathbf{w}^T \psi(h, \mathbf{z})$. Here, $\psi(h, \mathbf{z})$ is a task-dependent feature map (see Sachan et al. (2015) and Sachan et al. (2016) for details).

3.2 Deep Learning Models

We briefly review two neural network models for QA – Iyyer et al. (2014) and Weston et al. (2014).

QANTA: QANTA (Iyyer et al., 2014) answers quiz bowl questions using a dependency tree structured recursive neural network. It combines predictions across sentences to produce a question answering neural network with trans-sentential averaging. The model is optimized using AdaGrad (Duchi et al., 2011). In quiz bowl, questions typically consist of four to six sentences and are associated with factoid answers. Every sentence in the question is guaranteed to contain clues that uniquely identify its answer, even without the context of previous sentences[1]. Recently, QANTA had beaten the well-known Jeopardy! star Ken Jennings at an exhibition quiz bowl contest.

Memory Networks: Memory networks (Weston et al., 2014) are essentially recurrent neural networks with a long-term memory component. The memory can be read and written to, and can be used for prediction. The memory can be seen as

[1] Refer to Figure 1 in (Iyyer et al., 2014) for an example

acting like a dynamic knowledge base. The model is trained using a margin ranking loss and stochastic gradient descent. It was evaluated on a set of synthetic QA tasks. For each task, a set of statements were generated by a simulation of 4 characters, 3 objects and 5 rooms using an automated grammar with characters moving around, picking up and dropping objects are given, followed by a question whose answer is typically a single word[2].

4 Curriculum Learning

Studies in cognitive science (Skinner, 1958; Peterson, 2004; Krueger and Dayan, 2009) have shown that humans learn much better when the training examples are not randomly presented but organized in increasing order of difficulty. The idea of *shaping*, which consists of training a machine learning algorithm with a curriculum was first introduced by (Elman, 1993) in the context of grammatical structure learning using a recurrent connectionist network. This idea also lent support for the much debated Newport's "less is more" hypothesis (Goldowsky and Newport, 1993; Newport, 1990) that child language acquisition is aided, rather than hindered, by limited cognitive resources. Curriculum learning (Bengio et al., 2009) is a recent idea in machine learning, where a curriculum is designed by ranking samples based on manually curated difficulty measures. These measurements are usually not known in real-world scenarios, and are hard to elicit from humans.

4.1 Self-paced Learning

Self-paced learning (SPL) (Kumar et al., 2010; Jiang et al., 2014a; Jiang et al., 2015) reformulates curriculum learning as an optimization problem by jointly modeling the curriculum and the task at hand. Let $\mathbf{v} \in [0,1]^{|Q|}$ be the weight vector that models the weight of the sample questions in the curriculum. The SPL model includes a weighted loss term on all samples and an additional self-paced regularizer imposed on sample weights $\mathbf{v}$. SPL formulation for the ERM framework described in eq 1 can be rewritten as:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^{|Q|}} \sum_{q_i \in Q} v_i L_{\mathbf{w}}(a_i^*, f_{\mathbf{w}}(q_i); \mathcal{K}) + g(v_i, \lambda) + \Omega(\mathbf{w})$$

The problem usually has closed-form solution with respect to $\mathbf{v}$ (described later; lets call the solution $\mathbf{v}^*(\lambda; L)$ for now). $g(\mathbf{v}, \lambda)$ is usually called the self-paced regularizer with the "age" or "pace" parameter λ. g is convex with respect to $\mathbf{v} \in [0,1]^{|Q|}$. Furthermore, $\mathbf{v}(\lambda; L)$ is monotonically decreasing with respect to L, and $\lim_{L\to 0} \mathbf{v}^*(\lambda; L) = 1$ and $\lim_{L\to\infty} \mathbf{v}^*(\lambda; L) = 0$. This means that the model inclines to select easy samples (with smaller losses) in favor of complex samples (with larger losses). Finally, $v^*(\lambda; L)$ is monotonically increasing with respect to λ, and $\lim_{\lambda\to 0} \mathbf{v}^*(\lambda; L) = 0$ and $\lim_{\lambda\to\infty} \mathbf{v}^*(\lambda; L) \leq 1$. This means that when the model "ages" (i.e. the age parameter λ gets larger), it tends to incorporate more, probably complex samples to train a 'mature' model.

Four popular self-paced regularizers in the literature (Kumar et al., 2010; Jiang et al., 2014a; Zhao et al., 2015) are *hard*, *soft logarithmic*, *soft linear* and *mixture*. These SP-regularizers, summarized with corresponding closed form solutions for $\mathbf{v}$ are shown in Table 1. Hard weighting is usually less appropriate as it cannot discriminate the importance of samples. However, soft weighting assigns real-valued weights and reflects the latent importance of samples in training. The soft linear regularizer linearly weighs samples with respect to their losses and the soft logarithmic penalizes the weight logarithmically. Mixture weighting combines both hard and soft weighting schemes. We can solve the model in the SPL regime by iteratively updating $\mathbf{v}$ (closed form solution for $\mathbf{v}$ is shown in Table 1) and $\mathbf{w}$ (by CCCP, AdaGrad or SGD), and gradually increasing the age parameter λ to let harder and harder problems in.

Since its inception, variations of SPL such as self-paced re-ranking (Jiang et al., 2014a), self-paced learning with diversity (Jiang et al., 2014b), self-paced multiple-instance learning (Zhang et al., 2015) and self-paced curriculum learning (Jiang et al., 2015) have been proposed. The techniques have been shown to be useful in some computer vision tasks (Lee and Grauman, 2011; Kumar et al., 2011; Tang et al., 2012; Supancic and Ramanan, 2013; Jiang et al., 2014a). SPL is different from active learning (Settles, 1995) in the sense that the training examples (and labels) are already provided and the solution only orders the examples to achieve a better solution. On the other hand, active learning tries to interactively query

[2] Refer to Table 1 in (Weston et al., 2015) for examples

Regularizer	$g(\mathbf{v};\lambda)$	$v^*(\lambda;L)$
Hard	$-\lambda\mathbf{v}$	$\begin{cases} 1, \textit{if } L \leq \lambda \\ 0, \textit{o/w} \end{cases}$
Soft Linear	$\lambda(\frac{1}{2}\mathbf{v}^2 - \mathbf{v})$	$\begin{cases} -\frac{L}{\lambda} + 1, \textit{if } L \leq \lambda \\ 0, \textit{otherwise} \end{cases}$
Soft Logarithmic	$\sum\limits_{q_i \in Q} \left((1-\lambda)v_i - \frac{(1-\lambda)^{v_i}}{log(1-\lambda)}\right)$	$\begin{cases} \frac{log(L+1-\lambda)}{log(1-\lambda)}, \textit{if } L \leq \lambda \\ 0, \textit{o/w} \end{cases}$
Mixed	$\frac{\gamma^2}{\mathbf{v}+\frac{\gamma}{\lambda}}$	$\begin{cases} 1, \textit{if } L \leq \left(\frac{\lambda\gamma}{\lambda+\gamma}\right)^2 \\ 0, \textit{if } L \geq \lambda^2 \\ \gamma\left(\frac{1}{\sqrt{L}} - \frac{1}{\lambda}\right), \textit{o/w} \end{cases}$

Table 1: Various SP-regularizers for SPL.

the user (or another information source) to achieve a better model with few queries. Curriculum learning is also related to teaching dimension (Khan et al., 2011) which studies the strategies that humans follow as they teach a target concept to a robot by assuming a teaching goal of minimizing the learner's expected generalization error at each iteration. One can also think of curriculum learning as an approach for achieving a better local optimum in non-convex problems.

5 Improved Curriculum Learning Heuristics

SPL selects questions based on the local loss term of the question. This is not the only way to define 'easiness' of the question. Hence, we suggest some other heuristics for selecting the order of questions to be presented to our learning algorithm. The heuristics select the next question $q_i \in Q \setminus Q_0$ given the current model ($\mathcal{M}$) and the set of questions already presented for learning (Q_0). We assume access to a minimization oracle (CCCP/AdaGrad/SGD) for the QA models. We explore the following heuristics:

1) Greedy Optimal (GO): The simplest and greedy optimal heuristic (Schohn and Cohn, 2000) would be to pick a question $q_i \in Q \setminus Q_0$ which has the minimum expected effect on the model. The expected effect on adding q_i can be written as:

$$\sum_{a_{ij} \in A_i} p(a_i^* = a_{ij}) \sum_{q_j \in Q_0 \cup q_i} \mathbb{E}\left[L_{\mathbf{w}}(a_j^*, f_{\mathbf{w}}(q_j); \mathcal{K})\right].$$

$p(a_i^* = a_{ij})$ can be estimated by normalizing $S_{\mathbf{w}}(q,a;\mathcal{K})$. $\sum\limits_{q_j \in Q_0 \cup q_i} \mathbb{E}\left[L_{\mathbf{w}}(a_j^*, f_{\mathbf{w}}(q_j); \mathcal{K})\right]$ can be estimated by retraining the model on $Q_0 \cup q_i$.

2) Change in Objective (CiO): Choose the question $q_i \in Q \setminus Q_0$ that causes the smallest increase in the objective. If there are multiple questions with the smallest increase in objective, pick one of them randomly.

3) Mini-max (M^2): Chooses question $q_i \in Q \setminus Q_0$ that minimizes the regularized expected risk when including the question with the answer candidate a_{ij} that yields the maximum error.

$$\hat{q}_i = \arg\min_{q_i \in Q \setminus Q_0} \max_{a_{ij} \in A_i} L_{\mathbf{w}}(a_{ij}, f_{\mathbf{w}}(q_i); \mathcal{K})$$

4) Expected Change in Objective (ECiO): In this greedy heuristic, we pick a question $q_i \in Q \setminus Q_0$ which has the minimum expected effect on the model. The expected effect can be written as $\sum\limits_{a_{ij} \in A_i} p(a_i^* = a_{ij}) \times \mathbb{E}\left[L_{\mathbf{w}}(a_i^*, f_{\mathbf{w}}(q_i); \mathcal{K})\right]$. Here, $p(a_i^* = a_{ij})$ can be achieved by normalizing $S_{\mathbf{w}}(q,a;\mathcal{K})$ and $\mathbb{E}\left[L_{\mathbf{w}}(a_i^*, f_{\mathbf{w}}(q_i); \mathcal{K})\right]$ can be estimated by running inference for q_i.

4) Change in Objective-Expected Change in Objective (CiO - ECiO): We pick a question $q_i \in Q \setminus Q_0$ which has the minimum value of the difference between the change in objective and the expected change in objective. Intuitively, the difference represents how much the model is surprised to see this new question.

5) Correctly Answered (CA): Pick a question $q_i \in Q \setminus Q_0$ which is answered by the model $\mathcal{M}$ with the minimum cost $\Delta(\hat{a}_i, a_i^*)$. If there are multiple questions with minimum cost, pick one of them randomly.

6) Farthest from Decision Boundary (FfDB): This heuristic applies for latent structural SVMs only. Here, we choose the question $q_i \in Q \setminus Q_0$ whose predicted answer $\hat{a}_i$ is farthest from the decision boundary: $\max\limits_{\mathbf{z}^*} w^T \phi(q_i, a^*, \mathbf{z}^*, \mathcal{K}) = \max\limits_{\hat{\mathbf{z}}} w^T \phi(q, \hat{a}, \hat{\mathbf{z}}, \mathcal{K}) + \Delta(\hat{a}, a^*)$.

5.1 Timing Considerations:

A key consideration in applying the above heuristics is efficiency as the QA models considered (latent structural SVM and deep learning) are computationally expensive. Among our selection strategies, *GO* and *CiO* require updating the model, *M*^{2}, *ECiO*, *CA* and *FfDB* require performing inference on the candiate questions, while *CiO* - *ECiO* requires both retraining as well as inference. Consequently, *M*^{2}, *ECiO*, *CA* and *FfDB* are most efficient. We can also gain considerable speed-up by picking questions in batches. This results in significant speed-up with small loss in accuracy. We will discuss the batch question selection setup in more detail in our experiments.

5.2 Smarter Selection Strategies:

We further describe some improvements to the above selection strategies:

1) Ensemble Strategy: In this strategy, we combine all of the above heuristics into an ensemble. The ensemble computes the ratio of the score of the suggested question pick and the average score over remaining $Q \setminus Q_0$ questions for all the heuristics and picks the question with the highest ratio. As we will see in our results, this ensemble works well in practice.

2) Importance-Weighting (IW): Importance weighting is a common technique in active learning literature (Tong and Koller, 2002; Beygelzimer et al., 2009; Beygelzimer et al., 2010), which mitigates the problem that if we query questions actively instead of selecting them uniformly at random, the training (and test) question sets are no longer independent and identically distributed (i.i.d.). In other words, the training set will have a sample selection bias that can impair prediction performance. To mitigate this, we propose to sample questions from a biased sample distribution $\widetilde{D}$. To achieve $\widetilde{D}$, we introduce the weighted loss $\widetilde{L}_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K}) = \widetilde{w}(q,a) \times L_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K})$ where $\widetilde{w}(q,a)$ is the weighting function $\widetilde{w}(q,a) = \frac{p_D(q,a)}{p_{\widetilde{D}}(q,a)}$ which represents how likely it is to observe (q,a) under D compared to $\widetilde{D}$. In this setting, we can show that the generalization error under $\widetilde{D}$ is the same as that under D:

$$
\begin{aligned}
&\mathbb{E}_{(q,a)\sim\widetilde{D}}\left[\widetilde{L}_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K})\right] \\
&= \int_{(q,a)} p_{\widetilde{D}}(q,a) \frac{p_D(q,a)}{p_{\widetilde{D}}(q,a)} L_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K}) d(q,a) \\
&= \int_{(q,a)} p_D(q,a) L_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K}) d(q,a) \\
&= \mathbb{E}_{(q,a)\sim D}\left[L_{\mathbf{w}}(a, f_{\mathbf{w}}(q); \mathcal{K})\right]
\end{aligned}
$$

Thus, given appropriate weights $\widetilde{w}(q,a)$, we modify our loss-function in order to compute an unbiased estimator of the generalization error. Each question-answer is assigned with a non-negative weight. For latent structural SVMs, one can minimize the weighted loss by simply multiplying the corresponding regularization parameter C_i with a corresponding term. In neural networks, this is simply achieved by multiplying the gradients with the corresponding weights. The weights can be set by an appropriate heuristic, e.g. proportional to distance from the decision boundary.

5.3 Incorporating Diversity with Explore and Exploit (E&E):

The strategy of learning from easy questions first and then gradually handling harder questions is intuitive as it helps the learning process. Yet, it has one key deficiency. Under curriculum learning, by focusing on easy questions first, our learning algorithm is usually not exposed to a diverse set of questions. This is particularly a problem for deep-learning approaches that learn representations during the process of learning. Hence, when a harder question arrives, it is usually hard for the learner to adjust to this new question as the current representation may not be appropriate for the new level of difficulty. This motivates our E&E strategy.

The explore and exploit strategy ensures that while we still select easy questions first, we also want to make our selection as diverse as possible. We define a measure for diversity as the angle between the hyperplanes that the question samples induce in feature space: $\angle(\phi(q_i, a_i^*, \mathbf{z}_i^*, \mathcal{K}), \phi(q_{i'}, a_{i'}^*, \mathbf{z}_{i'}^*, \mathcal{K})) = \mathit{Cosine}^{-1}\left(\frac{|\phi(q_i, a_i^*, \mathbf{z}_i^*, \mathcal{K})\phi(q_{i'}, a_{i'}^*, \mathbf{z}_{i'}^*, \mathcal{K})|}{||\phi(q_i, a_i^*, \mathbf{z}_i^*, \mathcal{K})||||\phi(q_{i'}, a_{i'}^*, \mathbf{z}_{i'}^*, \mathcal{K})||}\right)$. The *E&E* solution picks the question which optimizes a convex combination of the curriculum learning objective and the sum of angles between the candidate question pick and questions in Q_0. The convex combination is tuned on the development set.

6 Experiments

6.1 Datasets

As described, we study curriculum learning on four different tasks. The first task is question answering for reading comprehensions. We use MCTest-500 dataset (Richardson et al., 2013), a freely available set of 500 stories (300 train, 50 dev and 150 test) and associated questions to evaluate our model. Each story in MCTest has four

		Machine Comprehension	Science QA	QANTA	Memory Networks
	No Curriculum (NC)	66.62±0.22	42.77±0.04	70.40±0.07	75.03±0.06
SPL	Hard	67.36±0.16	43.85±0.18	70.66±0.19	71.01±0.09
	Soft Linear	68.04±0.17	43.80±0.22	71.65±0.18	72.33±0.07
	Soft Log	68.89±0.16	44.19±0.20	71.92±0.16	73.32±0.09
	Mixed	69.47±0.18	44.86±0.20	72.89±0.19	74.28±0.13
Heuristics	CA	66.86±0.06	42.93±0.08	70.78±0.08	70.96±0.04
	M^2	66.98±0.12	43.19±0.17	71.02±0.18	69.73±0.06
	ECiO	67.39±0.14	44.00±0.22	71.66±0.19	71.01±0.07
	GO	67.65±0.12	44.35±0.15	71.94±0.17	71.28±0.06
	CiO	68.20±0.10	44.56±0.12	72.61±0.14	71.98±0.06
	FfDB	68.32±0.11	44.78±0.13	-	-
	CiO-ECiO	68.65±0.13	44.97±0.11	73.34±0.10	73.22±0.05
Heur++	Ensemble	69.26±0.08	45.48±0.07	74.11±0.07	74.24±0.04
	+IW	69.86±0.10	45.86±0.12	75.02±0.15	74.55±0.05
	+E&E	69.93±0.13	46.57±0.17	76.24±0.15	77.64±0.11
	+IW+E&E	**70.16**±0.14	**46.68**±0.19	**76.89**±0.18	**77.85**±0.09
SPL+E&E	Hard	68.03±0.17	44.50±0.20	72.34±0.19	74.43±0.06
	Soft Linear	68.51±0.19	44.43±0.21	73.16±0.19	75.74±0.07
	Soft Log	69.27±0.18	44.92±0.20	73.47±0.18	76.63±0.10
	Mixed	69.89±0.21	45.58±0.21	74.39±0.21	77.12±0.15

Table 2: Accuracy on the test set obtained on the four experiments, comparing results when no curriculum (NC) was learnt, when we use self-paced learning (SPL) with four variations of SP-regularizers, the six heuristics and four improvements proposed by us. Each cell reports the mean±se (standard error) accuracy over 10 repetitions of each experimental configuration.

multiple-choice questions, each with four answer choices. Each question has exactly one correct answer. The second task is science question answering. We use a mix of 855 third, fourth and fifth grade science questions derived from a variety of regional and state science exams[3] for training and evaluating our model. We used publicly available science textbooks available through $ck12.org$ and Simple English Wikipedia[4] as texts required to answer the questions. The model retrieves a section from the textbook or a Wikipedia page (using a lucene index on the sections and Wikipedia pages) by querying for the hypothesis h_{ij} and then aligning the hypothesis to snippets in the document. For QANTA (Iyyer et al., 2014), we use questions from quiz bowl tournaments for training as in Iyyer et al. (2014). The dataset contains 20,407 questions with 2347 answers. For each answer in the dataset, its corresponding Wikipedia page is also provided. Finally, for memory networks (Weston et al., 2014), we use the synthetic QA tasks defined in Weston et al. (2015) (version 1.1 of the dataset). There are 20 different types of tasks that probe different forms of reasoning and deduction. Each task consists of a set of statements, followed by a question whose answer is typically a single word or a set of words. We report mean accuracy across these 20 tasks.

6.2 Results

We implemented and compared the six selection heuristics (§5) with the suggested improvements (§5.2) and self-paced learning (§4) with the explore and exploit extension for both alignment based models (§3.1) and two deep learning models (§3.2). We use accuracy (proportion of test questions correctly answered) as our evaluation metric. In all our experiments, we begin with zero training data (random initialization). For alignment based models, we select 1 percent of training set questions after every epoch (an epoch is defined as a single pass through the current training set by the optimization oracle) and add them to the training set based on the selection strategy. For deep learning models, we discovered that the learning was a lot slower so we added 0.1 percent of new training set questions after every epoch. Hyper parameters of the alignment based models and the deep learning models were fixed to the corresponding values proposed in their corresponding papers (pre-tuned for the optimization oracle on a held-out development set). All the results reported in this paper are averaged over 10 runs of each experiment.

Table 5.3 reports test accuracies obtained on all the QA tasks, comparing the aforementioned proposals against corresponding models when curriculum learning is not used. We can observe from

[3] http://aristo-public-data.s3.amazonaws.com/AI2-Elementary-NDMC-Feb2016.zip

[4] https://dumps.wikimedia.org/simplewiki/20151102/

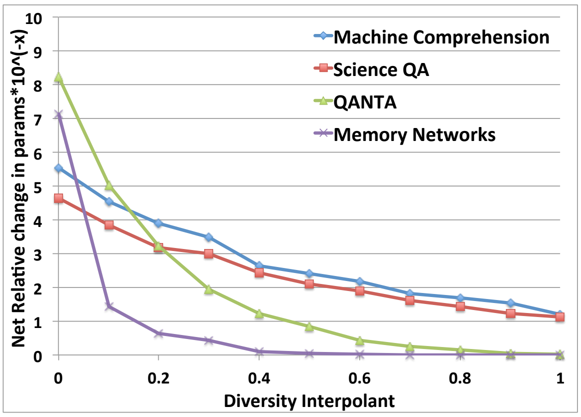


Figure 2: Relative change in parameters*10^{-x} where $x = 2$ for machine comprehension and science QA, 4 for QANTA and memory networks when CL is used.

these results that variants of SPL (and *E&E*) as well as the heuristics (and improvements) lead to improvements in the final test accuracy for both alignment-based models and QANTA.

The surprising ineffectiveness of the heuristics and SPL for memory networks essentially boils down to the abrupt restructure of memory the model has to do for curriculum learning. We provide support for this argument in Figure 2 which plots the net relative change in all the parameters $\mathbf{W}$ until convergence $\left(\frac{1}{\textit{No. of parameters}} \sum_{e:epoch=1}^{\infty} \frac{||\mathbf{W}_{e+1} - \mathbf{W}_{e}||_1}{||\mathbf{W}_{e}||_1} \right)$ for each of the four tasks on the model *Ensemble+E&E* against the linear interpolant used to tune the explore and exploit combination. As the interpolant grows from 0 to 1, more and more diverse questions get selected. We can observe that the change in parameters decreases as more diverse questions are selected for all the four tasks. Furthermore, once we bring in diversity (change the interpolant from 0 to 0.1), the relative change in parameters drops sharply for both neural network approaches. The drop is sharpest for memory networks. Easier examples usually require less memory than hard examples. Memory networks have no incentive to utilize only a fraction of its state for easy examples. They simply use the entire memory capacity. This implies that harder examples appearing later require a restructuring of all memory patterns. The network needs to change its memory representation every time in order to free space and accommodate the harder example. This process of memory pattern restructuring is difficult to achieve, so it could be the reason for the relatively poor performance of naive curriculum learning and SPL strategies. However, as we can see from the previous results, the explore and exploit strategy of mixing in some harder examples avoids the problem of having to abruptly restructure memory patterns. The extra samples of all difficulties prevent the network from utilizing all the memory on the easy examples, thus eliminating the need to restructure memory patterns.

From Table 5.3 , we can observe that the choice of the SP-regularizer is important. The soft regularizers perform better than the hard regularizer. The mixed regularizer (with mixture weighting) performs even better. We can also observe that all the heuristics work as well as SPL, despite being a lot simpler. The heuristics arranged in increasing order of performance are: *CA*, *M*^{2}, *ECiO*, *GO*, *CiO*, *FfDB* and *CiO-ECiO*,. The differences between the heuristics are larger for alignment-based models and smaller for deep learning models. The *ECiO* heuristic has very similar performance to SPL with hard SP-regularizer. This is understandable as SPL also selects 'easy' questions based on their expected objective value. The *Ensemble* is a significant improvement over the individual heuristics. Importance weighting (*IW*) and the explore and exploit strategies (*E&E*) provide further improvements. *E&E* is crucial to making curriculum learning work for deep learning approaches as described before. Motivated by the success of *E&E*, we also extended it to SPL[5] by tuning a convex combination as before. *E&E* provides improvements across all the experiments for all the SPL experiments. While, the strategy is more important for memory networks, it leads to improvements on all the tasks.

In order to understand the curriculum learning process and to test the hypothesis that the procedure indeed selects easy questions first, successively moving on to harder questions, we plot the number of questions of grade 3, 4 and 5 picked by *SPL*, *Ensemble* and *Ensemble+E&E* against the epoch number in Figure 3. We can observe that all the three methods pick more questions from grade 3 initially, successively moving on to more and more grade 4 questions and finally more grade 5 questions. Both *Ensemble* and *Ensemble+E&E* are more aggressive at learning this curriculum than *SPL*. *Ensemble* becomes too aggressive so

[5] This is different from Jiang et al. (2014c) which encourages diversity in samples across groups. On the other hand, we encourage diversity in feature space.

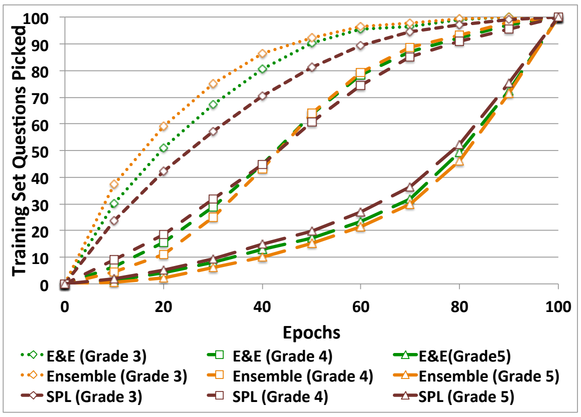


Figure 3: Number of grade 3, 4 and 5 questions picked vs Epoch for various CL approaches for Science QA.

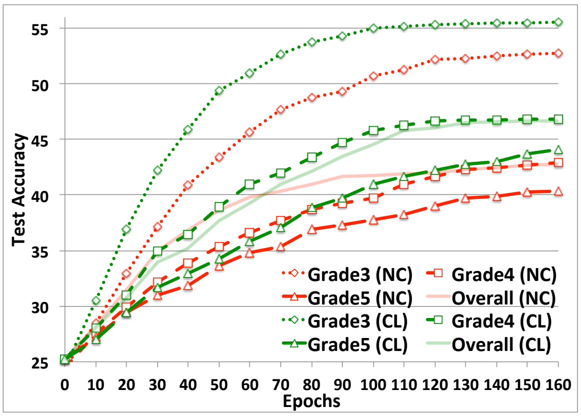


Figure 4: Test split accuracy on grade 3, 4 and 5 questions picked vs Epoch for Science QA when CL is used/not used.

E&E, initially increases the number of grade 4 and grade 5 questions received by the learner, thereby incorporating diversity in learning.

In order to further the claim that curriculum learning follows the principal of learning simpler concepts first and then learning successively harder and harder concepts, we plot the test accuracy on grade 3, 4 and 5 questions with curriculum learning (CL) – i.e. *Ensemble+E&E* and without curriculum learning (NC) against the epoch number in Figure 4. Here, we can see that the test accuracy increases for questions in all three grade levels. With curriculum learning, the accuracy on grade 3 questions rises sharply in the beginning. This rise is sharper than the case when curriculum learning is not used. Grade 3 test accuracy for curriculum learning then saturates (saturates earlier compared to the case when curriculum learning is not used). The improvements due to curriculum learning for grade 4 questions mainly occur in epochs 30-140. The final epochs of curriculum learning see greater gain in test accuracy for grade 5 questions over the case when curriculum learning is not used. All these experiments together support the intuition of curriculum learning. The models indeed pick and learn from easier questions first and successively learn from harder and harder questions. We also tried variants of our models where we used curriculum learning on grade 3 questions, followed by grade 4 and grade 5 questions. However, this did not lead to significant improvements. Perhaps, this is because questions that are easy for humans may not always correspond to what is easy for our algorithms. Characterizing what is easy for algorithms and how it relates to what is easy for humans is an interesting question for future research.

7 Conclusion

Curriculum learning is inspired by the way humans acquire knowledge and skills: by mastering simple concepts first, and progressing through information with increasing difficulty to grasp more complex topics. We studied self-paced learning, an approach for curriculum learning that expresses the difficulty of a data sample in terms of the value of the objective function and builds the curriculum via a joint optimization framework. We proposed a number of heuristics, an ensemble, and several improvements for selecting the curriculum that improves upon self-paced learning. We stressed on another important aspect of human learning – diversity, that requires that the right curriculum should not only arrange the data samples in increasing order of difficulty but should also introduce the learner to a small number of samples that are sufficiently dissimilar to the samples that have already been introduced to the learning process. We showed that our heuristics when coupled with diversity lead to significant improvements in a number of question answering tasks. The approach is quite general and we hope that this paper will encourage more NLP researchers to explore curriculum learning in their own works.

Acknowledgments

We thank the anonymous reviewers, along with Emmanouil A. Platanios and Snigdha Chaturvedi for their valuable comments and suggestions that helped improve the quality of this paper. This work was supported by the following research grants: NSF IIS1218282, NSF IIS1447676 and AFOSR FA95501010247.

References

[Arai and Matsuzaki2014] Noriko H Arai and Takuya Matsuzaki. 2014. The impact of ai on education– can a robot get into the university of tokyo? In *Proc. ICCE*, pages 1034–1042.

[Bengio et al.2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

[Beygelzimer et al.2009] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 49–56. ACM.

[Beygelzimer et al.2010] Alina Beygelzimer, John Langford, Zhang Tong, and Daniel J Hsu. 2010. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems*, pages 199–207.

[Cantor1946] Nathaniel Freeman Cantor. 1946. *Dynamics of learning*. Foster and Stewart publishing corporation, Buffalo, NY.

[Clark and Etzioni2016] Peter Clark and Oren Etzioni. 2016. My computer is an honor student - but how intelligent is it? standardized tests as a measure of ai. In *Proceedings of AI Magazine*.

[Duchi et al.2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

[Elman1993] Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

[Fujita et al.2014] Akira Fujita, Akihiro Kameda, Ai Kawazoe, and Yusuke Miyao. 2014. Overview of todai robot project and evaluation framework of its nlp-based problem solving. *World History*, 36:36.

[Goldowsky and Newport1993] B.N. Goldowsky and E.L. Newport. 1993. Modeling the effects of processing limitations on the acquisition of morphology: The less is more hypothesis. In *Proceedings of the 11th West Coast Conference on Formal Linguistics*.

[Iyyer et al.2014] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of Empirical Methods in Natural Language Processing*.

[Jiang et al.2014a] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. 2014a. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the ACM International Conference on Multimedia*, pages 547–556. ACM.

[Jiang et al.2014b] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. 2014b. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086.

[Jiang et al.2014c] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. 2014c. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086.

[Jiang et al.2015] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Jurafsky and Martin2000] Daniel Jurafsky and James H Martin. 2000. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall.

[Khan et al.2011] Faisal Khan, Bilge Mutlu, and Xiaojin Zhu. 2011. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems*, pages 1449–1457.

[Krueger and Dayan2009] Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

[Kumar et al.2010] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197.

[Kumar et al.2011] M Pawan Kumar, Haithem Turki, Dan Preston, and Daphne Koller. 2011. Learning specific-class segmentation from diverse data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1800–1807. IEEE.

[Kushman et al.2014] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[LeCun et al.2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

[Lee and Grauman2011] Yong Jae Lee and Kristen Grauman. 2011. Learning the easy things first: Self-paced visual category discovery. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1721–1728. IEEE.

[Newport1990] Elissa L Newport. 1990. Maturational constraints on language learning. *Cognitive science*, 14(1):11–28.

[Peterson2004] Gail B Peterson. 2004. A day of great illumination: Bf skinner's discovery of shaping. *Journal of the Experimental Analysis of Behavior*, 82(3):317–328.

[Richardson et al.2013] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

[Sachan et al.2015] Mrinmaya Sachan, Avinava Dubey, Eric P Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[Sachan et al.2016] Mrinmaya Sachan, Avinava Dubey, and Eric P. Xing. 2016. Science question answering using instructional materials. *CoRR*, abs/1602.04375.

[Schohn and Cohn2000] Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *Proceedings of the 17th Annual International Conference on Machine Learning*. ACM.

[Seo et al.2014] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram understanding in geometry questions. In *Proceedings of AAAI*.

[Settles1995] Burr Settles. 1995. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

[Skinner1958] Burrhus F Skinner. 1958. Reinforcement today. *American Psychologist*, 13(3):94.

[Supancic and Ramanan2013] James Supancic and Deva Ramanan. 2013. Self-paced learning for long-term tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2379–2386.

[Tang et al.2012] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. 2012. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646.

[Tong and Koller2002] Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66.

[Weston et al.2014] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

[Weston et al.2015] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

[Yih et al.2013] Wentau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

[Yu and Joachims2009] Chun-Nam Yu and T. Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of International Conference on Machine Learning (ICML)*.

[Yuille and Rangarajan2003] A. L. Yuille and Anand Rangarajan. 2003. The concave-convex procedure. *Neural Comput.*

[Zhang et al.2015] Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. 2015. A self-paced multiple-instance learning framework for co-saliency detection. June.

[Zhao et al.2015] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. 2015. Self-paced learning for matrix factorization. In *Proceedings of AAAI*.