Active Learning and Curriculum Learning

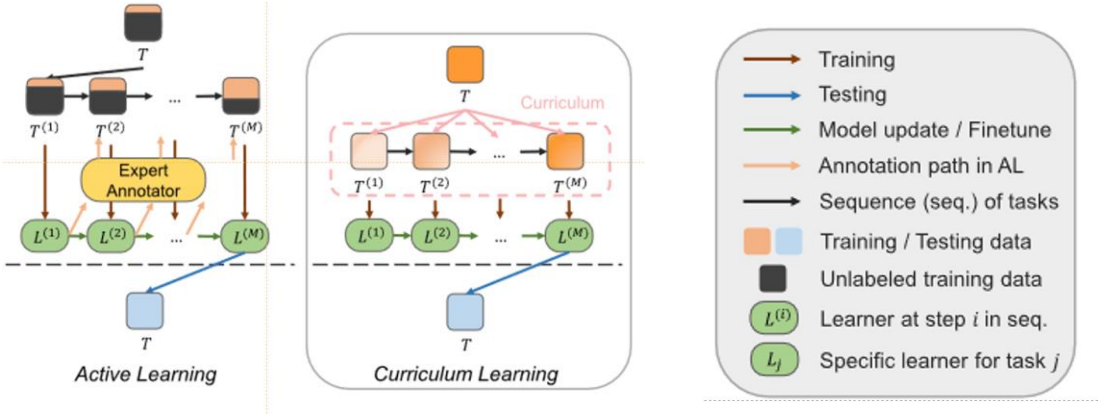
How to choose the best data to learn from

Antonio Carta

antonio.carta@unipi.it

Today's Lecture

- **Active learning**: given a large unlabeled dataset, choose the next sample to annotate
- **Curriculum learning**: given a dataset, find the optimal sampling order to learn from it



X. Wang et al."A Survey on Curriculum Learning." TPAMI 2022

Motivation

- Up to now, we assumed the data was given to us, without any control
- This is not true in many applications
  - We collect data over time (and we don’t throw the old one)
  - We annotate data over time
  - We can create a stream (curriculum) from a large dataset to improve training
- Each step can be improved if we can
  - Identify the best data to collect/annotate
  - Identify the best data the model should use for training

Motivation (2)

- We need to evaluate the impact of each sample on the resulting model
  - What are the properties of «important» samples?
  - Not all samples are equal
- We need to estimate their importance
  - A priori
  - Without labels

Example Forgetting in Offline/Joint Training

- **forgetting event**: sample x was classified correctly at iteration t and incorrectly at iteration $t + k$
- Observations:
  - certain examples are forgotten with high frequency (hard samples)
  - some not at all (easy samples)
  - easy samples generalize across neural architectures;
    - They are a property of the data, not the specific model
  - a significant fraction of (easy) examples can be omitted from the training data set without hurting the final accuracy

M. Toneva et al. “An Empirical Study of Example Forgetting during Deep Neural Network Learning.” ICLR 2019

Noisy Examples are Harder

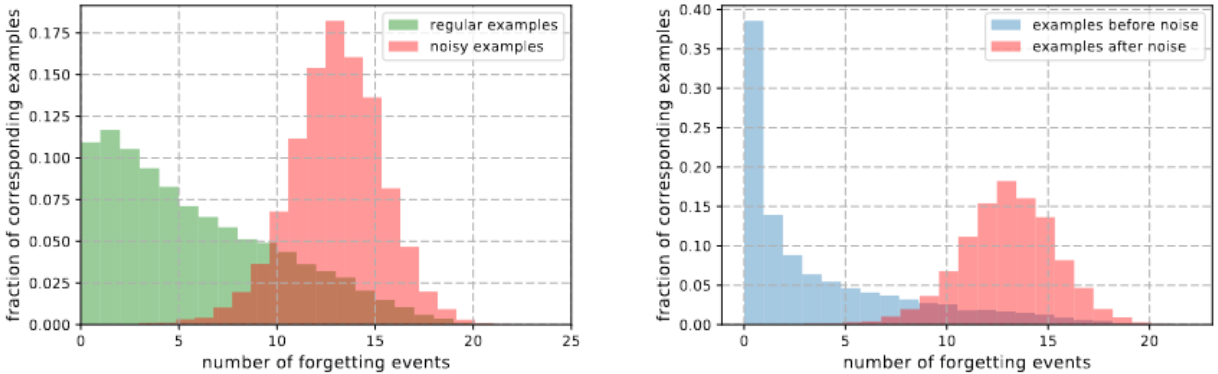


Figure 3: Distributions of forgetting events across training examples in *CIFAR-10* when 20% of labels are randomly changed. *Left.* Comparison of forgetting events between examples with noisy and original labels. The most forgotten examples are those with noisy labels. No noisy examples are unforgettable. *Right.* Comparison of forgetting events between examples with noisy labels and the same examples with original labels. Examples exhibit more forgetting when their labels are changed.

M. Toneva et al. "An Empirical Study of Example Forgetting during Deep Neural Network Learning." ICLR 2019

Easy Samples Can Be Removed

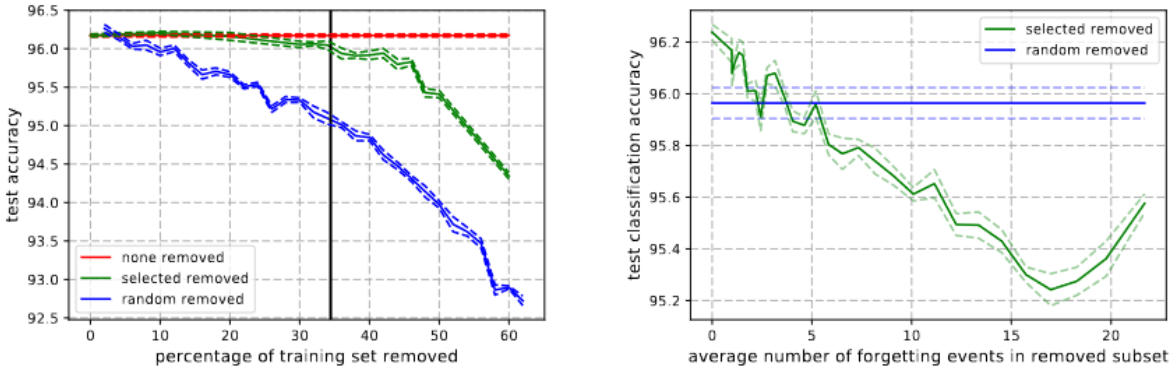


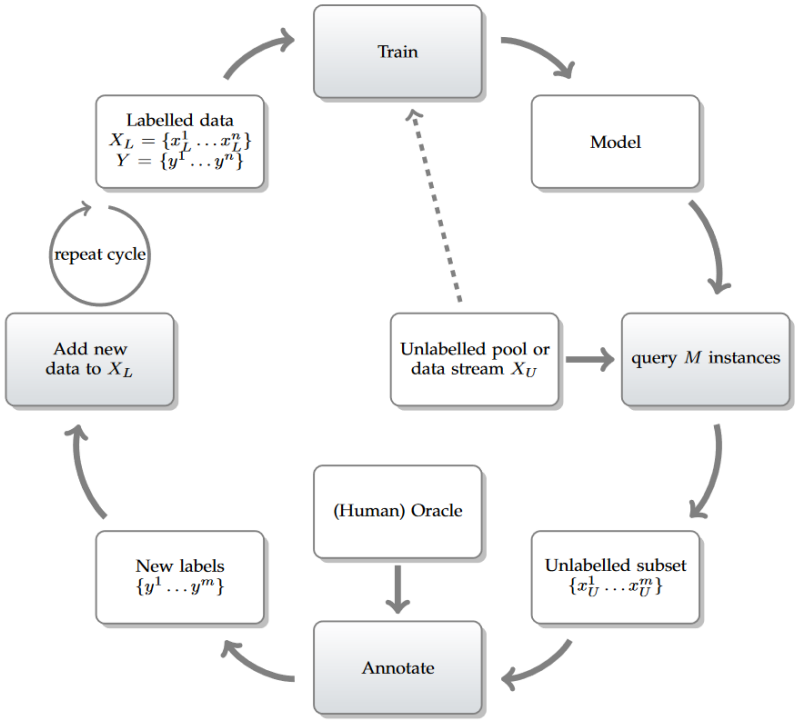
Figure 5: *Left* Generalization performance on *CIFAR-10* of ResNet18 where increasingly larger subsets of the training set are removed (mean +/- std error of 5 seeds). When the removed examples are selected at random, performance drops very fast. Selecting the examples according to our ordering can reduce the training set significantly without affecting generalization. The vertical line indicates the point at which all unforgettable examples are removed from the training set. *Right* Difference in generalization performance when contiguous chunks of 5000 increasingly forgotten examples are removed from the training set. Most important examples tend to be those that are forgotten the most.

M. Toneva et al. "An Empirical Study of Example Forgetting during Deep Neural Network Learning." ICLR 2019

Active Learning

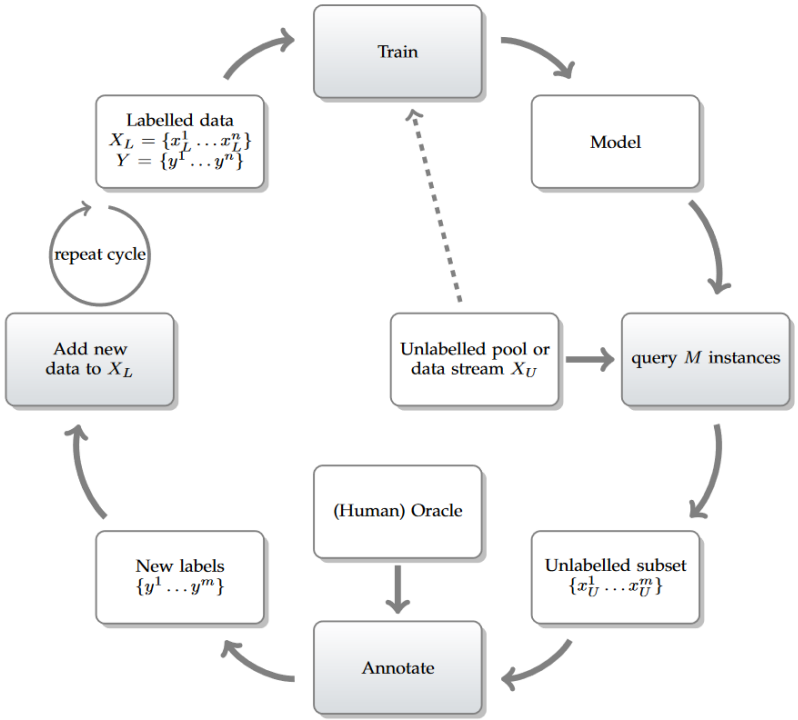
Active Learning

- We are given a large set of unlabeled data and a smaller subset of labeled data
- We train a model
- We want to improve the model: Which samples should we annotate?
- After the new data is annotated, retrain on all the data (new and old samples)



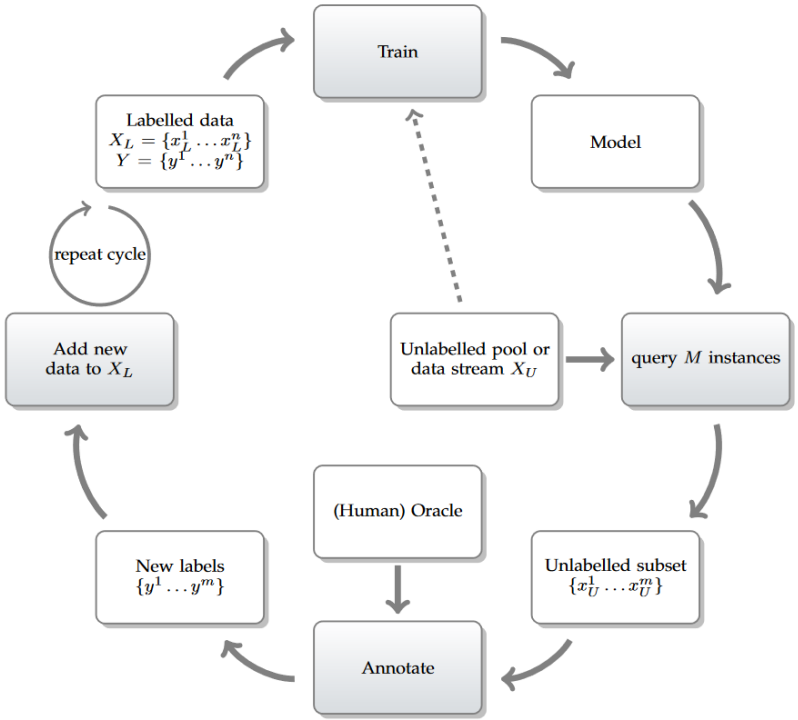
Active Learning

- **Acquisition function**: the method that chooses the samples and labels them
  - Inputs:
    - current model
    - set of unlabeled samples
    - Current set of labeled samples
  - Outputs: selected samples from the unlabeled set
- How do we design it?



Setup and Assumptions

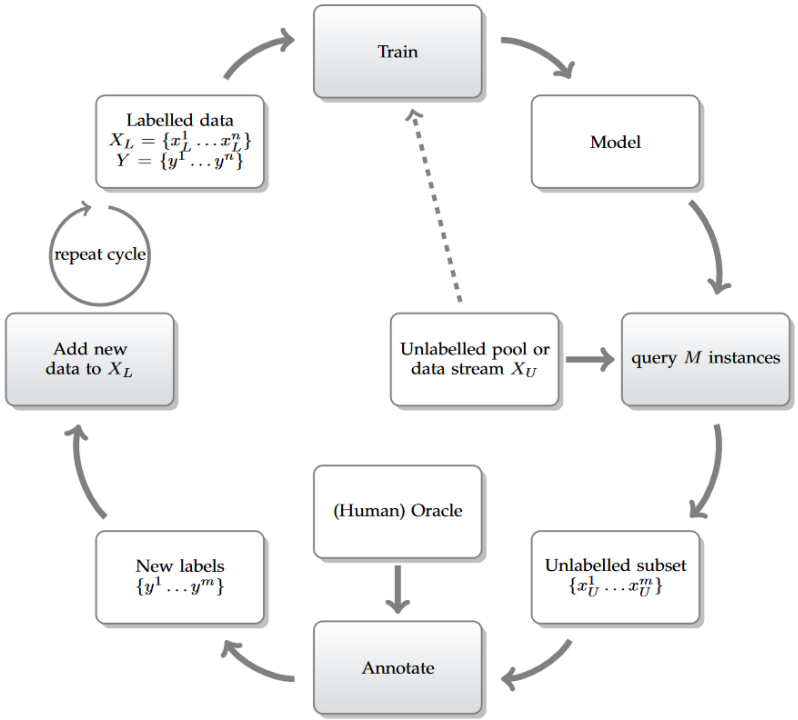
- **We have the entire (unlabeled) dataset available from the beginning**
  - Most methods can also be generalized to continual data collection
- **Cumulative training on all the data**
  - Unlike continual learning methods, we reuse all the data seen up to now
  - We ignore computational cost



Setup and Assumptions

Oracle/Annotators

- perfect oracle: annotators don't make mistakes
- In practical problems, human annotators are imperfect
- The best samples to annotate may also be the most difficult to annotate (and more expensive)
- Example: in many practical problems we ask multiple annotators when they may disagree (e.g. cancer data, translation)



Active Learning Methods

- **Version space reduction**: represent the entire subspace of hypotesis consistent with your data (version space) and refine it with the new data
- **Uncertainty and heuristics**: design measures to pick interesting samples
- **Coresets and representation learning**: model the data distribution and maximize coverage and diversity

Intuition based on Version Spaces

Version Space: the set of hypothesis (models) consistent with the training data
Learning with Version Spaces: restricting the space to solutions consistent with the training data
Active learning with Version Spaces: Picking the samples with more disagreement allows to find a smaller version space.

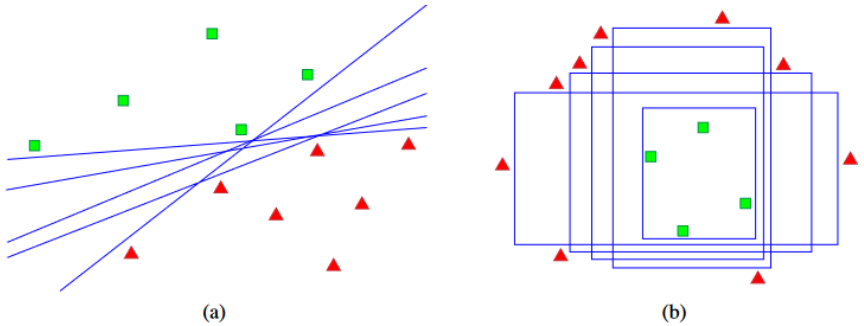


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, Burr. "Active Learning Literature Survey."

Heuristics

- Find a good measure of usefulness of the samples
  - Using the current model
  - That doesn’t the targets
- Which samples are more useful?
  - Hard samples
  - Samples where the model is less certain

Heuristics – Uncertainty Sampling

1. Create an initial classifier
2. While teacher is willing to label examples
   (a) Apply the current classifier to each unlabeled example
   (b) Find the b examples for which the classifier is least certain of class membership
   (c) Have the teacher label the subsample of b examples
   (d) Train a new classifier on all labeled examples

Figure 1. An algorithm for uncertainty sampling with a single classifier.

Lewis & Gale, "A Sequential Algorithm for Training Text Classifiers", ACM-SIGIR 1994

Heuristics – Entropy

- query samples with largest entropy

$$\mathrm{H}(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x) = \mathbb{E}[-\log p(X)]$$

- Low entropy: [0.1, 0.1, 0.8]
- High entropy: [0.3, 0.3, 0.4]

- What is the drawback of uncertainty/entropy-based sampling?

McKay, “Information-Based Objective Functions for Active Data Selection”, Neural Computation 4, 1992

Heuristic – Tradeoff

- What are the most uncertain samples?
  - Hard samples <- good, we want to label them
  - Noise <- useless
  - Outliers <- maybe useful, probably not the best choice
- **IDEA**: the best samples are hard enough but not too hard that may be noise/outliers
- It's a form of exploration/exploitation tradeoff. Too much novelty is not always the best choice because the most novel data is not always representative of the expected data

Joshi et al, "Multi-Class Active Learning for Image Classification", CVPR 2009

Model Calibration

- **Model calibration**: a model is calibrated if its probabilities are calibrated. That is, if a model predicts a class with probability p, then the prediction is correct p% of the time.

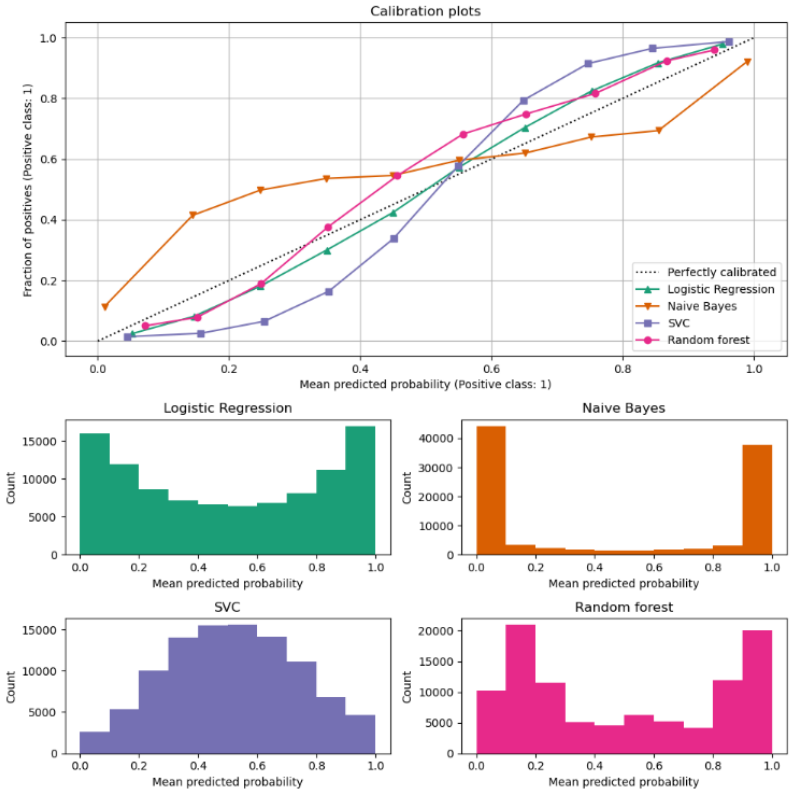
- Entropy-based heuristics assume that the output probabilities are correct and calibrated

- **In general, DNNs are not calibrated**
  - During training, we push the probabilities to 0/1, without any care about calibration
  - The difference between p=0.02 and p=0.1 is probably not significant, but it makes a big difference in entropy

Model Calibration – Examples

In the figure

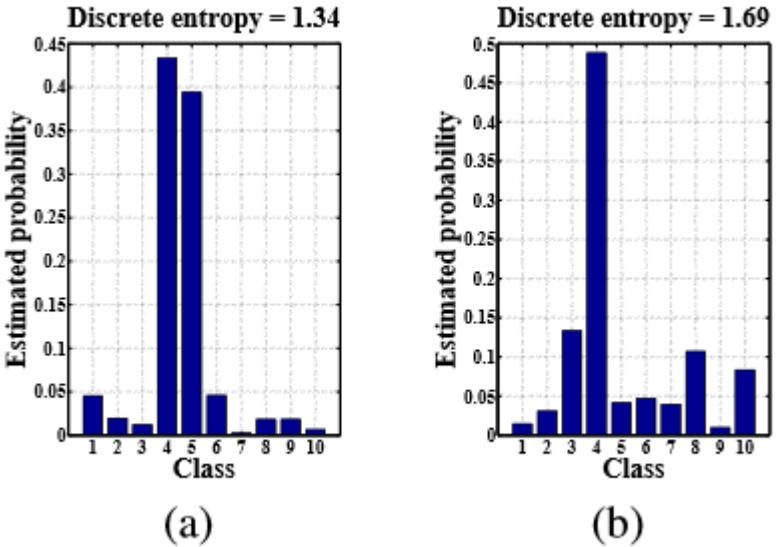
- Top figure: calibration plot
- Bottom figure: histogram of class probabilities
- Logistic regression is calibrated
- The others are not
- Naive Bayes is extremely uncalibrated. The histogram shows almost binary probabilites



1.16. Probability calibration — scikit-learn 1.4.2 documentation

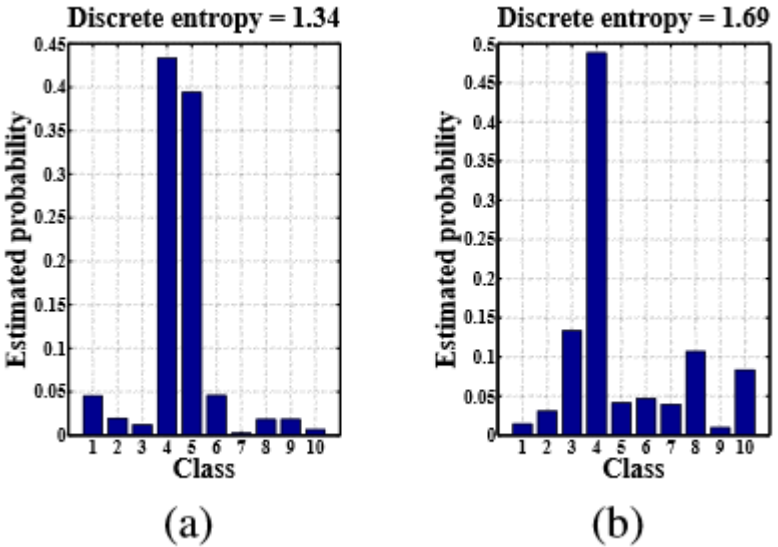
Heuristic – Best vs Second Best (BvsSB)

- Example: (a) has lower entropy than (b)
  - (a) is uncertain between two classes
  - (b) is confident on the best class
- Entropy is sensitive to:
  - large number of classes
  - Very low probabilities
- Uncalibrated models with a large number of classes will have unreliable probability values for most classes



Heuristic – Best vs Second Best (BvsSB)

- **Best-versus-Second-Best**: difference between highest probability and second best
- Alternative measure of uncertainty
- More robust to number of classes and smaller values for unimportant classes



Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Empirical Results

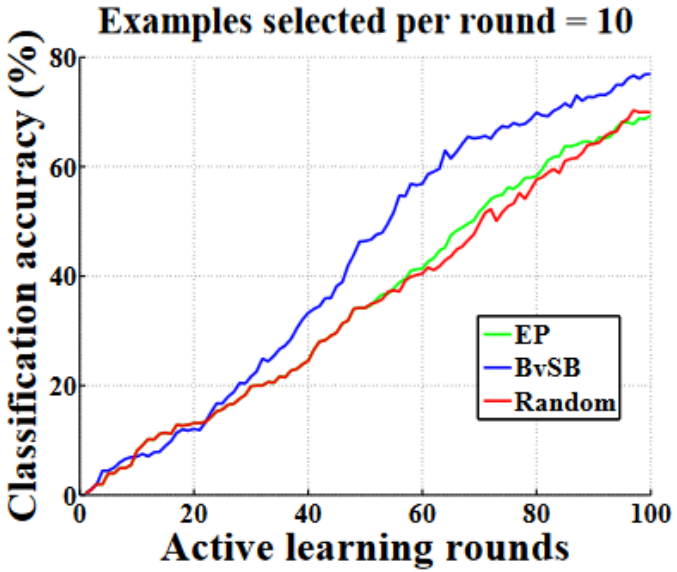


Figure 4. Active learning on the Caltech-101 dataset.

Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Query by Committee

- Committee of models $\mathcal{C} = \{\theta^{(1)}, \dots, \theta^{(C)}\}$
- Each votes on the labeling of new samples
- IDEA: pick the instances with most disagreement
- Example of metric: Vote Entropy $x^*_{VE} = \underset{x}{\text{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$
  - $V(y_i)$ number of votes for label i
  - C committee size

H.S. Seung et al. “Query by committee”. ACM Workshop on Computational Learning Theory, 1992.

Active Learning – Coresets

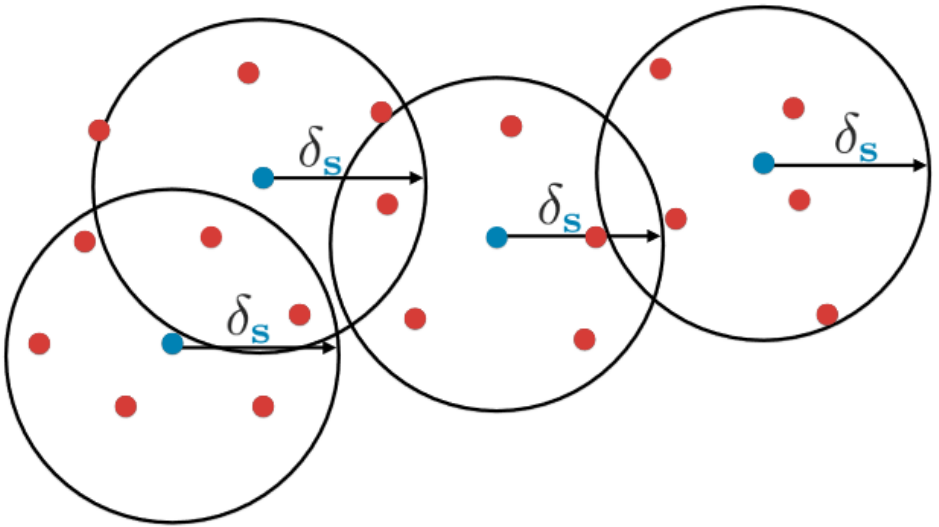
Coresets

- Uncertainty estimates are not robust against outliers and noise
- Uncertainty estimates may be inaccurate at the beginning of training
- Different approach: try to annotate diverse samples

- Questions
  - We need to model the labeled and unlabeled data distributions
  - We need to pick the diverse samples from unlabeled data

Coresets

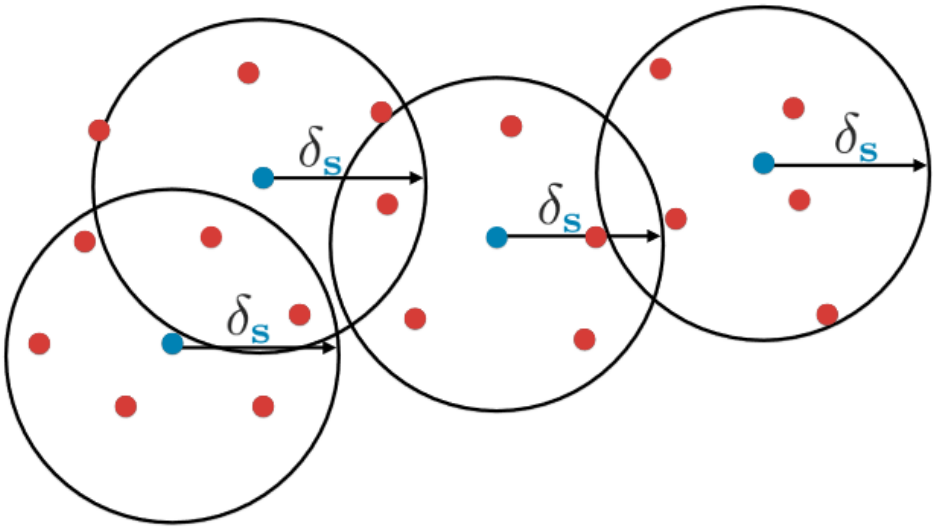
- We need to select a bounded set of representative samples
- Purely unsupervised selection, based only on the distances



K-center problem

Combinatorial optimization problem

- Given n cities (samples)
- We want to build k warehouses (selected samples)
- Minimize the maximum distance between a city and a wharehouse



In our case, we select «representative samples»

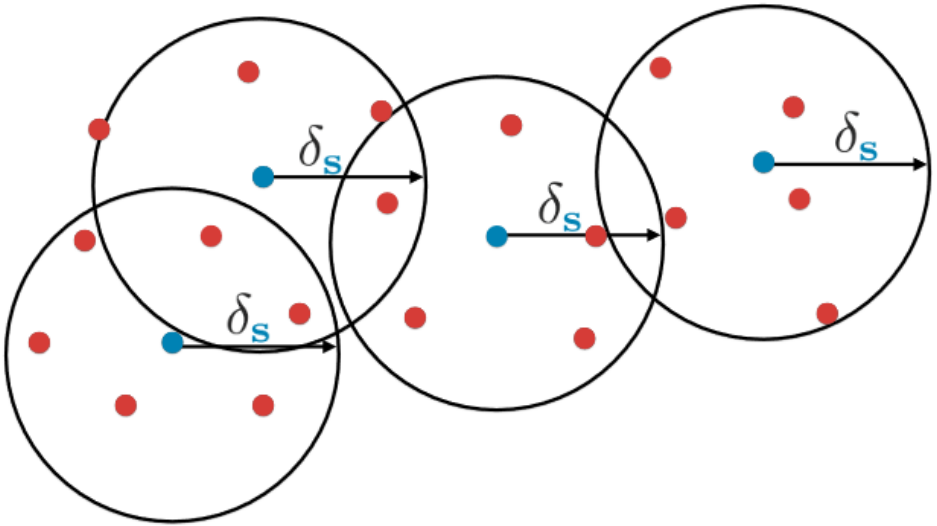
https://en.wikipedia.org/wiki/Metric_k-center

K-center problem

Formal definition:

- Input: a set $\mathbf{V} \subseteq \mathcal{X}$, and a parameter k.
- Output: a set $\mathcal{C} \subseteq \mathbf{V}$ of k points.
- Goal: Minimize the cost
  $r^{\mathcal{C}}(\mathbf{V}) = \max_{v \in V} \mathrm{d}(\mathrm{v}, \mathcal{C})$

This is an NP-hard problem



https://en.wikipedia.org/wiki/Metric_k-center

K-center Greedy

- Approximate solution with greedy algorithm
- Repeat until you have enough samples
  - Given s current pool
  - Find the most distant point u
  - Add it to the pool

Algorithm 1 k-Center-Greedy

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and a budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

Results

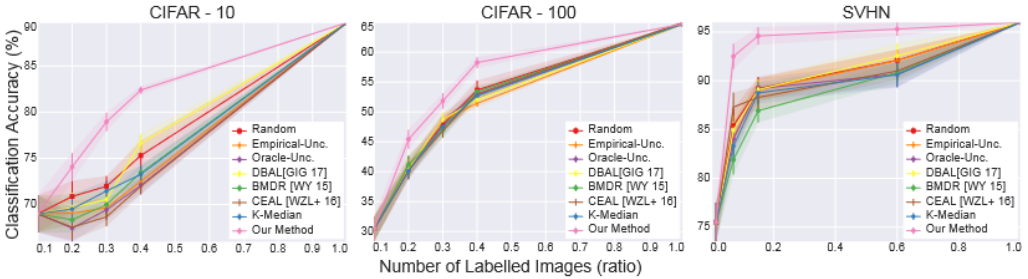


Figure 3: Results on Active Learning for Weakly-Supervised Model (error bars are std-dev)

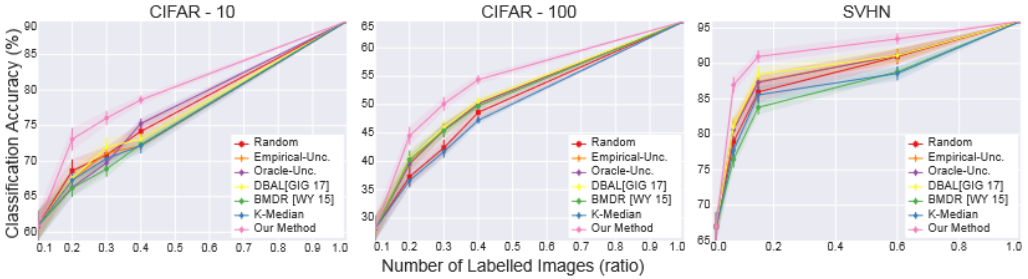


Figure 4: Results on Active Learning for Fully-Supervised Model (error bars are std-dev)

VAAL – Variation Adversarial Active Learning

- **IDEA**: pool-based semi-supervised active learning algorithm that implicitly learns this sampling mechanism in an adversarial manner
- does not depend on the performance of the task
- learns a latent space using a variational autoencoder (VAE) and an adversarial network trained to discriminate between unlabeled and labeled data

S. Samarth et al. “Variational Adversarial Active Learning.”

VAAL

- VAE tries to trick the adversarial network into predicting that all data points are from the labeled pool
- the adversarial network learns how to discriminate between dissimilarities in the latent space

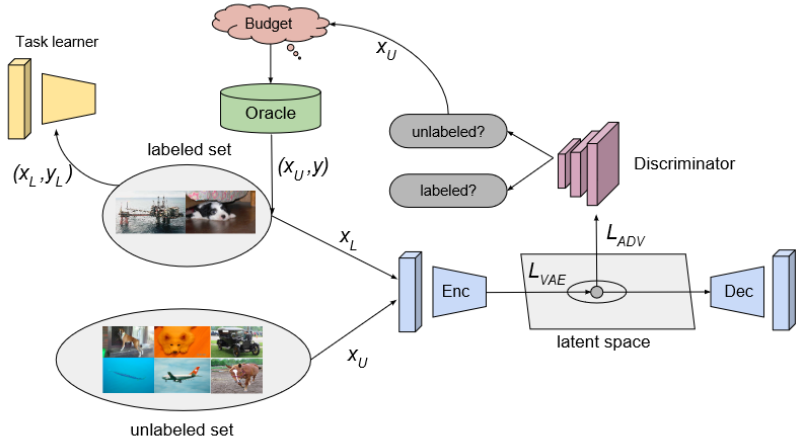


Figure 1. Our model (VAAL) learns the distribution of labeled data in a latent space using a VAE optimized using both reconstruction and adversarial losses. A binary adversarial classifier (discriminator) predicts unlabeled examples and sends them to an oracle for annotations. The VAE is trained to fool the adversarial network to believe that all the examples are from the labeled data while the discriminator is trained to differentiate labeled from unlabeled samples. Sample selection is entirely separate from the main-stream task for which we are labeling data inputs, making our method to be *task-agnostic*

S. Samarth et al. "Variational Adversarial Active Learning."

VAAL - VAE Loss

- (X_L, Y_L) labeled set
- X_U unlabeled pool
- β-VAE
  - $$\mathcal{L}_{\text{VAE}}^{\text{trd}} = \mathbb{E}[\log p_\theta(x_L \mid z_L)] - \beta \mathrm{D}_{\mathrm{KL}}\big(q_\phi(z_L \mid x_L) \parallel p(z)\big) + \mathbb{E}[\log p_\theta(x_U \mid z_U)] - \beta \mathrm{D}_{\mathrm{KL}}\big(q_\phi(z_U \mid x_U) \parallel p(z)\big)$$
  - learns to encode both labeled and unlabeled data in the same latent space
  - q_ϕ encoder p_θ decoder $p(z)$ prior

VAAL – Adversial Loss and Discriminator

- $\mathcal{L}_{\text{VAE}}^{adv} = -\mathbb{E}\left[\log\left(D\left(q_\phi(z_L \mid x_L)\right)\right)\right] - \mathbb{E}\left[\log\left(D\left(q_\phi(z_U \mid x_U)\right)\right)\right]$
  - *VAE tries to «fool» the discriminator, mapping labeled and unlabeled data in the same class*
  - *Total VAE loss* $\mathcal{L}_{\text{VAE}} = \lambda_1 \mathcal{L}_{\text{VAE}}^{\text{trd}} + \lambda_2 \mathcal{L}_{\text{VAE}}^{adv}$

- $\mathcal{L}_D = -\mathbb{E}\left[\log\left(D\left(q_\phi(z_L \mid x_L)\right)\right)\right] - \mathbb{E}\left[\log\left(1 - D\left(q_\phi(z_U \mid x_U)\right)\right)\right]$
  - Discriminator D learns to discriminate labeled data (left) from unlabeled (right)

VAAL – Training

Algorithm 1 Variational Adversarial Active Learning

Input: Labeled pool (X_L, Y_L), Unlabeled pool (X_U), Initialized models for θ_T, θ_{VAE}, and θ_D

Input: Hyperparameters: epochs, λ_1, λ_2, α_1, α_2, α_3

for $e = 1$ to epochs **do**
  sample $(x_L, y_L) \sim (X_L, Y_L)$
  sample $x_U \sim X_U$
  Compute $\mathcal{L}_{\text{VAE}}^{trd}$ by using Eq. 1
  Compute $\mathcal{L}_{\text{VAE}}^{adv}$ by using Eq. 2
  $\mathcal{L}_{\text{VAE}} \leftarrow \lambda_1 \mathcal{L}_{\text{VAE}}^{trd} + \lambda_2 \mathcal{L}_{\text{VAE}}^{adv}$
  Update VAE by descending stochastic gradients:
  $\theta'_{VAE} \leftarrow \theta_{VAE} - \alpha_1 \nabla \mathcal{L}_{\text{VAE}}$
  Compute $\mathcal{L}_{\text{D}}$ by using Eq. 3
  Update D by descending its stochastic gradient:
  $\theta'_D \leftarrow \theta_D - \alpha_2 \nabla \mathcal{L}_{\text{D}}$
  Train and update T:
  $\theta'_T \leftarrow \theta_T - \alpha_3 \nabla \mathcal{L}_{\text{T}}$
end for
return Trained $\theta_T, \theta_{VAE}, \theta_D$

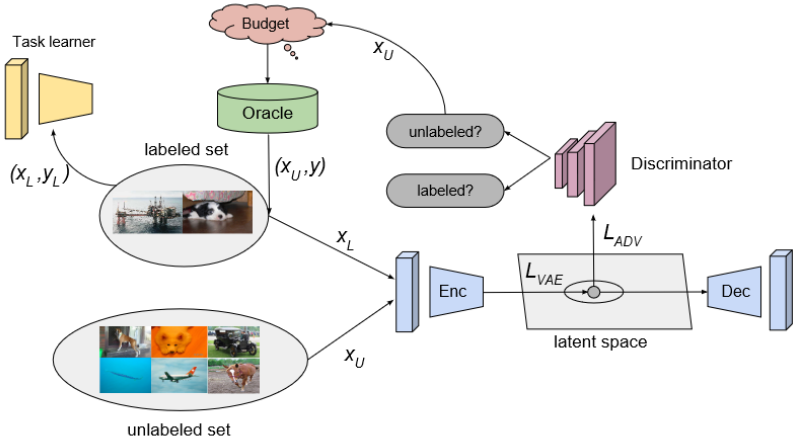


Figure 1. Our model (VAAL) learns the distribution of labeled data in a latent space using a VAE optimized using both reconstruction and adversarial losses. A binary adversarial classifier (discriminator) predicts unlabeled examples and sends them to an oracle for annotations. The VAE is trained to fool the adversarial network to believe that all the examples are from the labeled data while the discriminator is trained to differentiate labeled from unlabeled samples. Sample selection is entirely separate from the main-stream task for which we are labeling data inputs, making our method to be *task-agnostic*

S. Samarth et al. "Variational Adversarial Active Learning."

VAAL – Sampling

- **Sampling**: We use the probability associated with the discriminator's predictions as a score
  - collect b samples in every batch
  - Select those where the discriminator has the lowest confidence
  - Annotate them with the oracle.

Algorithm 2 Sampling Strategy in VAAL

Input: b, X_L, X_U

Output: X_L, X_U

1: Select samples (X_s) with $\min_b\{\theta_D(z_U)\}$
2: $Y_o \leftarrow \mathcal{ORACLE}(X_s)$
3: $(X_L, Y_L) \leftarrow (X_L, Y_L) \cup (X_s, Y_o)$
4: $X_U \leftarrow X_U - X_s$
5: **return** X_L, X_U

S. Samarth et al. "Variational Adversarial Active Learning."

VAAL - Results

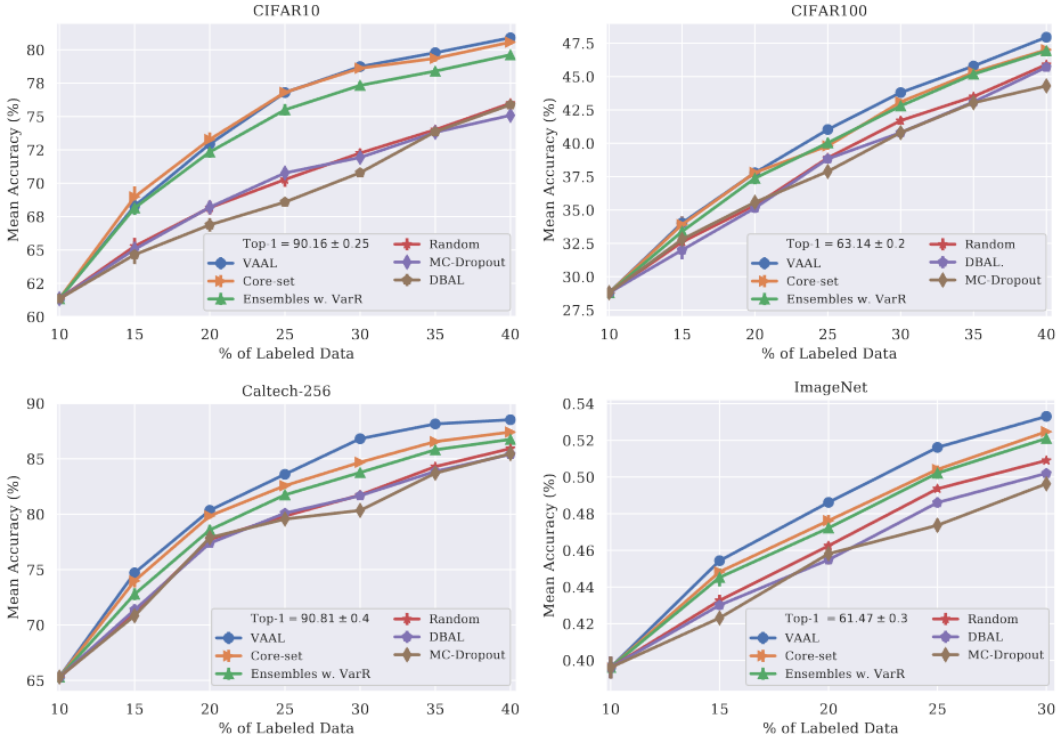


Figure 2. VAAL performance on classification tasks using CIFAR10, CIFAR100, Caltech-256, and ImageNet compared to Core-set [43], Ensembles w. VarR [1], MC-Dropout [13], DBAL [16], and Random Sampling. Best visible in color. Data and code required to reproduce are provided in our code repository

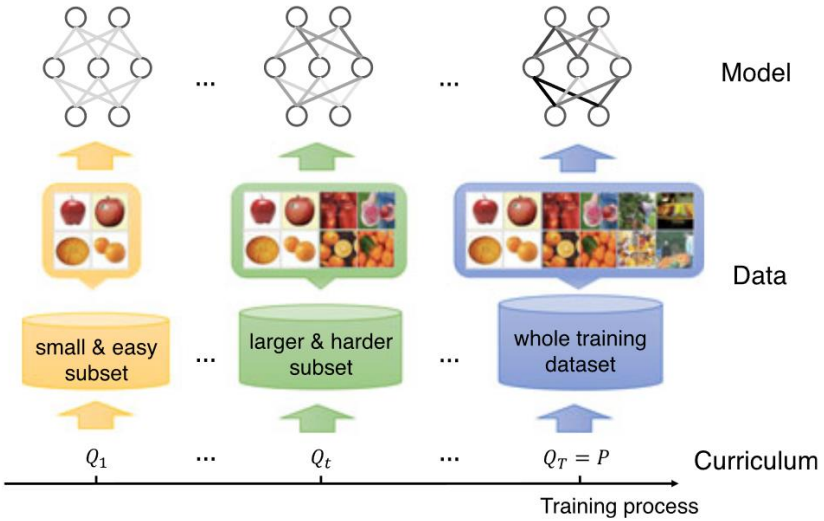
S. Samarth et al. “Variational Adversarial Active Learning.”

Curriculum Learning (CL)

Motivations

- Humans learn in curricula: from the easier data to the harder data
- **Q**: Can a DNN benefit from a designed curricula?
- **A**: yes, but not always

- **IDEA**: Order data in experiences by difficulty
  - OBJECTIVE 1: Improve final accuracy
  - OBJECTIVE 2: Improve convergence speed



X. Wang et al."A Survey on Curriculum Learning." TPAMI 2022

Motivations

Optimization perspective

- The task is hard to optimize and training in a single step results in a poor solution
- CL guides towards better parameters by learning easier subtasks first, then generalizing them to the harder and general problem
- Examples: RL with sparse rewards, GAN training

Denoising Perspective

- Tasks is noisy, poor and uneven quality of the annotations, heterogenous (e.g. cheap large-scale data)
- Learning on cleaner subsets first improve the early phase of training
- Examples: neural machine translation, unsupervised learning

Definition - Curriculum Learning

- A curriculum is a sequence of training criteria over T training steps: $\mathcal{C} = \langle Q_1, \ldots, Q_t, \ldots, Q_T \rangle$

- Each criterion Q_t is a reweighting of the target training distribution $Q_t(z) \propto W_t(z)P(z) \;\; \forall \text{ example } z \in \text{training set } D$

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Definition - Curriculum Learning (2)

$$Q_t(z) \propto W_t(z)P(z)$$

- The entropy of distributions gradually increases $H(Q_t) < H(Q_{t+1})$
  - Diversity and information should gradually increase
  - Gradually introduce new/harder concepts
- The weight for any example increases $W_t(z) \leq W_{t+1}(z) \quad \forall z \in D$
  - The training data grows over time
  - Old samples remain available (cumulative training)
- The final weighting corresponds to the real distribution $Q_T(z) = P(z)$
  - Training ends on the real data distribution

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

CL Design

Two components:

- **Difficulty Measurer**: decides the difficulty of each sample
- **Training Scheduler**: decides the sequence of data subsets

Predefined Curriculum – Difficulty Measures

- Difficulty Measurer is hand-made
- No data-driven decision
- **Complexity**: structural complexity of the samples
  - Example: sentence length
- **Diversity**: too much diversity at sample/group/element level
  - Example: rare words, entropy
- **Noise estimation**: clean data is easier
  - Example: Different data collection methods result in different noise levels

Difficulty Measurer*	Angle	Data Type	∝ Easy
Sentence length [86], [107]	Complexity	Text	–
Number of objects [122]	Complexity	Images	–
# conj. [50], #phrases [113]	Complexity	Text	–
Parse tree depth [113]	Complexity	Text	–
Nesting of operations [131]	Complexity	Programs	–
Shape variability [6]	Diversity	Images	–
Word rarity [50], [86]	Diversity	Text	–
POS entropy [113]	Diversity	Text	–
Mahalanobis distance [14]	Diversity	Tabular	–
Cluster density [11], [31]	Noise	Images	+
Data source [10]	Noise	Images	/
SNR/SND [7], [89]	Noise	Audio	–
Grammaticality [66]	Domain	Text	+
Prototypicality [113]	Domain	Text	+
Medical based [44]	Domain	X-ray film	/
Retrieval based [18], [82]	Domain	Retrieval	/
Intensity [30]/Severity [111]	Intensity	Images	+
Image difficulty score [106], [114]	Annotation	Images	–
Norm of word vector [68]	Multiple	Text	–

** Abbreviations: POS = Part Of Speech, SNR = Signal to Noise Ratio, SND = Signal to Noise Distortion, Domain = Domain knowledge, # conj. = number of coordinating conjunctions.*
The "+" in ∝ Easy means the higher the measured value, the easier the data example, and the "–" has the opposite meaning.

X. Wang et al."A Survey on Curriculum Learning." TPAMI 2022

Predefined Curriculum – Training Scheduler

- **Discrete**: update data after a fixed number of epochs or convergence on the current subset
- **Continuous**: update data after each epoch

Baby Step Algorithm

- Hand-made schedule
- Cumulative training
- Goes to next step after a fixed number of epochs or convergence

Algorithm 1. The Baby Step Training Scheduler [12]

Input: $\mathcal{D}$: training dataset; $\mathcal{C}$: the Difficulty Measurer;
Output: M^*: the optimal model.

$\mathcal{D}' = \text{sort}(\mathcal{D}, \mathcal{C})$;
$\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^k\} = \mathcal{D}'$ where $\mathcal{C}(d_a) < \mathcal{C}(d_b),\ d_a \in \mathcal{D}^i, d_b \in \mathcal{D}^j,\ \forall i < j$;
$\mathcal{D}^{train} = \emptyset$;
for $s = 1 \cdots k$ **do**
    $\mathcal{D}^{train} = \mathcal{D}^{train} \cup \mathcal{D}^s$;
    while not converged for p epochs **do**
        train$(M, \mathcal{D}^{train})$;
    end while
end for

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Automatic Curriculum

- **Self-paced-learning (SPL)**: the students measures the difficulty according to its loss
- **Transfer Teacher**: strong pretrained teacher measures the difficulty
- **RL Teacher**: models curriculum design as a reinforcement learning problem, using the student feedback as reward

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Self-Paced Learning

- **IDEA**: at each step, train the model only on the p% easiest examples
- Measure difficulty using the loss for each sample
- Increase the percentage of data over time
- **Automatic**:
  - the difficulty is measured directly by the student
  - Embeds the curriculum design in the learning objective

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Self-Paced Learning

- l_i *loss for sample i*
- $g(\boldsymbol{v}; \lambda)$ self-paced regularizer
  - $\boldsymbol{v}$ is a vector of weights for each sample
  - λ the age coefficient
- **Learning objective**:

$$\min_{\boldsymbol{w};\boldsymbol{v}\in[0,1]^N} \mathbb{E}(\boldsymbol{w}, \boldsymbol{v}; \lambda) \sum_{i=1}^{N} v_i l_i + g(\boldsymbol{v}; \lambda).$$

- Example: l1 regularizer $g(\boldsymbol{v}; \lambda) = -\lambda \sum_{i=1}^{N} v_i$

Self-Paced Learning

$$g(\boldsymbol{v};\lambda) = -\lambda \sum_{i=1}^{N} v_i$$

- Training: Alternated optimization
  - Optimize v keeping w fixed $v_i^* = \arg\min_{v_i \in [0,1]} v_i l_i + g(v_i;\lambda)$
    - Convex objective. We can find the optimal solution
  - Optimize w keeping v fixed $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{N} v_i^* l_i$
    - SGD step with backpropagation

- Solution for v_i^*
  - $v_i^* = \begin{cases} 1, & l_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
  - Hard binary choice of samples

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Self-Paced Learning

Algorithm 2. Self-Paced Learning

Input: $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$: training dataset; f: the machine learning model; T: the maximum number of iterations;

Output: w: the optimal parameters of f.

1: Initialize $\boldsymbol{w}$, $\boldsymbol{v}$, $\lambda = \lambda_0$, $t = 0$.
2: **while** $t \neq T$ **do**
3: $t = t + 1$;
4: Update $\boldsymbol{v}^*$ by Eq. (9);
5: Update $\boldsymbol{w}^*$ by Eq. (10);
6: Update λ to a larger value; // to include harder data
7: **end while**

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

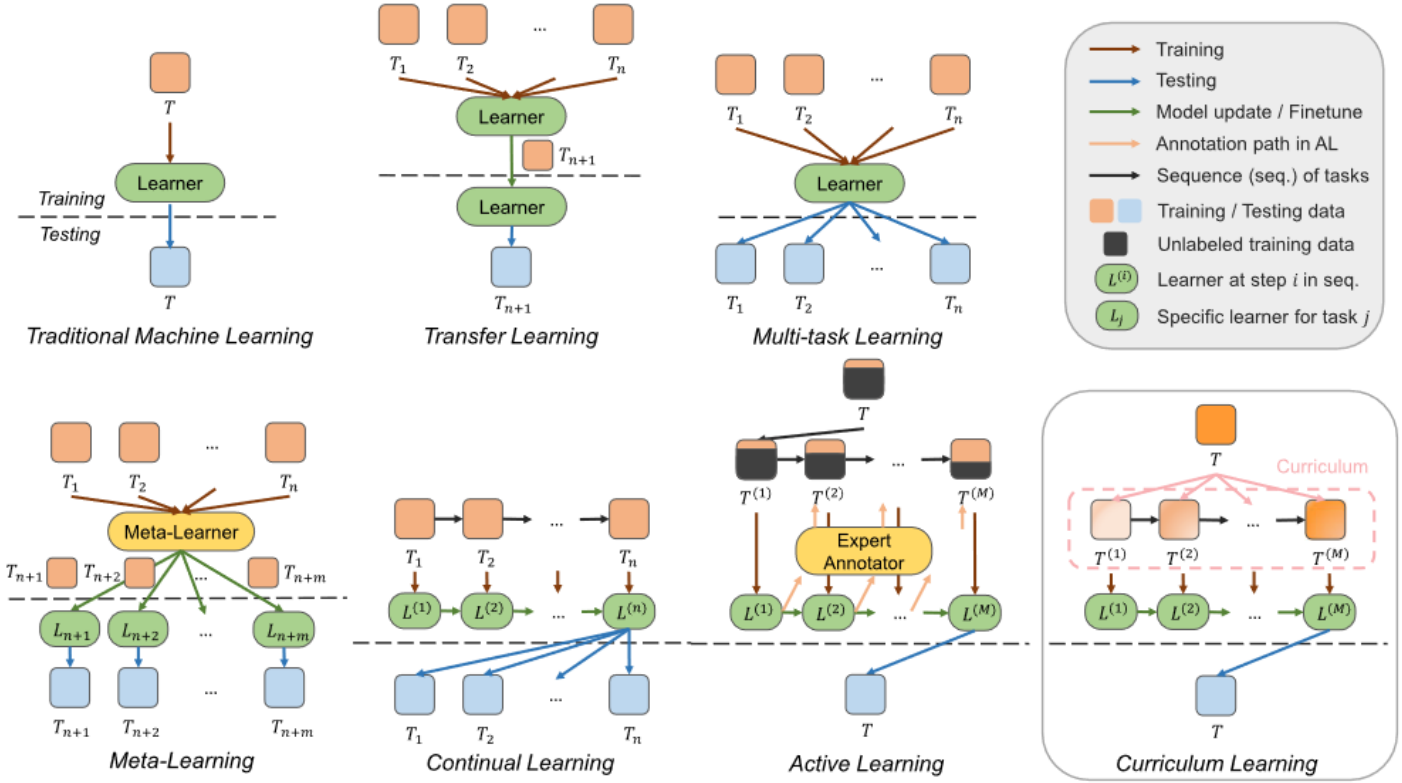
Curriculum Order – Hard-to-Easy

Is «easy to hard» the best choice?

- Sometimes, the **anti-curriculum** (hard-to-easy) is better
- Related: **Hard Example Mining** (HEM)
  - Sampling the harder examples can improve convergence rate
  - We have seen some examples in self-supervised learning
- As always, the optimal solution depends on the domain
  - Example: if your hard examples are too hard (e.g. sparse RL rewards) the easy-to-hard curriculum helps

X. Wang et al.“A Survey on Curriculum Learning.” TPAMI 2022

Overview



X. Wang et al."A Survey on Curriculum Learning." TPAMI 2022

Take-Home Messages

Often we can control the data acquisition

- Find the best examples to annotate
  - Via uncertainty or difficulty
  - By optimizing distribution coverage and diversity

- Craft better data orders
  - Curricula to learn in complex optimization problems (GAN, RL)
  - Improve convergence speed by
    - removing easy samples
    - learning easy samples first
    - mining hard examples

References

- Papers in the footnotes
- Open World Lifelong Learning
  A Continual Machine Learning Course
  - https://owll-lab.com/teaching/cl_lecture/
  - Recordings are available
  - The organization of these slides is partly based on this course
- More active learning methods:
  https://lilianweng.github.io/posts/2022-02-20-active-learning/