DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation

Wendi Li, Xiao Yang, Weiqing Liu, Yingce Xia, Jiang Bian

Background

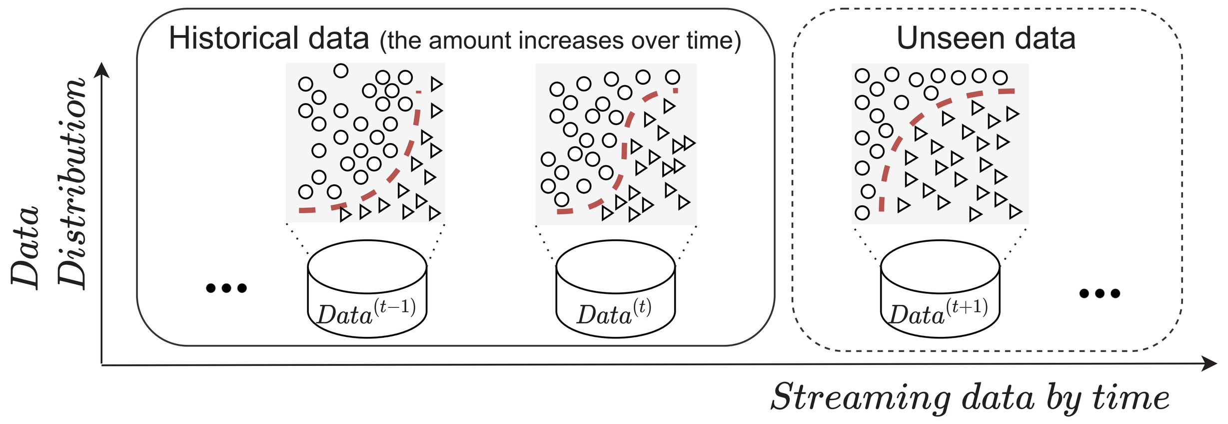

Figure 1: An example of concept drifts on streaming data.

Due to the non-stationary nature of the real-world environment, the data distribution could keep changing with continuous data streaming over time, this is a research problem called **concept drift**.
Concept drift causes the distribution gap in time-series data between different periods, which may result in a performance drop of the model trained on the historical data when making predictions on unseen future data.

Motivation

Concept drift is often hard to predict. Previous works adapt to the new concept **after the concept drifts**.

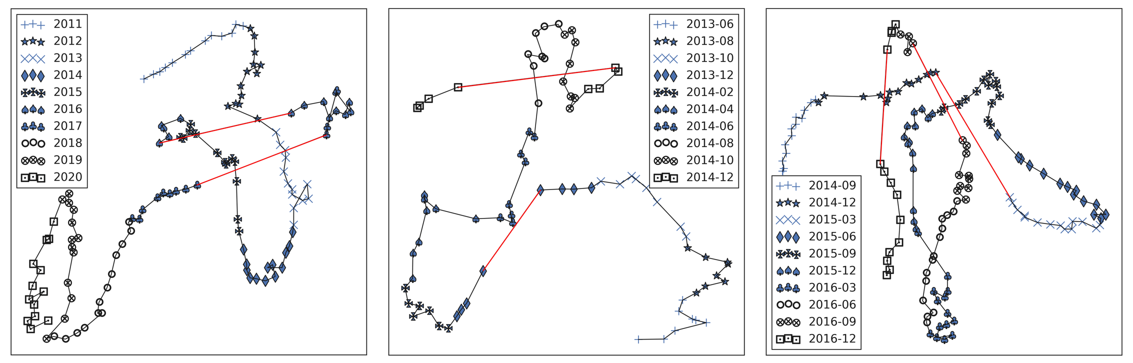

Figure 2: The visualization of concept drifts in three real-world datasets. Points represent the data distribution at a timestamp and are connected chronologically. The closer 2 points are, the more similar their distributions are.

However, in real-world scenarios, most concept drifts have a non-random trend rather than completely random, which is shown in Figure 2.
DDG-DA tries to predict and help forecasting models adapt to the new concept **before it happens**.

Contribution

To sum up, our contributions include

1. DDG-DA is the first method to model the evolving of data distribution in predictable concept drift scenarios.
2. We create a differentiable distribution distance to learn DDG-DA and provide related theoretical analysis.
3. Extensive experiments on different real-world concept-drift-predictable scenarios.

Contact Information

The open-source version of DDG-DA can be found on *Qlib*. *Qlib* provides a framework that supports research on concept drift, especially in financial scenarios.

Code https://github.com/microsoft/qlib/tree/main/examples/benchmarks_dynamic/DDG-DA

Emails Wendi.Li@wisc.edu, Xiao.Yang@microsoft.com, Weiqing.Liu@microsoft.com, Yingce.Xia@microsoft.com, Jiang.Bian@microsoft.com

Method Design — Framework

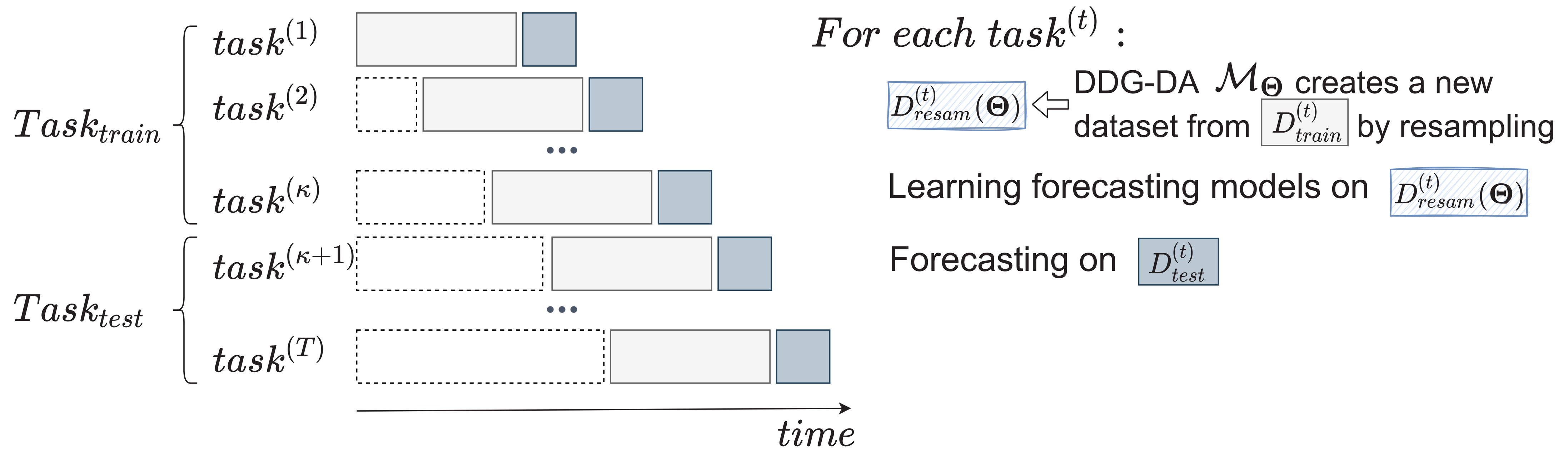

Figure 3: Training data (historical data) and test data (recent unseen data) change over time; the objective of each task is to improve the forecasting performance on test data.

Defining Task
Forecasting models are learned on historical data and make predictions on future data. To handle concept drift, forecasting models are adapted over time. Each chance to adapt forecasting models is called a *task* in our method.

Learning to Predict Data Distribution
For each task, DDG-DA will predict the future data distribution and create a new dataset from the training data by resampling. The data distribution of the resampled new dataset is expected to be closer to the future test dataset. Then, forecasting models will be learned on the new resampled dataset and make predictions on the test dataset in the future. DDG-DA will learn to predict data distribution on training tasks and inference on test tasks.

Method Design — Optimization

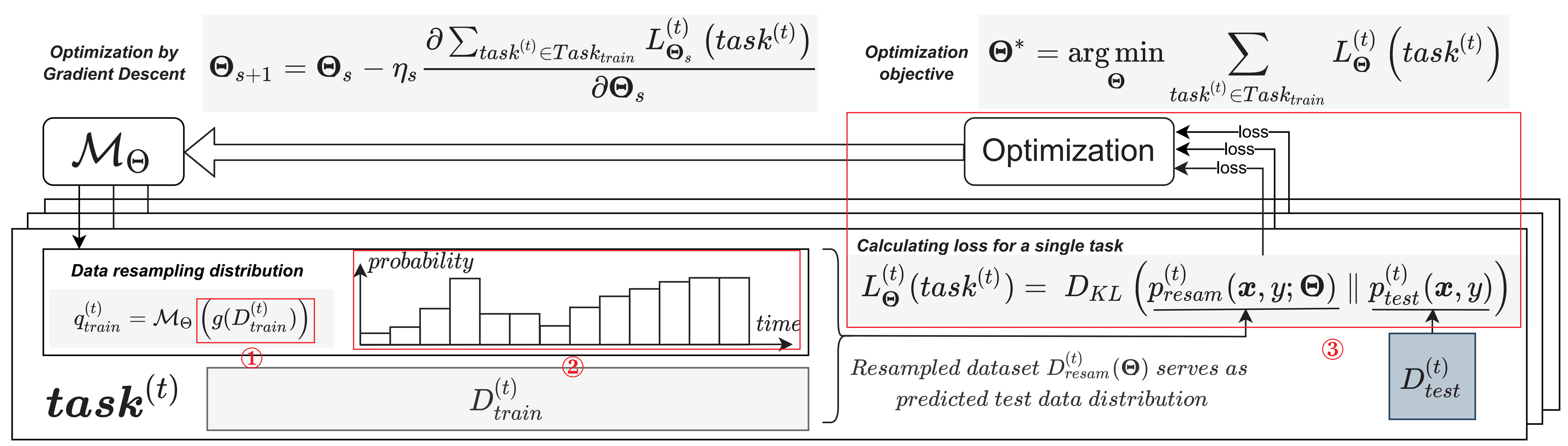

Figure 4: The learning process of DDG-DA; DDG-DA $\mathcal{M}_\Theta$ learns to guide the training process of forecasting model by generating dataset $D^{(t)}_{resam}(\Theta)$ resampled from $D^{(t)}_{train}$ with probability $q^{(t)}_{train}$. $q^{(t)}_{train}$ is the resampling probability given by $\mathcal{M}_\Theta$ at timestamp t.

The learning process of DDG-DA is shown in Figure 4.

① Extract historical data distribution information as features.
② DDG-DA outputs the data resampling probability. The objective of DDG-DA is to minimize the distribution difference between the resampled data and the future test data.
③ The data distribution difference becomes the loss. DDG-DA will be optimized based on the loss.

To make the loss differentiable, we first use a proxy model to approximate data distribution and formulate the loss as

$$\arg\min_{\Theta} \sum_{task^{(t)}\in Task_{train}} \left(\sum_{(x,y)\in D^{(t)}_{test}} \|y_{proxy}(x;\phi^{(t)}) - y\|^2 \right) \quad (1)$$
$$\text{s.t.} \quad \phi^{(t)} = \arg\min_{\phi} \sum_{(x',y')\in D^{(t)}_{resam}(\Theta)} \|y_{proxy}(x';\phi) - y'\|^2$$

Second, DDG-DA adopts models with closed-form solutions in the lower-level optimization and makes the loss differentiable.

Case Study

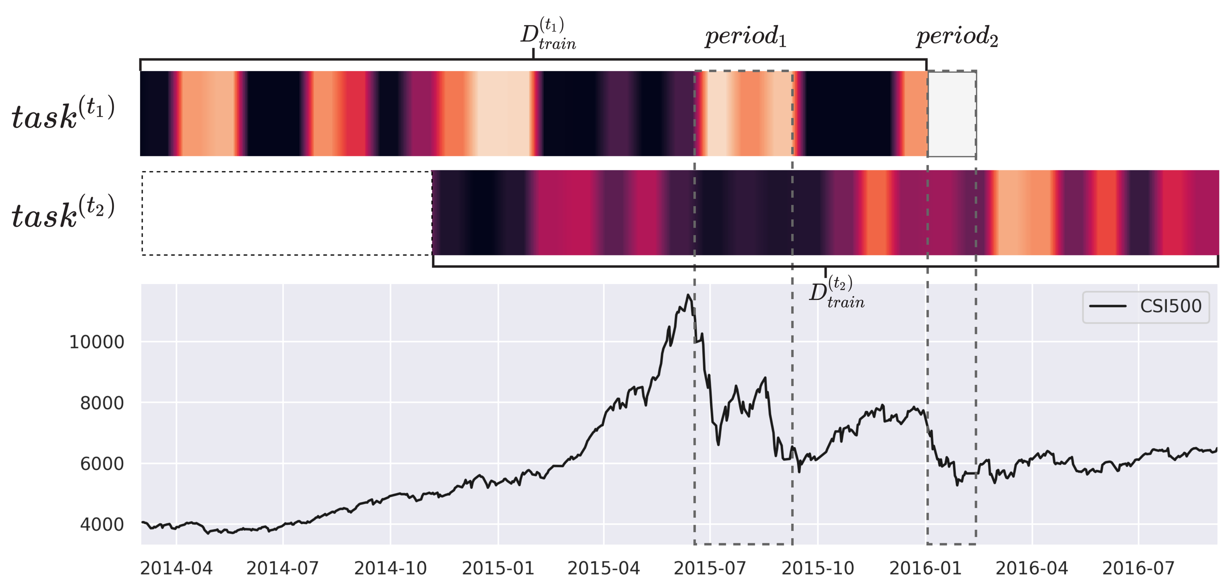

Figure 5: The sample probability is given by DDG-DA on the stock forecasting task in different tasks. The brighter the color, the greater the probability.

In each task, DDG-DA gives different resampling probability to data in different periods.
In different tasks, the data resampling probability distribution changes dynamically according to the stock market state in that period.

Experiment

The experiments in Table 1 are conducted on multiple datasets in three real-world scenarios(forecasting on stock price trends, electricity load, and solar irradiance). In most of the metrics, our method achieves the best performance.
The following methods are compared:

- **RR**: Periodically **R**olling **R**etrain model on data in memory with equal weights.
- **GF-Lin / GF-Exp**: Based on RR, **G**radual **F**orgetting by weights decaying **Lin**early / **Exp**onentially by time.
- **ARF**: **A**daptive **R**andom **F**orest does both internal and external concept drift detecting for each newly-created tree. The final prediction will be obtained by its voting strategy.
- **Condor**: **Con**cept **D**rift via m**o**del **R**euse is an ensemble method that handles non-stationary environments by both building new models and assigning weights for previous models.

Method	Stock Price Trend Forecasting					Electricity Load		Solar Irradiance		
	IC	*ICIR*	*Ann.Ret.*	*Sharpe*	*MDD*	*NMAE*	*NRMSE*	*Skill* (%)	*MAE*	*RMSE*
RR	0.1178	1.0658	0.1749	1.5105	-0.2907	0.1877	0.9265	7.3047	21.7704	48.0117
GF-Lin	0.1227	<u>1.0804</u>	0.1739	1.4590	-0.2690	0.1843	0.9109	<u>9.3503</u>	21.6878	<u>46.9522</u>
GF-Exp	0.1234	1.0613	0.1854	1.5906	-0.2984	0.1839	0.9084	9.2652	21.6841	46.9963
ARF	0.1240	1.0657	0.1994	1.8844	**-0.1176**	<u>0.1733</u>	<u>0.8901</u>	8.6267	<u>21.0962</u>	47.3270
Condor	<u>0.1273</u>	1.0635	<u>0.2157</u>	<u>2.1105</u>	-0.1624	—	—	—	—	—
DDG-DA	**0.1312**	**1.1299**	**0.2565**	**2.4063**	<u>-0.1381</u>	**0.1622**	**0.8498**	**12.1327**	**18.7997**	**45.5110**

Table 1: Performance comparison of the concept drift adaptation methods.