Dynamic Curriculum Learning with Item Response Theory

Anonymous AAAI 2020 Submission

Abstract

CL has been shown to be effective. Many measures of "difficulty" are based on heuristics and not on true difficulty. we want to change that by leveraging methods from psychometrics. Experiments on vision and language data show that using learned difficulty and ability to set up a curriculum is effective. Curriculum that is dynamically determined by the current competency of the model as estimated by Item Response Theory.

Introduction

- go over CL - why it works - why there are currently issues with it (heuristics) - bottlenecks to some of the existing approaches

Curriculum learning is a well-studied area of artificial intelligence, with results demonstrating that building a curriculum of training data where easy examples are shown first can speed up learning and improve generalization, as has also been shown in humans. The basic premise is that machine learning (ML) models are trained according to a "curriculum" that sorts training examples according to difficulty. At first, the model is trained with only the easiest examples, and more difficult examples are gradually added according to some schedule. A major gain for CL methods is that model convergence can often be quicker when the model is trained according to some curriculum (CITE a bunch of examples here). Model efficiency has always been a major aim of the research community (cite old papers), but recently with the size of models continuing to grow and better understanding of the impact of model training on the environment there is a renewed need for efficiency (CITE EMMA).

A major drawback of existing CL techniques is that they rely on heuristics to measure the difficulty of data, and either ignore the competency of the model at its present state or rely on heuristics there as well. For example, often for natural language processing (NLP) tasks, sentence length is considered a proxy for difficulty (cite naacl19 and original paper). Similarly, the original CL paper used the number of objects in an image as a proxy for difficulty in an image recognition task (CCITE Bengio). Competency was recently introduced as a mechanism to dictate when new examples should be added to the training data (CITE NAACL 19), however in that work the competency schedule was ad hoc, and did not actually look at the competency of the model but assumed a schedule according to learning heuristics (CITE NAACL). It would be better if model competency was actually measured, so that the training data could be appropriately matched with the model at a given point in the training.

Recent work has shown that it is possible to estimate both the difficulty of examples and the ability of deep learning models as latent variables based on model performance (cite emnlp19). Item Response Theory (IRT) is a well-studied methodology in the psychometric literature for test set construction and subject evaluation. A typical IRT model will estimate latent parameters such as difficulty for examples under consideration for inclusion in a test set. This is done by administering a test to a large number of human subjects, collecting and grading their responses as correct or incorrect, and using the student-response data matrix to estimate the latent traits of the data. Once learned, these latent parameters can be used to estimate latent ability parameters of future test-takers, based on their graded responses to the examples. IRT has not seen wide adoption in the ML community, primarily due to the fact that fitting IRT models typically requires a large amount of human annotated data for each example. However, recent work has shown that it is possible to fit IRT models using machine-generated data instead of human-generated data as the input to the IRT models (CITE US). Because one can learn example difficulty and subject ability together, IRT is an interesting framework to consider for the problem of CL.

In this work we propose Dynamic Curriculum Learning with IRT (DCL-IRT), a novel framework that uses the estimated ability of a model at a specific point in the training process to identify appropriate training data. At each training epoch, the latent ability of the model is estimated. Based on this estimate, only training data that the model has a reasonable chance of labeling correctly is included in training. As the model improves, the estimated ability will improve, and more training data will be added.

Our contributions are as follows: (i) we propose a novel CL framework, DCL-IRT, which automatically selects training data based on the estimated ability of the model, (b) we show that model training using DCL-IRT leads to faster convergence than traditional training and better performance than baseline CL methods, (c) we provide an analysis of the DCL-IRT training regime to show why certain training examples hurt instead of help generalization. This is the first work to learn a model competency during training that is directly comparable to the difficulty of the training data pool.

All code used for this work will be released upon publication, and are included as supplemental material.

Methods

- brief review of IRT (focus on rasch model because that is what we're using)

- talk about how to fit the model using variational inference

- selecting data as the model is ready for it (based on theta)

Curriculum Learning

In a traditional CL framework, training data examples are ordered according to some notion of difficulty, and the training set shown to the learner is augmented at a set pace with more and more difficult examples.

Typically, the model's current performance is not taken into account (CONFIRM THAT THIS IS TRUE FOR EVERYTHING BUT NAACL PAPER). Recent work has incorporated a notion of "competency," but in actual fact just used a slightly more complex inclusion rate calculation that does not actually consider the competency of the model but simulates it.

Learning Latent Parameters with IRT

Learning the latent difficulty parameters of training examples can be done off-line using existing techniques (CITE EMNLP PAPER). Traditional IRT model fitting relies on marginal maximum likelihood (MML) estimation, where the latent ability parameters (θ) are assumed to be random effects and integrated out. Latent item parameters are estimated using an expectation maximization (EM) method (CITE BOCK and AIKEN). These methods assume human subjects, and were typically used for relatively small data sets of no more than a hundred items or so. A major bottleneck of using IRT methods on machine learning data sets is the fact that each subject would have to label each (or most of) the examples in order to have enough response patterns to estimate the latent parameters. Asking humans to annotate tens or hundreds of thousands of examples can't be done, but recent work has shown that the human subjects can be replaced with an ensemble of machine learning models, and the response patterns from this "artificial crowd" can be used to estimate item parameters. Such models can be fit using variational inference methods, where the latent parameters are estimated via variational distributions (CITE Minka and Us).

Estimating the ability of a model at a point in time is done with a "scoring" function. Since the difficulties of the examples are known, we can maximize the likelihood of the data given the response patterns to obtain the ability estimate. All that is required is a single forward pass of the model on the data, as is typically done with a test set.

Dynamic Curriculum Learning with IRT

We propose DCL-IRT, where the training examples are selected dynamically at each training epoch based on the estimated ability of the model at that point. With DCL-IRT, model ability can be estimated according to a well-studied psychometric framework as opposed to heuristics. At the same time, the estimated ability of the model ($\hat{\theta}$) is on the same scale as the difficulty parameters of the data, so there is a principled recipe for selecting data at any given training epoch.

The first step of DCL-IRT is to estimate the ability of the model using the scoring function (Sec. above). To do this we use the full training set, but crucially, only to get response data, not to update parameters. Model outputs are obtained for the training set, and graded as correct or incorrect as compared to the gold standard label. This response pattern is then used to estimate model ability at the current epoch ($\hat{\theta}_e$).

Once ability is estimated, data selection is done by comparing estimated ability to the examples' difficulty parameters. Each example in the training pool has an estimated difficulty parameter (b_x). If the difficulty of an example is less than or equal to the estimated ability, then the example is included in training for this epoch. Examples where the difficulty is greater than estimated ability are not included.

With DCL-IRT, the training data size does not have to be monotonically increasing. If a model's performance suffers as a result of adding data too quickly, then this will be reflected in lower ability estimates, which leads to less data selected in the next epoch. This avoids a scenario where data is added too quickly at the expense of learning the easier examples. At the same time, if estimated model ability is high, then more data can be added more quickly, without artificially slowing down the learning due to a rigid schedule.

Algorithm XX shows all of the steps for DCL-IRT. Code implementing DCL-IRT is included as supplemental material and will be released upon publication.

Data and experiments

We experiment with four data sets (two from vision and two from NLP) to demonstrate the effectiveness of DCL-IRT across multiple domains: handwritten digit recognition, image recognition, sentiment analysis, and natural language inference.

MNIST

There are 60,000 training examples and 10,000 test examples in the data set. We retain the training/test split that is typically used for evaluation.

CIFAR

There are 50,000 training example and 10,000 test examples in the data set.

SSTB

There are XX,XXX examples in the data set.

SNLI

There are 500,000 ish examples in the data set.

Generating Response Patterns

In order to learn the difficulty parameters of the data we require a data set of response patterns. As previously mentioned, gathering enough labels for each example in the data sets to fit an IRT model would be prohibitively expensive if using humans. In addition, the annotation quality would be suspect due to the humans labeling tens of thousands of examples. Therefore we used artificial crowds to generate our response patterns (CITE EMNLP).

Briefly, for each data set an ensemble of neural network models are trained, using different subsets of the training data set. Training data is subsampled and corrupted (label flipping) so that performance across models in the ensemble is varied. Each trained model then labels the test set examples. These labels are graded correct/incorrect against the gold-standard test label and the output response patterns are used to fit an IRT model for the data. Prior work has shown that this is an effective way to generate a set of response patterns for fitting IRT models to machine learning data (cite us).

Experiments

In order to demonstrate the effectiveness of DCL-IRT we must show that the model is more efficient than standard supervised learning training while maintaining the level of performance in terms of test set accuracy. Any gains in predictive performance are an additional benefit, but are not the main goal. With this in mind we performed the following experiment.

For each data set, we trained a standard model architecture for a set number of epochs. We varied the training data available to the model at each epoch based on the type of curriculum applied:

- Baseline: At each epoch, the model has access to all of the data, shuffled and in mini-batches
- Ordered: At each epoch, the model has access to all of the data, but here the data is ordered according to difficulty (i.e. not shuffled)
- Simple-IRT: At each epoch, the model has access to $\frac{e}{N}$ training examples, where e is the current epoch and N is the full training set size. This strategy ignores the competence of the model and only uses the example difficulty as inclusion criteria
- DCL-IRT: At each epoch, model ability is estimated ($\hat{\theta}_e$, see §above) and all training examples where difficulty is less that $\hat{\theta}_e$ are included.

For Ordered and Simple, there are several options available in terms of which examples are introduced first. For each method we experiment with ordering the data in three ways:

- Easy-first: The data are ordered from easiest to hardest. At each epoch, more difficult examples are included in training
- Hard-first: The data are ordered from hardest to easiest. At each epoch, easier examples are included in training
- Middle-out: The data are ordered from smallest to largest in terms of the absolute value of the difficulty parameter. Recall that difficulty is roughly normally distributed (CLEAN THIS UP BECAUSE THAT ISN'T EXACTLY TRUE). At the first training epoch, data that are of an "average" difficulty are included in training. At each subsequent epoch, some data that are slightly easier and some data that are slightly harder are added to the training set.

For each data set a relatively simple model architecture was used. Performance in terms of test set accuracy is determined by using the development set accuracy as an early stopping indicator. Each model was trained for 200 epochs.

- experiments: ordered vs simple vs irt, balanced vs not balanced

Results

- plots and tables

- really show that using our method is efficient and practical

Using IRT leads to quicker convergence for the trained models (Figure XX). The vertical lines in each plot indicate the point at which the model has converged, based on early stopping using the development set accuracy. In most cases DCL-IRT leads to the fastest convergence, however for XX using the Simple-IRT curriculum ordered easiest to hardest leads to the fastest convergence. In all cases some use of IRT to determine the curriculum improves over the baseline training criterion.

By using DCL-IRT a curriculum can adapt during training according to the estimated ability of the model. DCL-IRT adds or removes training data based not on a fixed step schedule but rather by probing the model at each epoch and using the estimated ability to match data to the model (Figure XX). This way if a model has a high estimated ability early in training, then more data can be added to the training set more quickly, and learning isn't artificially slowed down due to the curriculum schedule. For each data set in question, DCL-IRT adds training data more quickly than a more traditional CL schedule, which leads to faster convergence.

Performance for ordered is strange. Sometimes it works very well (SSTB), sometimes it doesn't work at all (SNLI). More work is required here to see what is going on (PUT IT IN THE ANALYSIS SECTION).

Even though training with these curriculum use less data than the baseline, efficiency does not have a significant negative impact on generalizability in terms of test set performance (Table). The number of training examples required to reach convergence is lower than the baseline in each case for DCL-IRT. Change in performance compared to the baseline is often lower, but the delta is extremely small (usually XX or YY relative change). In particular, the amount of data needed to achieve very high performance on the SSTB task

Data Set	Experiment	Training Examples	Δ_b (%)	Accuracy (%)	Δ_b (%)
MNIST	Baseline	10,320,000	100.0	99.26	0
	EasyFirst	7,620,000	73.84	99.17	-0.09
	MiddleOut	6,420,000	**62.21**	99.2	-0.06
	Ordered	10,020,000	97.09	97.06	-2.22
	Theta	8,986,612	87.08	99.23	**-0.03**
CIFAR	Baseline	6,750,000	100.0	86.61	0
	EasyFirst	6,900,000	102.22	85.54	**-1.24**
	MiddleOut	3,250,000	**48.15**	85.54	**-1.24**
	Ordered	7,250,000	107.41	39.04	-54.92
	Theta	3,671,016	54.39	83.32	-3.80
SSTB	Baseline	8,687,892	100.0	85.71	0
	EasyFirst	2,912,757	**33.53**	85.38	-0.39
	MiddleOut	8,553,148	98.45	85.66	-0.06
	Ordered	2,155,136	24.81	86.92	**1.41**
	Theta	2,993,225	34.45	85.32	-0.46
SNLI	Baseline	10,983,680	100.0	78.08	0
	EasyFirst	25,839,060	235.25	77.64	**-0.56**
	MiddleOut	29,655,888	270.00	75.31	-3.55
	Ordered	98,853,120	900.00	11.86	-84.81
	Theta	14,118,510	**128.54**	77.34	-0.95

is roughly 25% of what a typical training regiment would require.

Performance by Difficulty

Does using DCL-IRT for training lead to a more interpretable output in terms of test set performance? That is, when a curriculum is employed, does the model perform better on easier test examples than difficult ones? A comparison of methods on test data binned by difficulty shows that model performance is similar when test set difficulty is taken into account (Figure XX). The main difference is that for the baseline models, performance is stratified by difficulty almost immediately, and there are small improvements across groups during training, while for the CL methods, there is consistent improvement across groups as training continues, and more data is added.

Related work

Lots to do here. need a thorough lit review as part of this paper

Conclusion