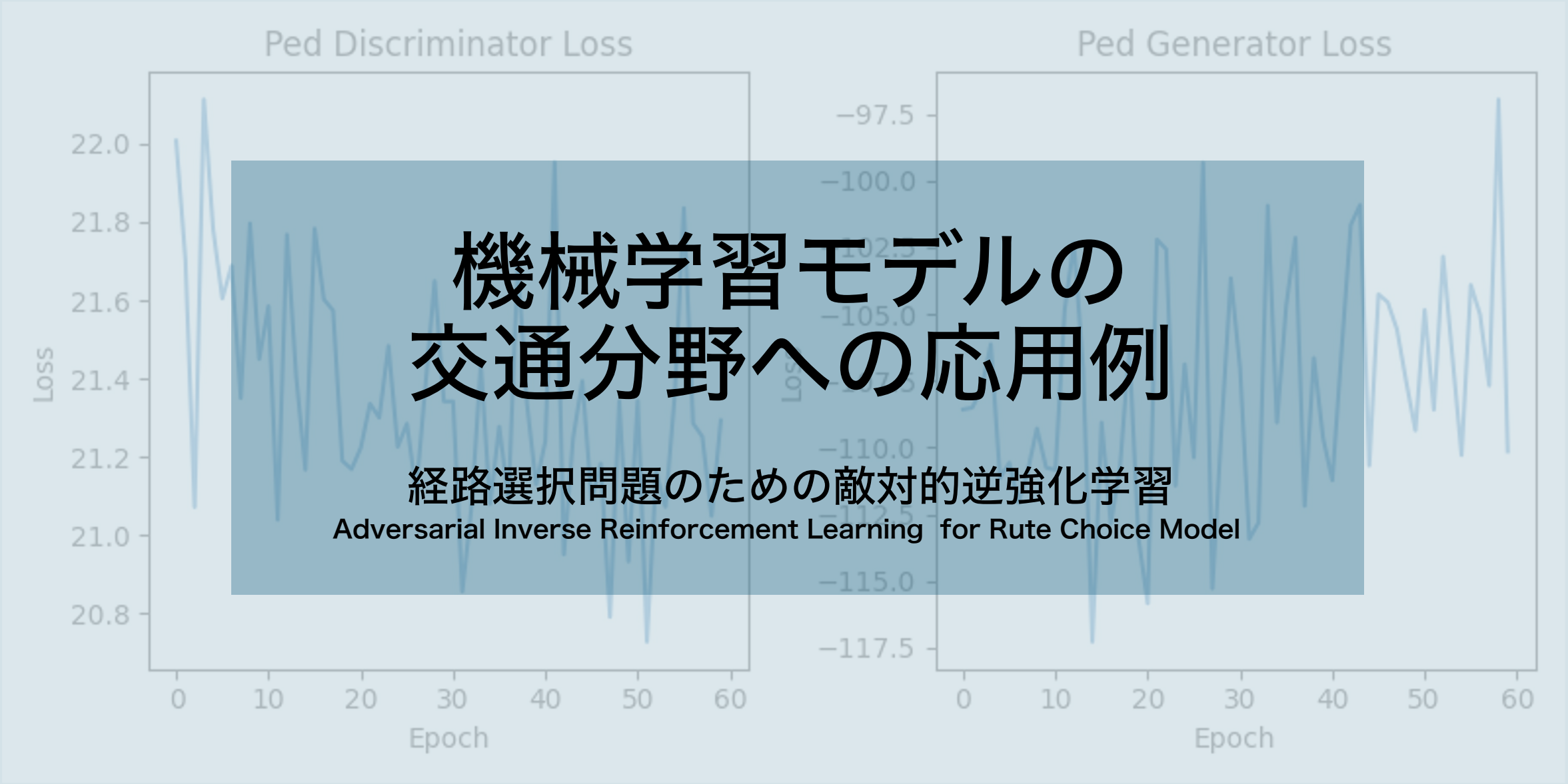


機械学習モデルの交通分野への応用例

経路選択問題のための敵対的逆強化学習

Adversarial Inverse Reinforcement Learning for Rute Choice Model

目次

- AIRL for RCMの概要
- 定式化
- 逆強化学習とは
- 逆強化学習としての定式化
- AIRLとは
- Generator
- Disctriminator
- 実装（機械学習の実装）
- AIRL の実装
- 実証実験の結果
- 機械学習の説明可能性(SHAP)

- **経路選択問題で機械学習をどう使うのか？**

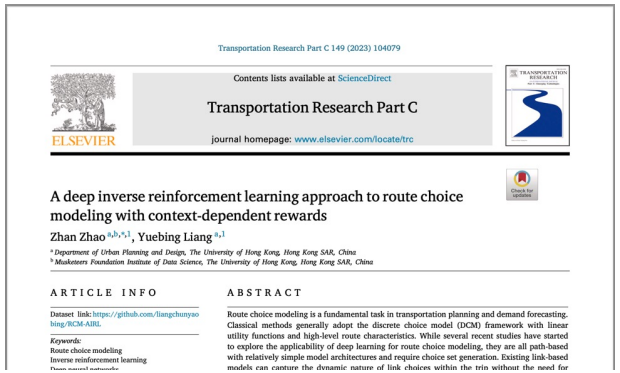
  How is the AIRL used in route choice problem?

- **データ処理から実装までの大体の流れ** The process of AIRL for RCM

AIRL for RCM

Adversarial Inverse Reinforcement Learning for Rute Choice Model

- **AIRLという逆強化学習アルゴリズムを経路選択問題に適用**した論文
- Zhan Zhao , Yuebing Liang,　2023　(←新しい！)


Transportation Research Part C 149 (2023) 104079

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A deep inverse reinforcement learning approach to route choice modeling with context-dependent rewards

Zhan Zhao [a,b,*,1], Yuebing Liang [a,1]

[a] Department of Urban Planning and Design, The University of Hong Kong, Hong Kong SAR, China
[b] Musketeers Foundation Institute of Data Science, The University of Hong Kong, Hong Kong SAR, China

ARTICLE INFO

Dataset link: https://github.com/liangchunyaobing/RCM-AIRL

Keywords:
Route choice modeling
Inverse reinforcement learning

ABSTRACT

Route choice modeling is a fundamental task in transportation planning and demand forecasting. Classical methods generally adopt the discrete choice model (DCM) framework with linear utility functions and high-level route characteristics. While several recent studies have started to explore the applicability of deep learning for route choice modeling, they are all path-based with relatively simple model architectures and require choice set generation. Existing link-based models can capture the dynamic nature of link choices within the trip without the need for

■従来モデルと比較したメリット

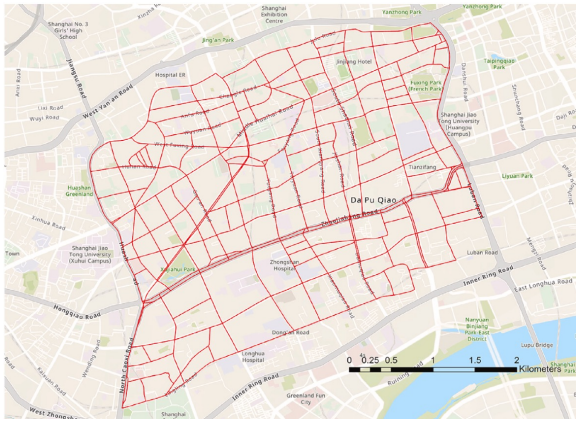
- **効用関数が線形に限定されず非線形も表せる**ため人間の複雑な嗜好を表現可能　**→精度が高い結果に** shows better result

  実証分析（中国の広範囲ネットワークで，RLモデルや他の機械学習モデルと比較しても精度が最も高いことが示された)

- 深層学習：**膨大なデータ量，　データ種類の増加に強い**

  →天気や個人属性などを考慮することが可能

中国のネットワーク図



経路選択問題の定式化

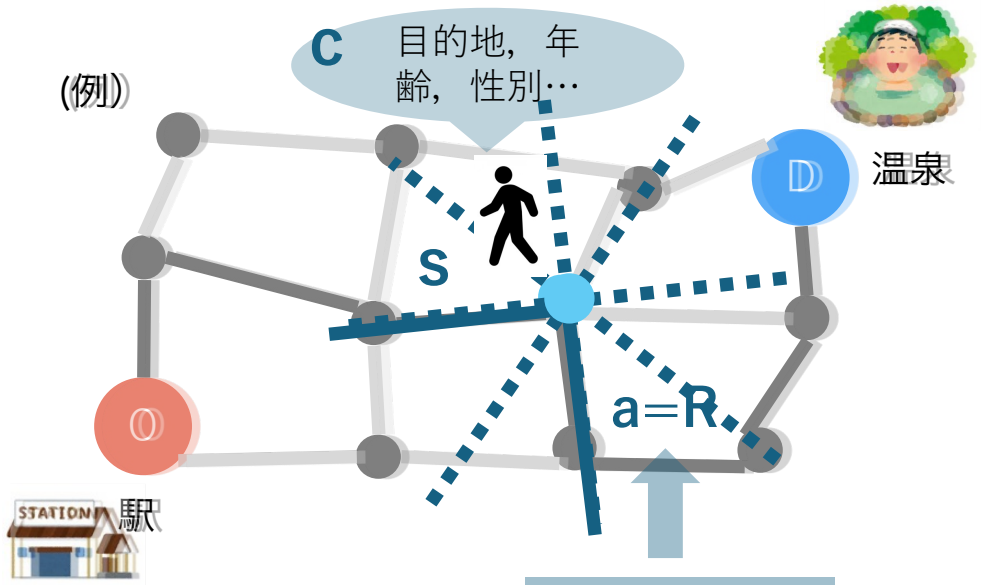
経路選択問題をマルコフ決定過程(MDP)として考える

←リンクベースの逐次的な経路選択という点ではRLの定式化と同じ!!

- **State(状態s)**: 道路ネットワークにおいて旅行者の**現在の位置**を示すリンク(current link)
- **Action(行動a)**：リンク間の移動選択（8方向のいずれか）R,Fなど
- **Context(コンテキストc)** :トリップ中変化せず，個人の経路選択に影響を及ぼす要素群ベクトル(consistent factor)

（目的地や目的、個人属性、日時、天気などex;distination）

- **Policy(ポリシーπ): 状態sとコンテキストcが与えられた時に行動aを決定する**関数(**π(a|s,c)**と表記）（π*は最適な選択行動を返す）
- **Reward function(報酬関数R): cが与えられた時、sの状態でaの行動を選択した時の効用（R(s,a|c)**と表記）



ポリシーπ

報酬関数
≒効用関数

ポリシーより報酬関数の方がより根本的で汎用性が高い
→報酬関数を求めたい（＝逆強化学習）

逆強化学習とは？ 補足

強化学習 Reinforcement Learning

予め定義された**報酬関数を最大化する行動を生み出す決定プロセス**(戦略，ポリシー)**を学習する**ことが目的

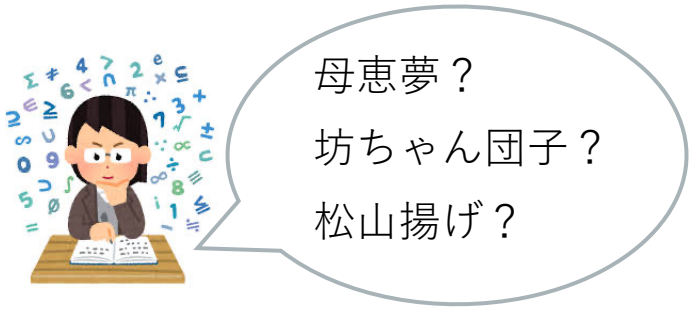
⇔

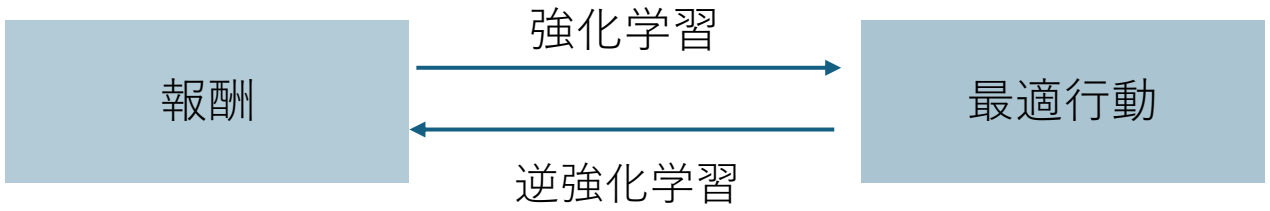
逆強化学習 Inverse Reinforcement Learning

データから，観察された人間の行動を説明する報酬関数を復元することが目的

報酬関数の方が汎用性が高い

(例)研究室にお土産買う時





逆強化学習として定式化

■ **目標：行動データXから，報酬関数Rを推定する** Goal ; To recover Reward function from observed behavior

- **行動データXの定義式:**　行動データXをs,a,cペアに分解

$$x_i = \{(s_1^{(i)}, a_1^{(i)}), (s_2^{(i)}, a_2^{(i)}), \ldots, (s_{T_i}^{(i)}, a_{T_i}^{(i)}) \mid c^{(i)}\},$$ ←エージェントiのOからDまでをリンク遷移(s,a)ペアに分割

↑i人目の軌跡　↑tステップ目

- **報酬関数Rの定義式**　パラメータθ関数として定義し、**θを学習させることで、Rの関数を形にする**

RLの効用関数と同じ！

$$R_\theta(x_i) = \sum_{t=1}^{T_i} \gamma^t R_\theta(s_t^{(i)}, a_t^{(i)} \mid c^{(i)}).$$ **エージェントiの報酬関数** OからDまでの各移動ステップの報酬関数を足す

全エージェントの行動データから確率の立式
（観測されたx(軌跡)の起こる確率はRのexpに比例するという法則）

$$P_\theta(x) = \frac{1}{Z}\exp\left(R_\theta(x)\right)$$ **行動xの起こる確率**　Zは全てのx（軌跡）のRθ(x)の**積分和**

全エージェントは最大の確率を選択して行動するものとする

$$\max_\theta \sum_{i=1}^{N} \log\left(P_\theta(x_i)\right)$$:→IRL問題は観測された軌跡に基づいて**尤度を最大化する問題**として組み立てられる

maximum likelihood problem

逆強化学習として定式化

- 目標：行動データから，報酬関数Rを推定する
  - 行動データXの定義式:　Xにはs,a,cの情報が含まれる

$$x_i = \{(s_1^{(i)}, a_1^{(i)}), (s_2^{(i)}, a_2^{(i)}), \ldots, (s_{T_i}^{(i)}, a_{T_i}^{(i)}) \mid c^{(i)}\},$$

←エージェントiのOからDまでをリンク遷移(s,a)ペアに分割

↑i人目の軌跡　↑tステップ目

  - 報酬関数Rの定義式　パラメータθ関数として定義し、θを学習させることで、Rの関数を形にする

$$R_\theta(x_i) = \sum_{t=1}^{T_i} \gamma^t R_\theta(s_t^{(i)}, a_t^{(i)} \mid c^{(i)}).$$

エージェントiの報酬関数

全エージェントの行動データから確率の立式
（観測されたx(軌跡)の起こる確率はRのexpに比例するという法則）

$$P_\theta(x) = \frac{1}{Z}\exp\big(R_\theta(x)\big)$$

行動xの起こる確率　Zは全てのx（軌跡）のRθ(x)の積分和

全エージェントは最大の確率を選択して行動するものとする

- **課題:Zを計算すること**

Challenge: computation of Z

従来はRLモデルのように線形と決めることで、パラメータ推定
→本研究ではZを直接求めず別のアプローチをとる(**AIRLを利用**)

目次　小結①

- AIRL for RCMの概要
- 定式化
- 逆強化学習とは
- 逆強化学習としての定式化
- AIRLとは
- Generator
- Disctriminator
- 実装（機械学習の実装）
- AIRL の実装
- 実証実験の結果
- 機械学習の説明可能性(SHAP)

まとめ

- **経路選択問題は逆強化学習の問題として定式化できる**

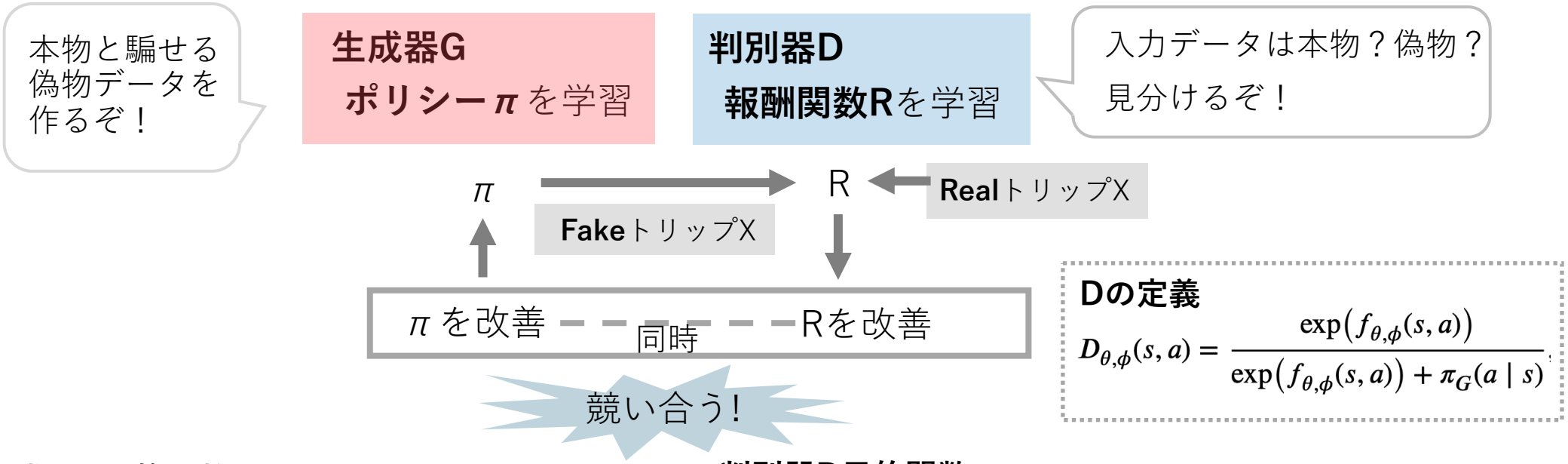
  Formulation as IRL

- **逆強化学習を解くのにAIRLアルゴリズムを用いる**

  use AIRL to solve IRL problem

AIRLって？ Adversarial IRL：敵対的逆強化学習

- **逆強化学習のアルゴリズム**の一種
- 生成ネットワークを援用し，**方策関数を決定する生成器G**と**報酬関数を決定する判別器D**という２つのモデルが**互いに競い合うアルゴリズムで同時に学習**　Competition improves both models,discriminator and generator



Dの定義

$$D_{\theta,\phi}(s,a) = \frac{\exp\left(f_{\theta,\phi}(s,a)\right)}{\exp\left(f_{\theta,\phi}(s,a)\right) + \pi_G(a \mid s)}$$

生成器G目的関数

$$\max_{\pi_G} \left[E_{\pi_G}[\log(D_{\theta,\phi})] - \log(1 - D_{\theta,\phi})\right]$$

判別器D目的関数

$$\min_{\theta,\phi} -E_D\left[\log\left(D_{\theta,\phi}(s,a)\right)\right] - E_{\pi_G}\left[\log\left(1 - D_{\theta,\phi}(s,a)\right)\right]$$

実データ　偽データ

Generator

Gの目標：OD（出発地と目的地）のペアから経路選択を生成する**ポリシーπGを学習**すること

Goal; To learn policy

- **Gの入力** input

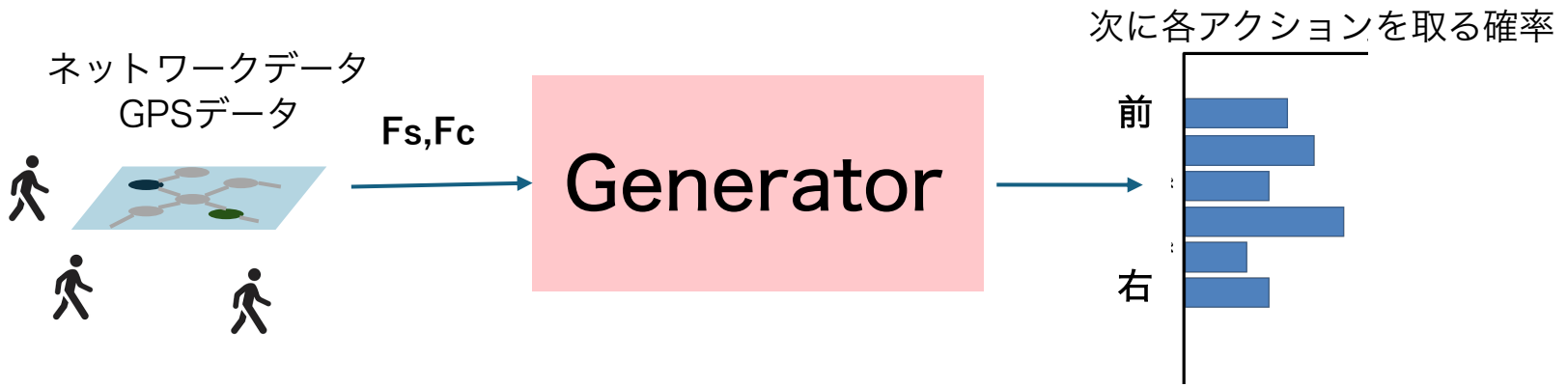
①**状態特徴量Fs**：現在のリンクの特徴（例：リンクの長さ）

②**コンテキスト特徴量Fc**：目的地に関する特徴（例：目的地までの距離）、その他（例：旅行者属性）

- **Gの出力** output

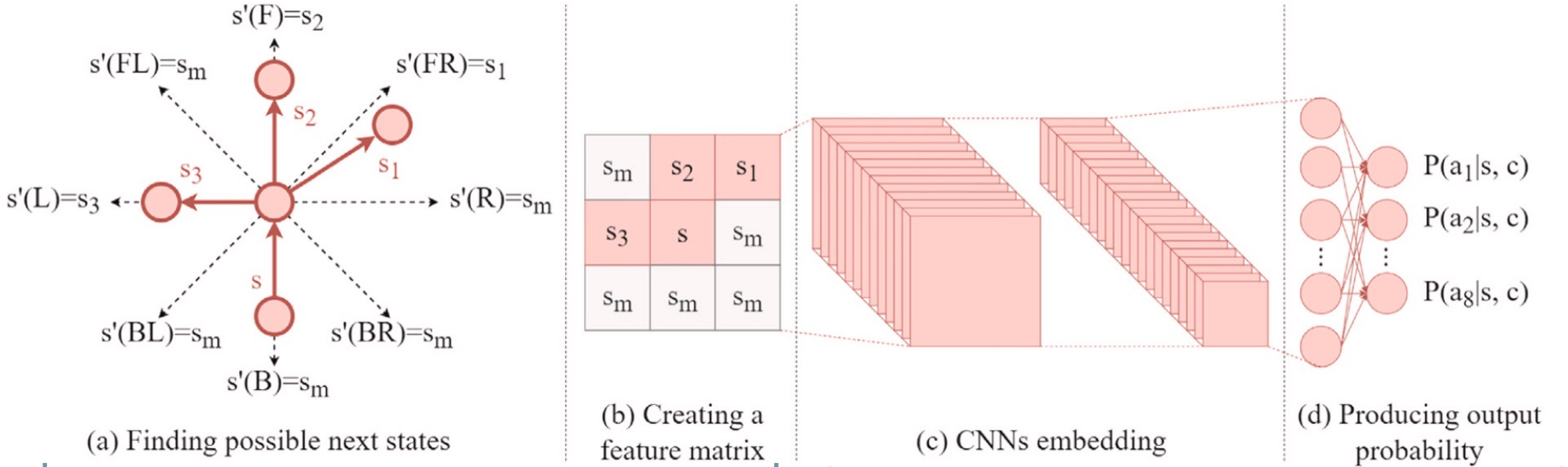
現在の状態sとコンテキストcが与えられた場合に旅行者が**各アクションを選択する確率分布**

- 隣接する状態間の空間的相関を考慮するために、畳み込みニューラルネットワーク（CNN←DNNの一種）を使用して、現在の状態と潜在的な次の状態のために$F = [Fs ; Fc]$を集約する



Generator

使用しているネットワーク構造（データ処理の流れ） Generator network



データ処理(Fs,Fcの整形)

NNでの学習

Step1

可能な次の状態を見つける。状態sが与えられた場合、各アクション（すなわち方向）$a \in A$に対して次の状態$s'(s,a)$を見つける。アクションが有効な状態につながらない場合や、$a \notin A(s)$の場合、$s'(s,a)$はマスク状態としてsmで示される。

Step2

特徴行列の作成。3×3の特徴行列に、状態特徴量Fsと可能な次の状態$S'(s)$のコンテキスト特徴量Fcを集約

Step3

畳み込みニューラルネットワーク（CNN）の埋め込みを学習する。カーネルサイズ2の2層のCNNを使用して、特徴行列から潜在空間ベクトルを学習

Step４

アクションの確率を生成する。CNNから学習した潜在空間ベクトルを入力として、2層の順伝播ニューラルネットワークとsoftmax関数を使用して出力を生成する。

Discriminator

Dの目標：実際の人間の軌跡とGeneratorが作った軌跡を見分けること

Goal; To discriminate real behavior from fake behavior generated from G.

- Dの入力

①**状態特徴量Fs**：現在のリンクの特徴（例：リンクの長さ）

②**コンテキスト特徴量Fc**：目的地に関する特徴（例：目的地までの距離）、その他（例：旅行者属性）

③**行動ベクトルF_a**：

- Dの出力

本物であると判断する確率D

$$D_{\theta,\phi}(s,a|c)=\frac{\exp\left(f_{\theta,\phi}(s,a|c)\right)}{\exp\left(f_{\theta,\phi}(s,a|c)\right)+\pi_G(a\mid s,c)}$$

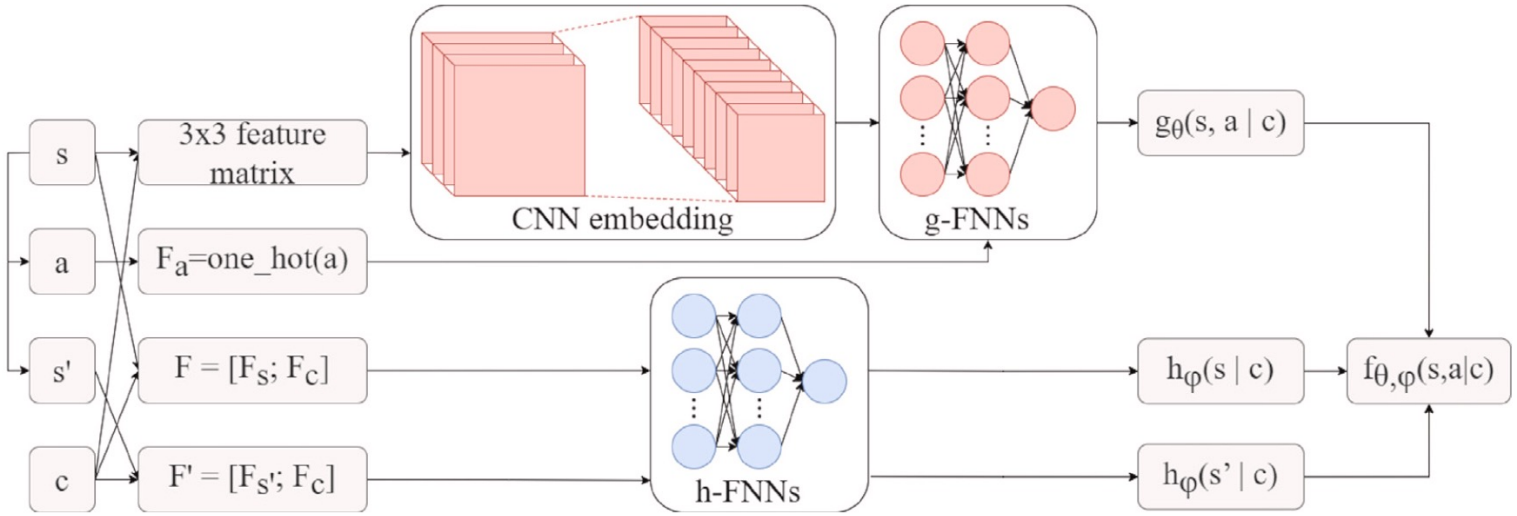
$$f_{\theta,\phi}(s,a|c)=g_\theta(s,a|c)+\gamma h_\phi(s'|c)-h_\phi(s|c).$$

$g_\theta(s,a|c)$ Reward approximator

$h_\phi(s|c)$. 不必要な更新を防ぐ役割

Discriminator

モデルイメージ



- fを知るにはg(Rの近似関数)とhが必要
- Rは次の状態に依存するので今の状態と、隣接する状態の特徴を畳み込んで入力するためCNNネットワークを使う(CNNにFs,Fcがいる)（Gの時と同じ感じ）
- Rにはactionも影響を及ぼす（人はまっすぐが好きとか、右とか左とかが好きとか）
- Actionの特徴はone-hotベクトルとして表し、CNNに$F = [Fs;Fc]$ がインプットされたレイヤーと繋ぐ
- gはこのCNNレイヤーによって学習される
- hもCNNレイヤーを使って学習できるが計算の都合上、g,hからfを出し、Dを求めたら、Rを以下の式で求める

$$R_{\theta,\phi}(s,a) = \log\big(D_{\theta,\phi}(s,a)\big) - \log\big(1 - D_{\theta,\phi}(s,a)\big) = f_{\theta,\phi}(s,a) - \log \pi_G(a \mid s)$$

機械学習の実装 復習

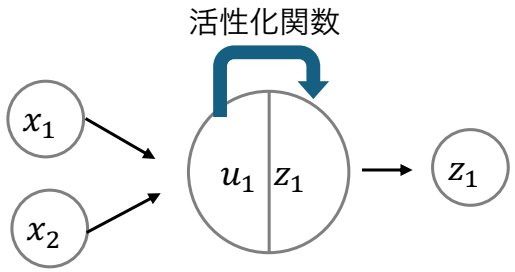
- **ニューラルネットワーク**：深層学習の基本的な仕組み　Neural Networks

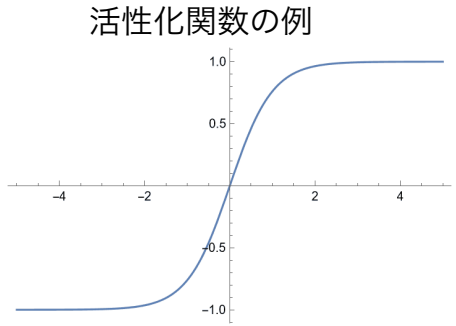
ニューラル素子：$\sigma = f(\omega \times x + \beta)$

入力　バイアス

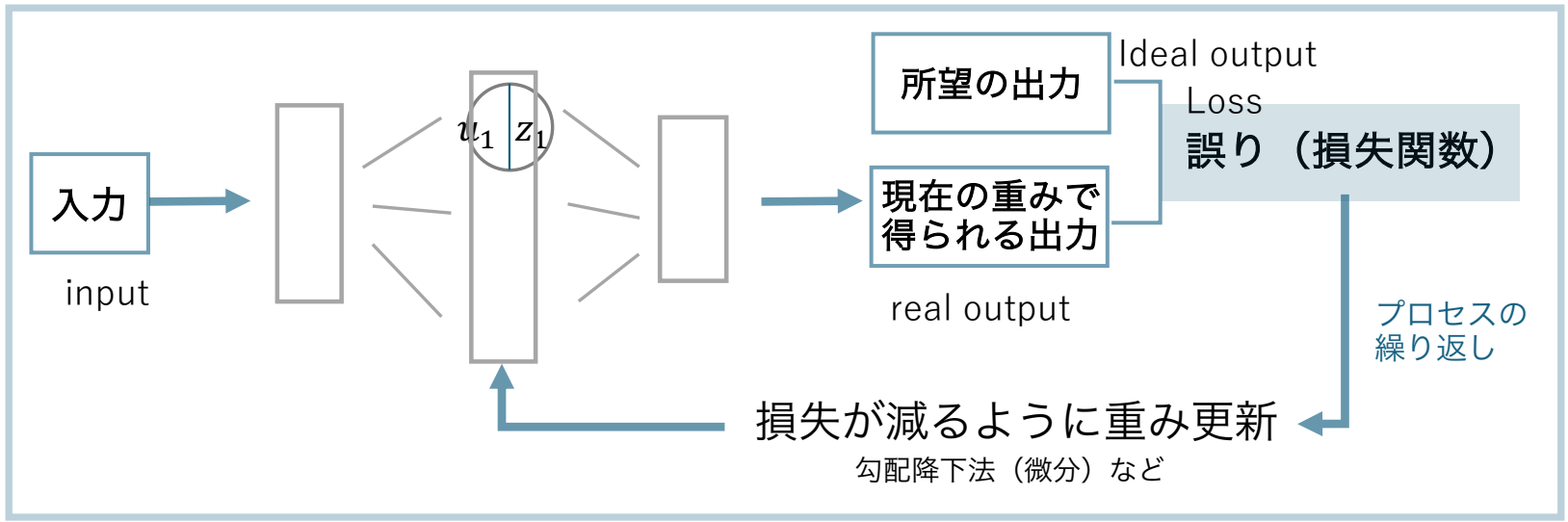
重みパラメータ

活性化関数





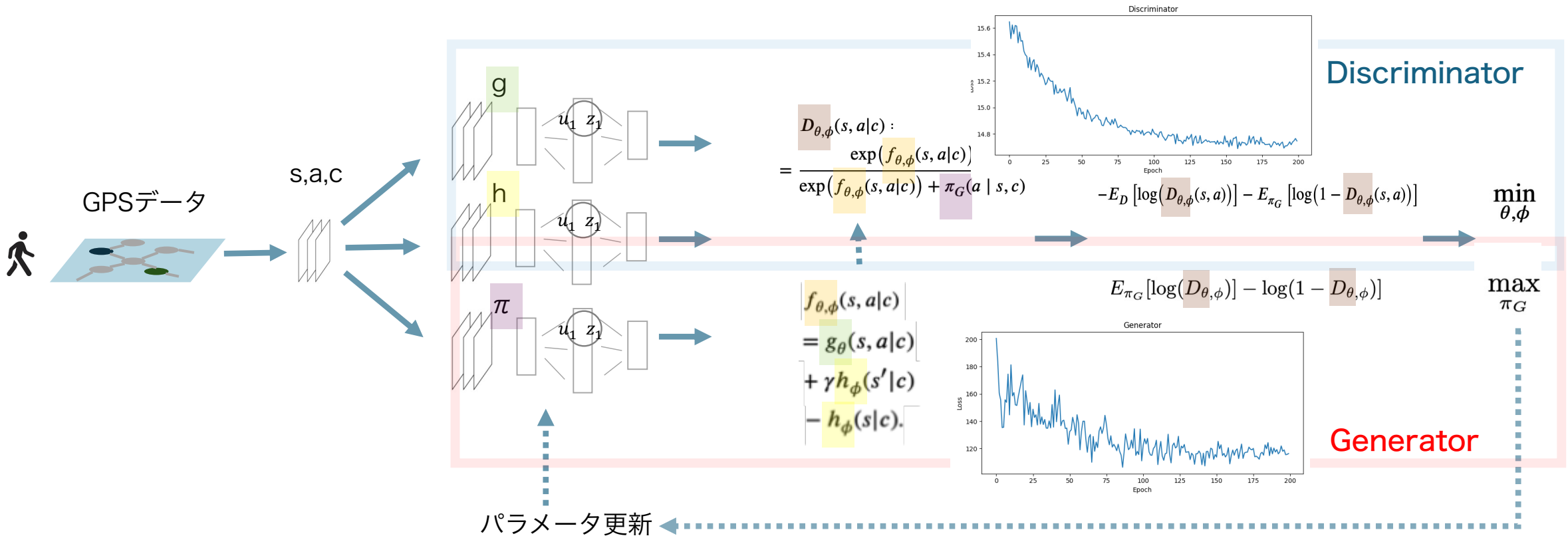
- **モデル構築の基本（パラメータの学習プロセス）**



参考：岡谷貴之著　深層学習 改訂第2版 (機械学習プロフェッショナルシリーズ)

AIRL実装の流れ

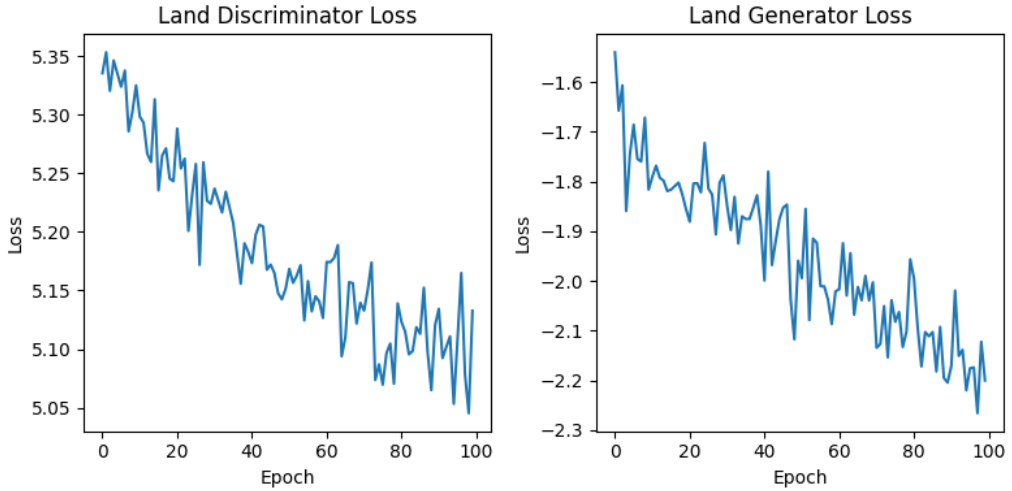
データ入力 → データ処理 → ニューラルネットワーク → 損失関数部品 → 損失関数 → 最適化



$R_{\theta,\phi}(s,a \mid c) = \log\left(D_{\theta,\phi}(s,a \mid c)\right) - \log\left(1 - D_{\theta,\phi}(s,a \mid c)\right)$：目標の報酬関数の復元達成

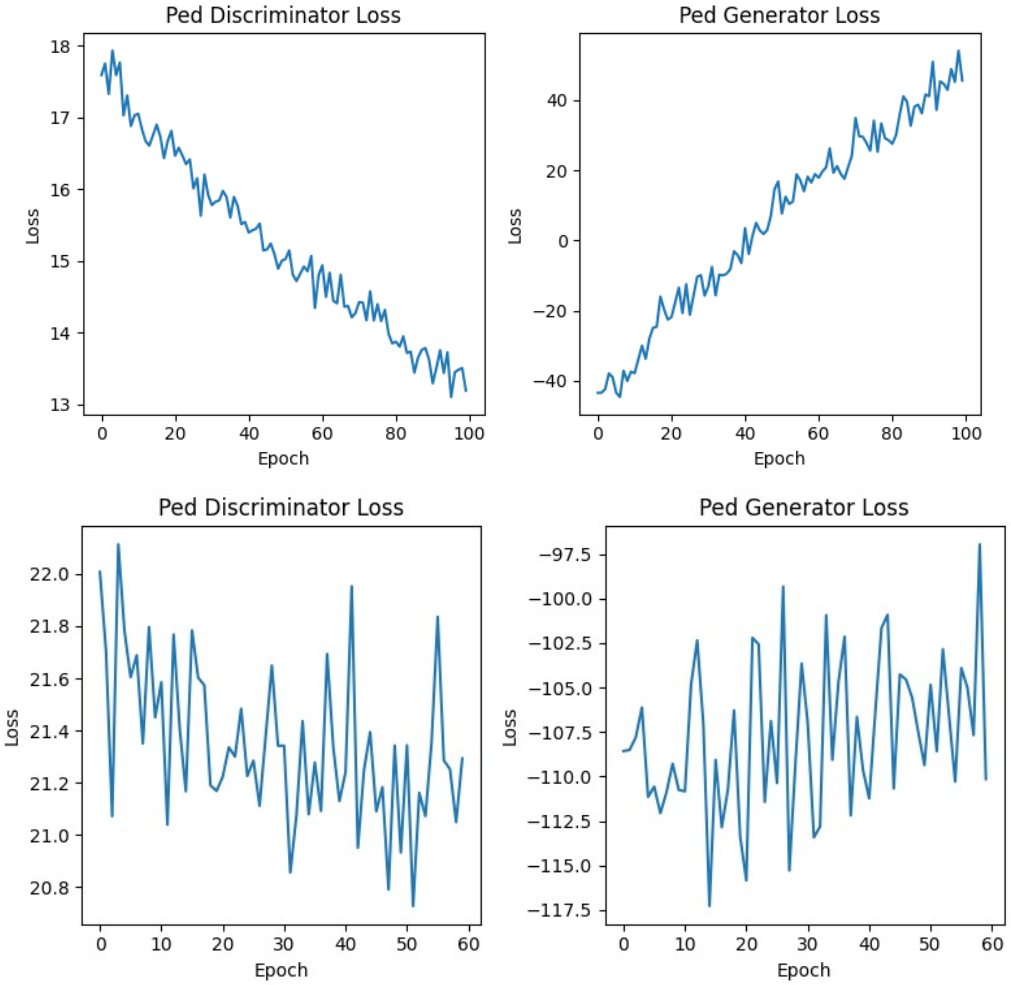
AIRL学習の可視化

- 成功例



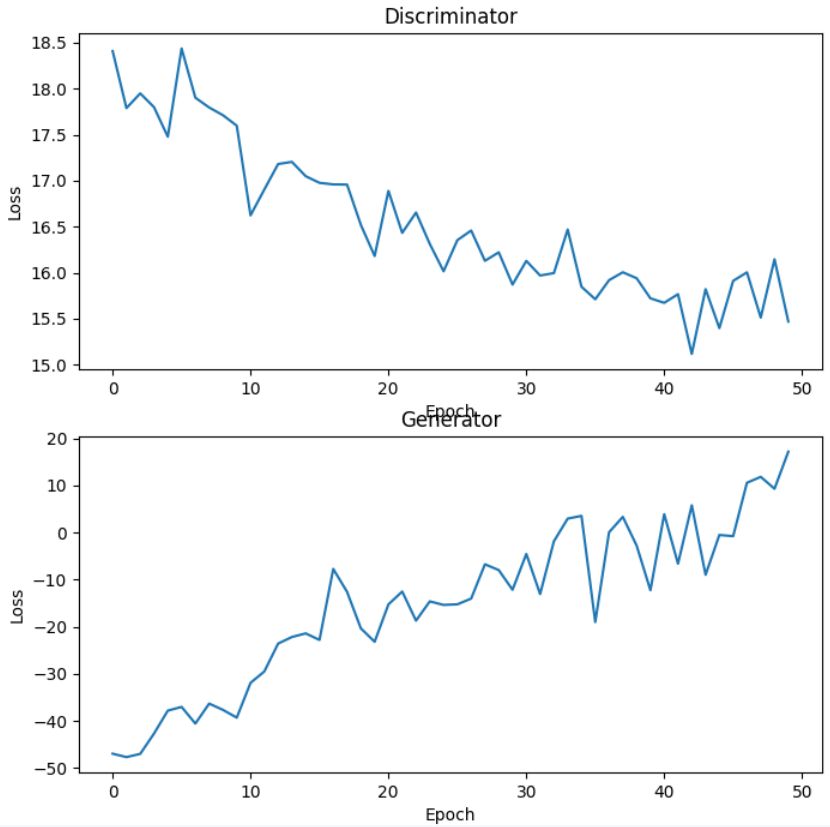
初期値，学習率の変更で調整

- 失敗例

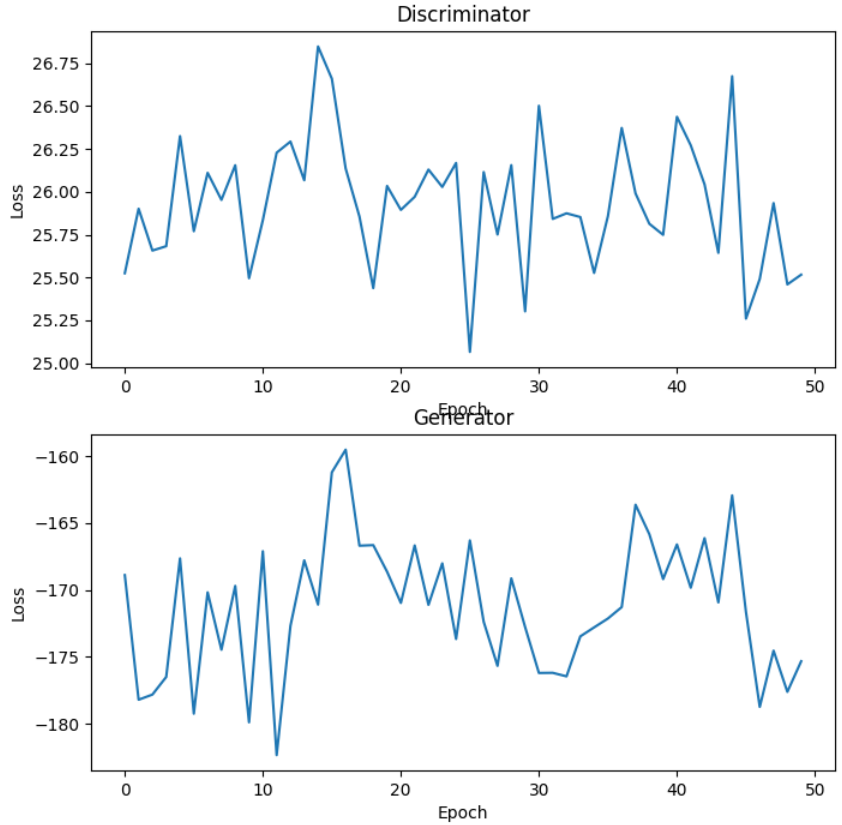


AIRL学習の可視化

学習率変更



学習率0.1倍に




```
self.optimizer_util = optim.Adam(self.util.parameters(), lr=0.0000001)#lr=0.000003
self.optimizer_value = optim.Adam(self.value.parameters(), lr=0.000001)#lr=0.00005
self.optimizer_policy = optim.Adam(self.policy.parameters(), lr=0.000001)#lr=0.00005
self.optimizer_ext = optim.Adam(self.ext.parameters(), lr=0.000001)#lr=0.00005
```

目次　小結②

- AIRL for RCMの概要
- 定式化
- 逆強化学習とは
- 逆強化学習としての定式化
- AIRLとは
- Generator
- Disctriminator
- 実装（機械学習の実装）
- AIRL の実装
- 実証実験の結果
- 機械学習の説明可能性(SHAP)

まとめ②

- **AIRLはGeneratorとDiscriminatorの２つのモデルが競い合いながら精度を高める**
- **DやGを構成する3つの関数がニューラルネットワークで学習され（機械学習），報酬関数が求まる**

実証実験　データ

実証実験で用いたデータ

- 上海の大手タクシー会社の１つから提供されたデータセット
- 10609台のタクシーのGPS追跡,　**24470のトリップデータ**

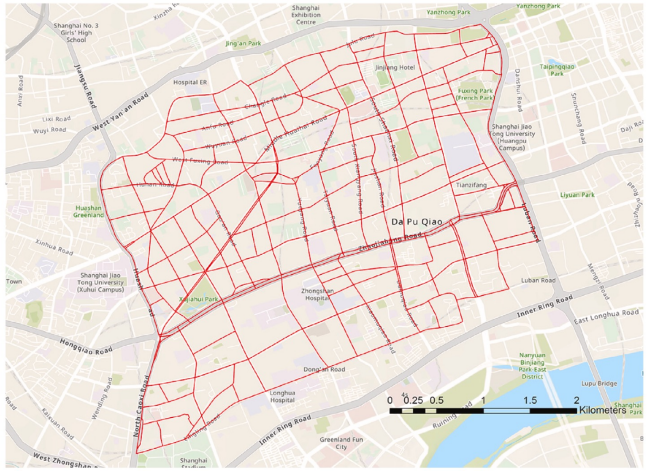


Fig. 5. Selected road network in Shanghai.

RCM-AIRLモデル比較する他のモデル

- Path Size Logit(PSL)：パスベースモデルの一つ。
- DNN-PSL：DNNを使用するPSLの拡張版。非線形な関係や柔軟なコンテキストを含めることができるのが特徴
- Recursive Logit：リンクベースモデル。価値関数は線形関数。パスの量は膨大だが、リンクベースなので全てのパスを抽出する必要がない

従来モデル

- RCM-BCモデル
- RCM-GAILモデル

AIRL類似モデル

実証実験　評価指標 Evaluation

モデルの精度の良さを測る指標 (実際のデータとモデルが予測するデータを比較する)

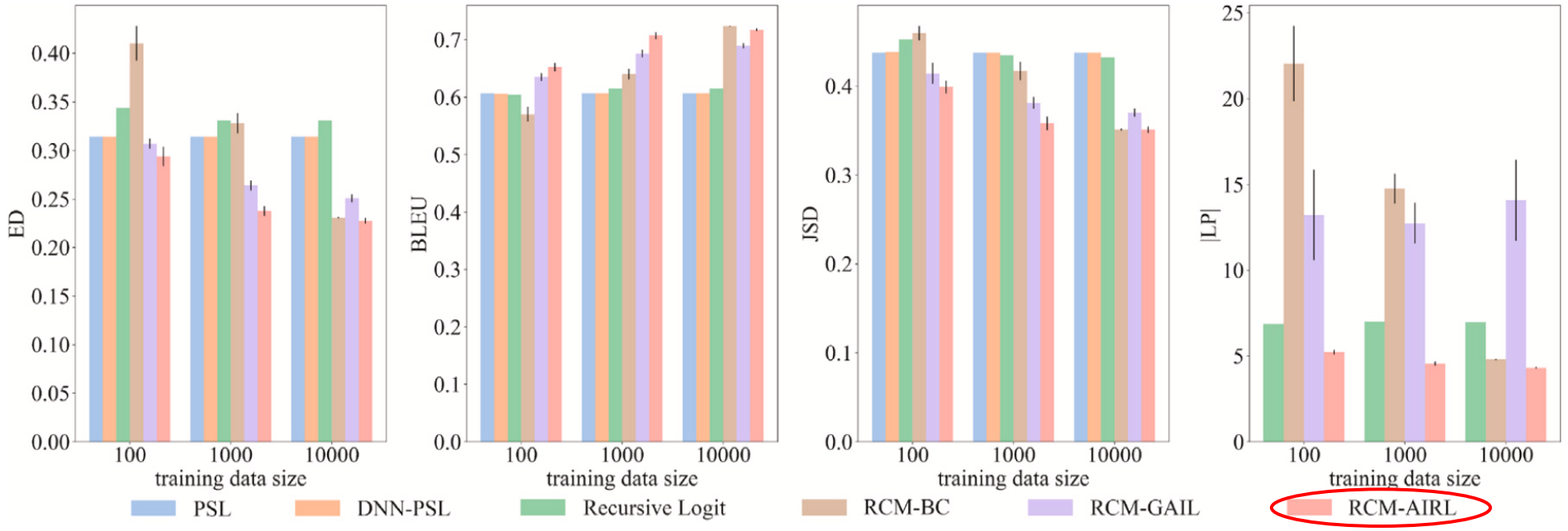
- Edit Distance(ED): 2 つの関連するものがどれくらい**異なっているか**を測る指標。
  一方を他方に変換する最小の操作回数の計算によって求める。
- BiLingual Evaluation Understudy score (BLEU):スコアは 2 つがどの程度似ているか、**近いか**を示す
- Jensen–Shannon Distance (JSD):2つのものの確率分布の**近さ**を測ることができる
- Log Probability (LP):与えられたモデルにおいて実データの経路の確率のlogをとったものの平均値

データ成形

- データをランダムに 5 つに分け、 1 つをテスト用、残り 4 つトレーニング用
- 学習用のデータサイズを100,1000,10000の 3 種類で実験
- 実験成功のために重要だとわかったこと：コンテクストに依存する報酬関数の設計

←ゴールに関連する情報をきちんと与えることで旅行者が合理的な行動として経路選択すると意味付けられるので、何度もループしたりゴールに到達しないという課題を解消する

結果　モデル比較



- RCM-BC、RCM-GAIL、RCM-AIRLは、既存のベースラインモデルよりも優れたパフォーマンスを示した
→**深層IRL（逆強化学習）/IL（模倣学習）手法がルート選択モデリングにおいて効果的**であることを示唆
(特にAIRLは全ての評価指標に関して良い結果)
- モデルの潜在的な不果実性を考慮するため、３回実験を行なったものの標準偏差をエラーバーとして示したところ、AIRLは割と安定

結果　交通流のシミュレーション結果

R^2：回帰モデルの予測の適合度を評価する統計的指標

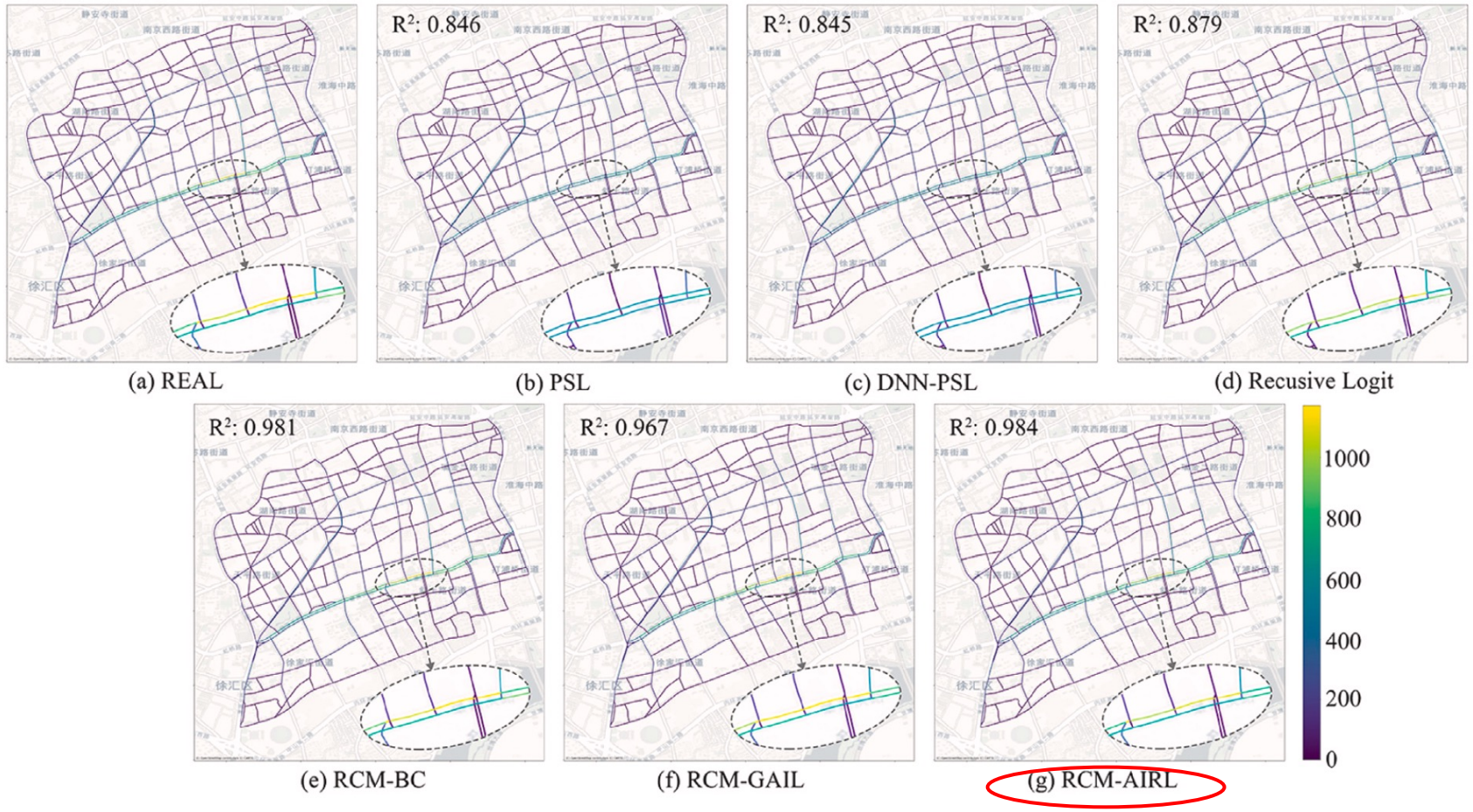


Fig. 7. Predicted link flow distribution using different models.

結果　あるODペアに対する実際のルート選択と予測

結果　計算時間

- FE: 特徴量エンジニアリングに必要な時間
- MO: モデルの最適化に必要な時間
- PG: 経路生成に必要な時間

モデルの訓練（学習）

モデルのテスト

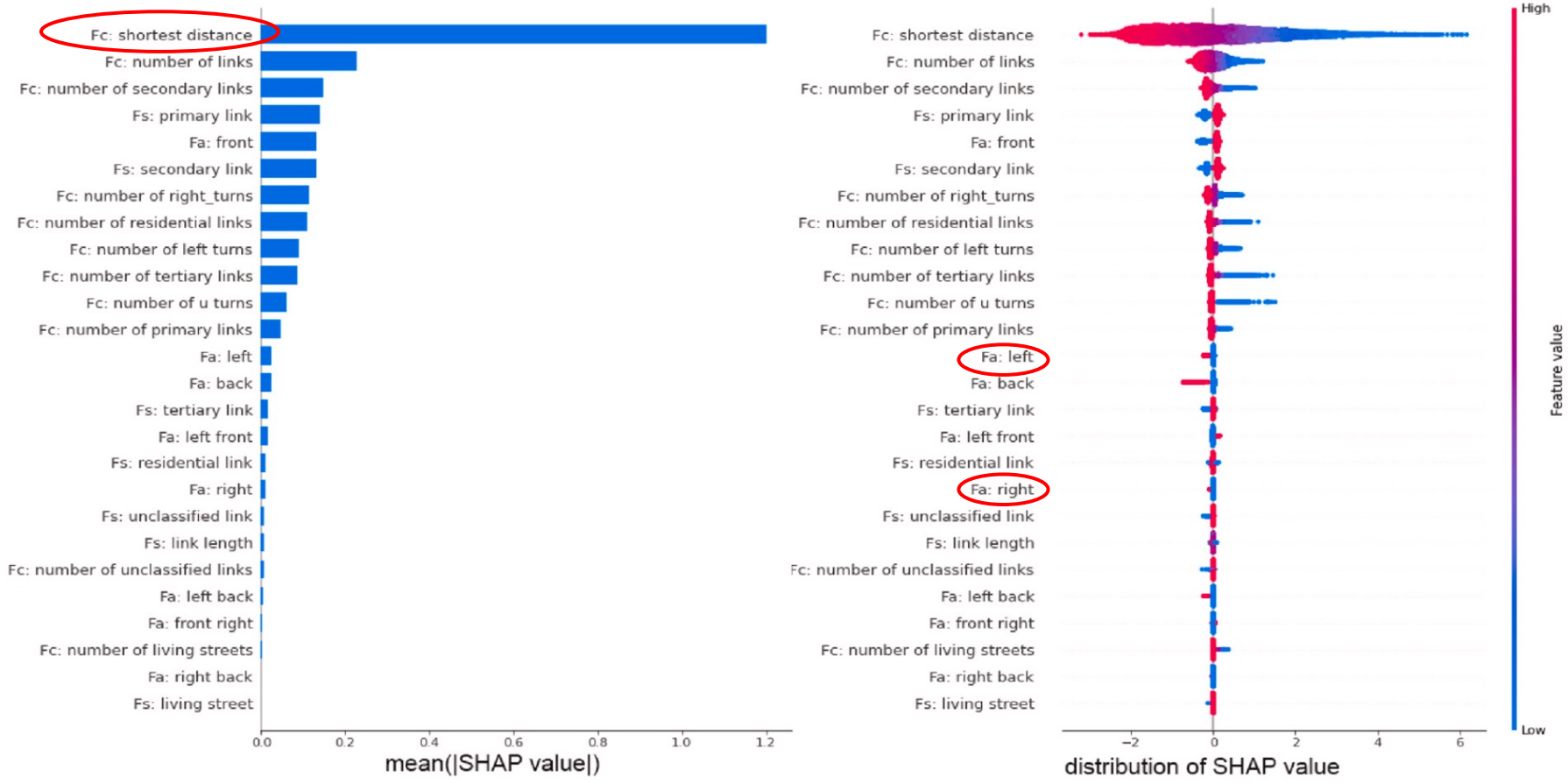
Table 3
Computation time for feature engineering (FE), model optimization (MO), and path generation (PG) of different models. The results are averaged over 5-fold cross validation.

Models		100 training trips			1000 training trips			10 000 training trips		
		FE(s)	MO(s)	PG(s)	FE(s)	MO(s)	PG(s)	FE(s)	MO(s)	PG(s)
Path-based	PSL	28.8	0.3	1370.5	277.8	1.8	1370.5	3166.7	12.6	1370.4
	DNN-PSL	28.8	0.8	1370.4	277.8	5.7	1370.4	3166.7	42.3	1370.4
Link-based	Recursive logit	0.0	22.2	4.0	0.0	44.4	4.0	0.0	238.8	4.1
	RCM-BC	3840.0	124.8	1.7	3840.0	1174.1	1.6	3840.0	11 487.4	1.6
	RCM-GAIL	3840.0	6112.8	1.7	3840.0	9305.3	1.6	3840.0	18 270.9	1.6
	RCM-AIRL	3840.0	5990.4	1.6	3840.0	8962.5	1.5	3840.0	18 307.0	1.5

モデル訓練は１度でよく，交通流の予測やシミュレーションにはモデルのテストにかかる時間が重要

SHAP　経路選択行動の”解釈可能性”について

- SHapley Additive exPlanations（SHAP）を使用して、入力特徴量の影響を理解
- SHAPは、機械学習モデルの説明を行うためのゲーム理論的なアプローチを用いる説明可能なAI技術



参考 定式化

3.2. Route choice as an inverse reinforcement learning problem

Recall that IRL for route choice modeling seeks to infer the reward function $R(s, a \mid c)$ given a set of observed trajectories $X = \{x_1, x_2, \ldots, x_N\}$ that are assumed to be drawn from an optimal policy $\pi^*(a \mid s, c)$. Note that each observed trajectory is a sequence of station–action pairs with a certain context. Let us denote the ith trajectory as $x_i = \{(s_1^{(i)}, a_1^{(i)}), (s_2^{(i)}, a_2^{(i)}), \ldots, (s_{T_i}^{(i)}, a_{T_i}^{(i)}) \mid c^{(i)}\}$, where $(s_t^{(i)}, a_t^{(i)})$ is the tth state–action pair of x_i, $c^{(i)}$ the context, and T_i the length of the trajectory.

In large road networks, there are many possible combinations of (s, a, c), making it difficult to obtain a robust estimate of $R(s, a \mid c)$. A common solution is to approximate it with a parameterized reward function $R_\theta(s, a \mid c)$, where θ is the function parameters to be learned. The specific function approximation can take many different forms, among which DNNs are generally more flexible and adept in dealing with high-dimensional features. As the total reward of a whole trajectory x_i is simply the sum of the discounted rewards for each state–action pair in the trajectory, the parameterized reward function can be expressed as

$$R_\theta(x_i) = \sum_{t=1}^{T_i} \gamma^t R_\theta(s_t^{(i)}, a_t^{(i)} \mid c^{(i)}). \tag{1}$$

While the underlying policy that generates the observed trajectories is assumed to be "optimal", actual routing behavior may be suboptimal and vary across individuals. Even for the same OD pair, different travelers may choose different routes. To address this issue, the principle of maximum entropy is adopted to handle behavioral suboptimality as well as stochasticity by operating on the distribution over possible trajectories. Following the MaxEntIRL formulation (Ziebart et al., 2008), the probability of observing any given trajectory x is proportional to the exponential of its cumulative reward:

$$P_\theta(x) = \frac{1}{Z} \exp\big(R_\theta(x)\big), \tag{2}$$

where the partition function Z is the integral of $R_\theta(x)$ over all possible trajectories. Therefore, we can frame the IRL problem as solving the maximum likelihood problem based on the observed trajectories:

$$\max_\theta \sum_{i=1}^{N} \log\big(P_\theta(x_i)\big). \tag{3}$$

参考

本

- 岡谷貴之著　深層学習 改訂第2版 (機械学習プロフェッショナルシリーズ)
- 松尾豊著　人工知能は人間を超えるか
- 斎藤 康毅　ゼロから作るDeep Learning ―Pythonで学ぶディープラーニングの理論と実装

論文

- A deep inverse reinforcement learning approach to route choice modeling with context-dependent rewards, Zhan Zhao, Yuebing Liang, 2023

https://www.sciencedirect.com/science/article/abs/pii/S0968090X23000682

- Learning Robust Rewards with Adversarial Inverse Reinforcement Learning, Justin Fu, Katie Luo, Sergey Levine,2017

https://arxiv.org/abs/1710.11248