正規表現に基づく乗算型重み更新アルゴリズムの効率化
改め

オンライン予測の理論と応用

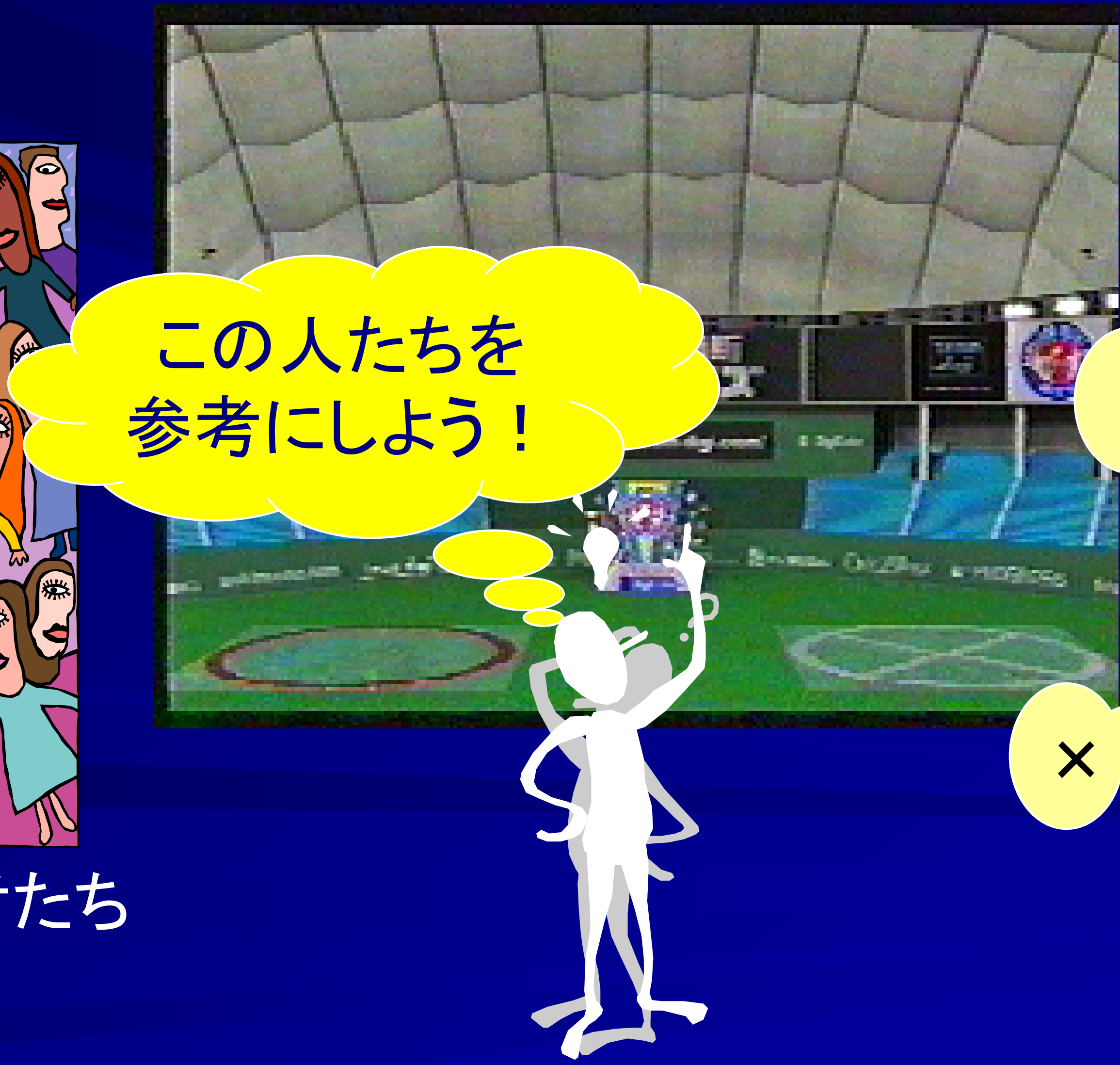
東北大学大学院情報科学研究科

A01班　瀧本 英二

クイズ王に匹敵する問題

「タイガーウッズは寅年生まれである．〇か×か？」

迷う参加者たち



クイズ王達
発見!!

正解は

×

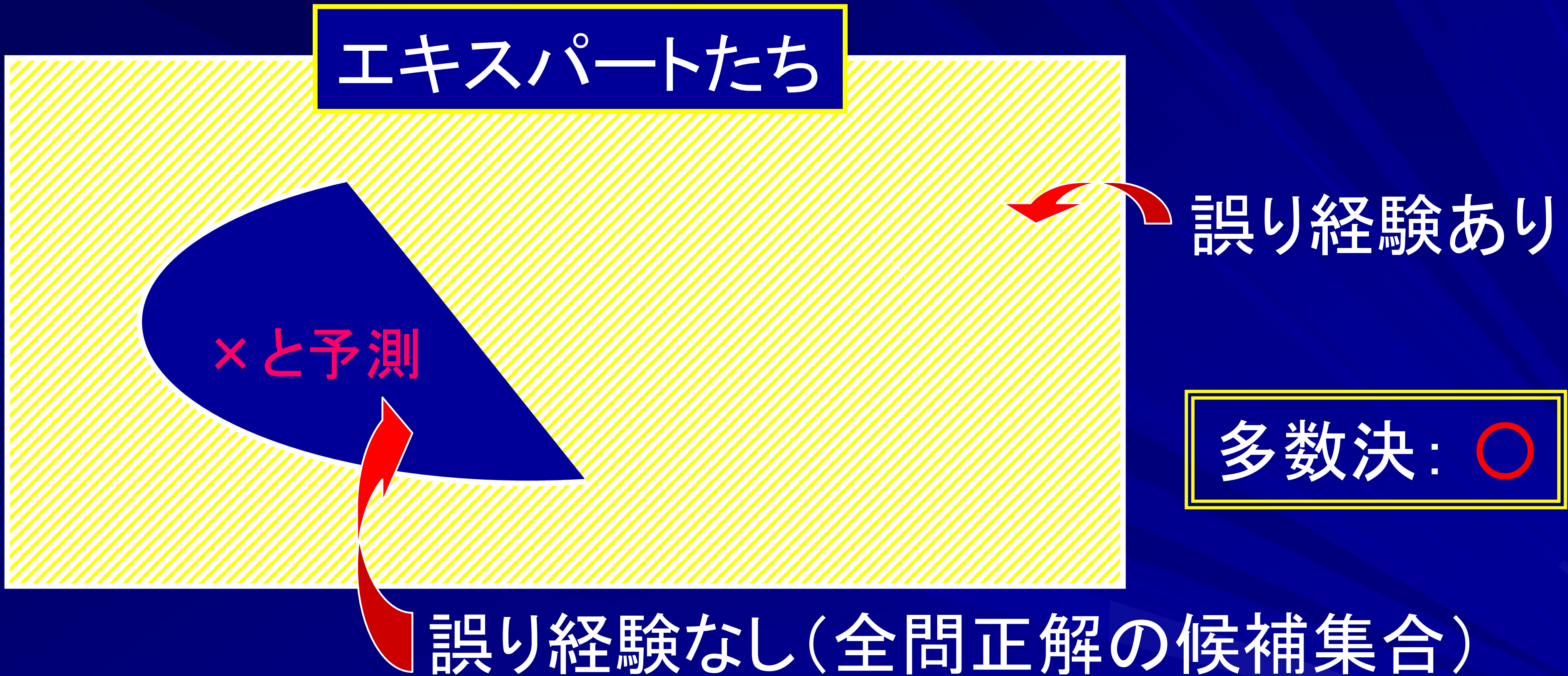
タイガーウッズはウサギ年生まれ

二分法(Halving Algorithm)

- クイズ王(エキスパート)N人のうち少なくとも1人は全問正解すると仮定
- 二分法:
「これまで一度も間違えていないエキスパートたちの予測の多数決に従う」

誤り回数は高々$\lfloor \log_2 N \rfloor$

二分法の解析



- 間違えたとき，候補は半分以下
- 誤り回数が m ⇒ 候補数 $\le N / 2^m$
- 仮定より， $1 \le$ 候補数

$$m \le \lfloor \log_2 N \rfloor$$

二分法 $\in$ 乗算型重み更新

- $v_i \in [0,1]$: エキスパート i の重み
- 初期重み $v_i = 1/N$
- $x_i \in \{-1,1\}$: エキスパート i の予測値
- 二分法の出力 : 重みつき多数決

  $\mathrm{sgn}(\sum_{i=1}^{N} v_i x_i \geq 0)$
- 重み更新

  $v_i \leftarrow v_i b_i$ / 規格化

ただし,

$b_i = 1$ ・・・ エキスパート i が正しかったとき

$\quad\;\; 0$ ・・・ エキスパート i が間違えたとき

重みつき平均アルゴリズム
―― 全問正解者がいない場合 ――

- 重み更新

  $v_i \leftarrow v_i\, b_i$ / 規格化

  ただし,

  $b_i = 1$　・・・ エキスパート i が正しかったとき

  　　$e^{-\eta}$ 0 ・・・ エキスパート i が間違えたとき

$$\text{誤り回数} \le m^* + O\left(\sqrt{m^* \ln N} + \ln N\right)$$

m^* は最も成績の良かったエキスパートの誤り回数

パーセプトロン $\in$ 加算型重み更新

- 学習目標 $v^* = (0, ..., 1, ..., 0)$ の線形分離関数（シングルトン）の学習
- パーセプトロンの出力：重みつき多数決

  $$y' = \mathrm{sgn}(\sum_{i=1}^{N} v_i x_i \geq 0)$$

- 重み更新

  $$v_i \leftarrow v_i + \eta\,(y-y')\,x_i$$

  ただし，y は真の値，η は学習定数
- 誤り回数 = $\Omega(N)$　cf. 二分法は $\lfloor \log N \rfloor$

パーセプトロン vs Winnow
加算型重み更新　　乗算型重み更新

- 学習目標が $v^* = (1, 0, 1, 0, 0, \ldots, 1)$ のとき
  ただし, $k = \sum_i v^*_i$
- パーセプトロン

  $v_i \leftarrow v_i + \eta\,(y-y')\,x_i$

  誤り回数 = $\Omega(kN)$

  カーネル手法と整合する
- Winnow

  $v_i \leftarrow v_i \exp(\eta\,(y-y')\,x_i)$ / 規格化

  誤り回数 = $O(k \log N)$　・・・ attribute efficient

  一般に, カーネル手法と整合しない

この問題と解析の特徴

- オンライン性
  - 予測と結果の提示が交互に繰り返される
- ユニバーサル性
  - エキスパートの予測や結果の系列に対して何の仮定もおかない
- 最悪値評価
  - アルゴリズムの性能を最悪の場合で評価する
- 相対評価
  - アルゴリズムの性能を，最適なエキスパートの成績を用いて相対的に評価する

オンライン予測

- モデル化
  - エキスパート統合モデル
- 統合アルゴリズム
  - Winnow (1988)
  - Aggregating Algorithm (1990)
  - Weighted Majority, Weighted Averaging (1993,1994,1999)
  - GD, EG family (1994)
  - Potential-Based (2001)
  - Following Perturbed Leader (2003)
- 応用
  - 学習，符号化，統計的推定，投資，ゲーム理論，最適化，…

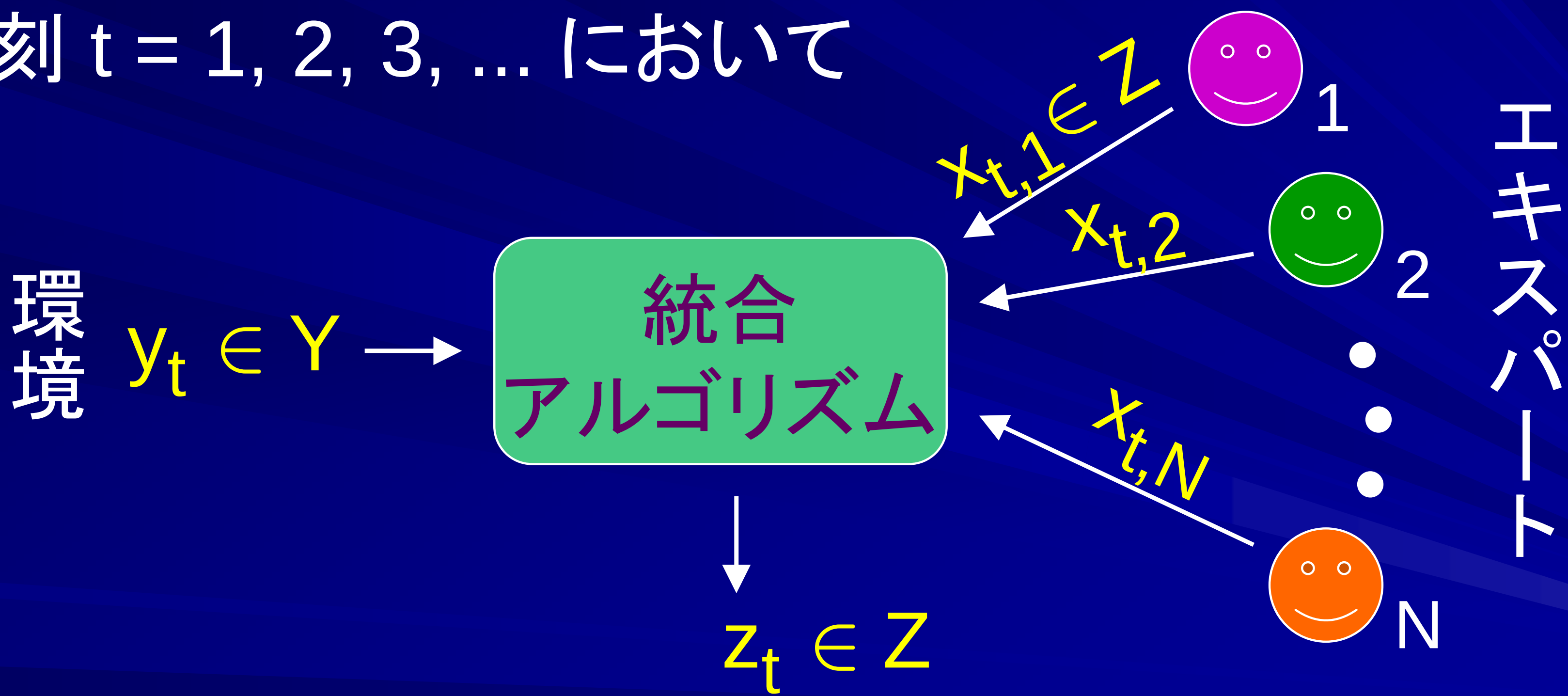
エキスパート統合モデル

Y ・・・真の結果値の集合

Z ・・・予測値の集合

$L: Y \times Z \to [0,\infty]$ ・・・損失関数

各時刻 $t = 1, 2, 3, \ldots$ において

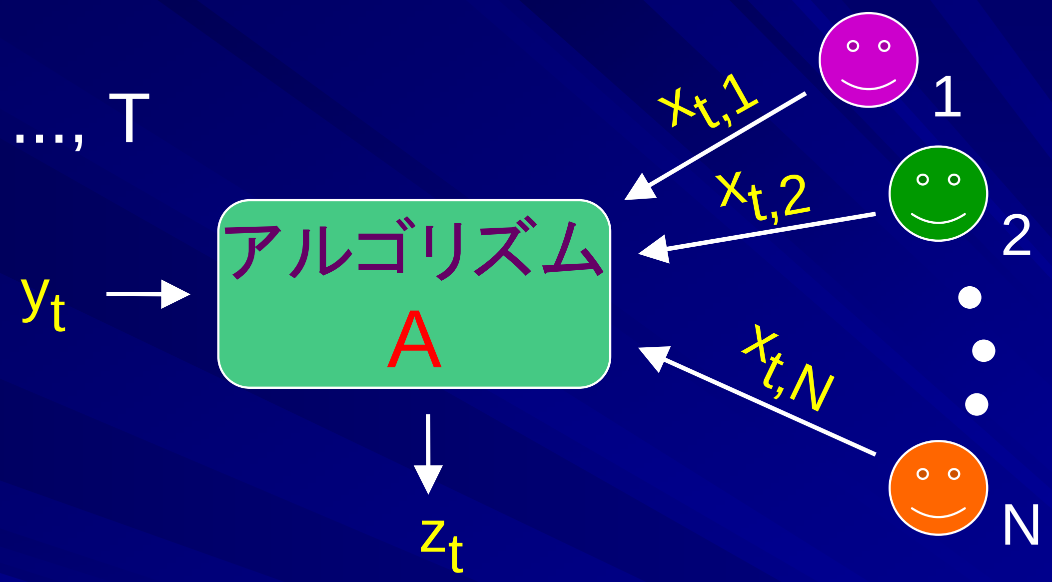


アルゴリズムの損失:　$L(y_t, z_t)$

エキスパート i の損失: $L(y_t, x_{t,i})$

相対損失(regret)

t = 1, 2, ..., T



- アルゴリズム A の累積損失: $\mathrm{Loss}_A^T = \sum_{t=1}^T L(y_t, z_t)$
- エキスパート i の累積損失: $\mathrm{Loss}_i^T = \sum_{t=1}^T L(y_t, x_{t,i})$
- アルゴリズム A の相対損失: $R_A^T = \mathrm{Loss}_A^T - \min_i \mathrm{Loss}_i^T$

目標: $R^T{}_A \to \text{small}$

クイズ王に匹敵する問題

- Y = {0,1}
- Z = {0,1}
- L(y, z) = |y − z|

$\left.\begin{array}{l}\\ \\ \\ \end{array}\right\}$ に相当

Loss^{T}_{A} = アルゴリズム A の誤り回数
Loss^{T}_{i} = エキスパート i の誤り回数
$m^* = \min_i \mathrm{Loss}^{T}_{i}$

$$\exists A, \quad R^T_A \leq 4\sqrt{m^* \ln N} + 2.5 \log_2 N$$

統合アルゴリズムの原理

Vovk の Aggregating Algorithm (AA)

- 各エキスパート i に対し，重み $v_{t,i}$ を保持
  $(v_{t,i} \geq 0, \sum_{1 \leq i \leq N} v_{t,i} = 1)$
- 予測の統合

$$\forall y \in Y, L(y, z_t) \leq -c(\eta) \ln \sum_{i=1}^{N} v_{t,i} e^{-\eta L(y, x_{t,i})}$$

を満たす $z_t \in Z$ を出力

c(η): 問題 (Y, Z, L) に固有の値

- 重み更新

$$v_{t+1,i} = \frac{v_{t,i} e^{-\eta L(y_t, x_{t,i})}}{\text{規格化係数}}$$

AAの性能

- 定理

任意のエキスパート，任意の環境の振る舞いに対し，最適なエキスパートを i^* = argmin$_i$ Loss$^T{}_i$ とすると，

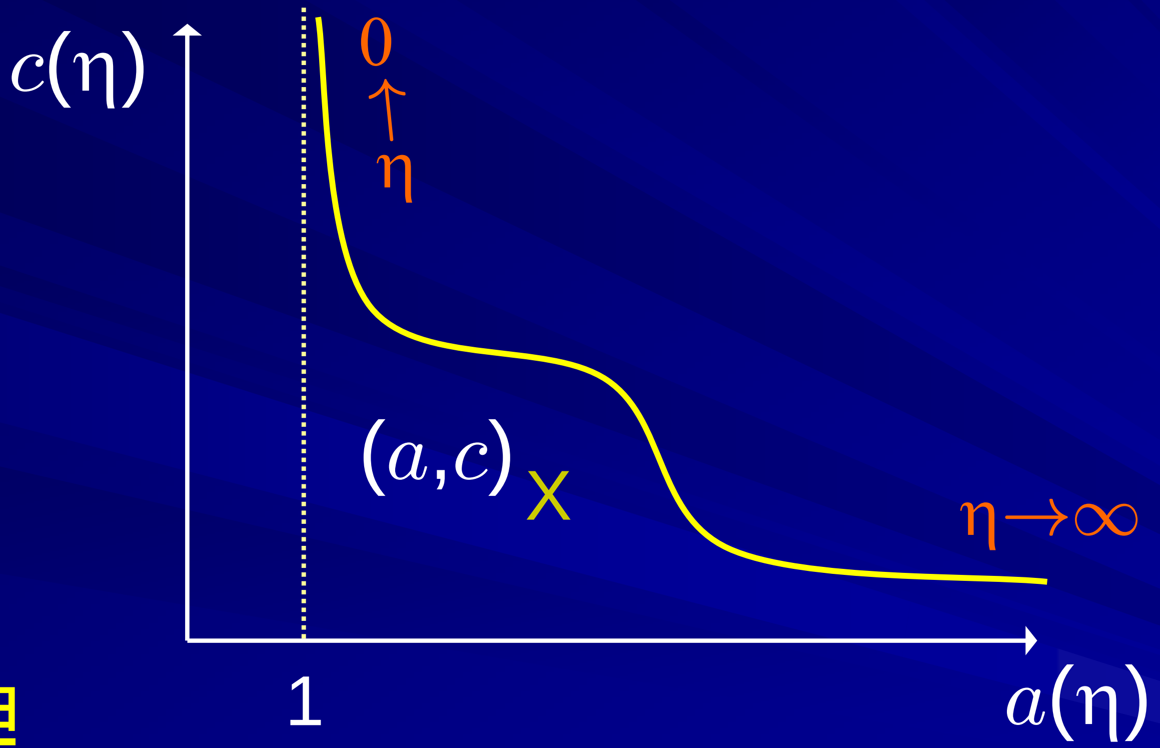
$$\mathrm{Loss}_{AA}^T \le c(\eta)\left(\eta \mathrm{Loss}_{i^*}^T + \ln \frac{1}{v_{1,i^*}}\right)$$

特に，初期重みが一様（ $v_{1,i}$ = 1/N）のとき，

$$\begin{aligned}\mathrm{Loss}_{AA}^T &\le c(\eta)\left(\eta \mathrm{Loss}_{i^*}^T + \ln N\right)\\ &\equiv a(\eta)\mathrm{Loss}_{i^*}^T + c(\eta)\ln N\end{aligned}$$

AAの最適性

$$\mathrm{Loss}_{AA}^{T} \le a(\eta)\mathrm{Loss}_{i^*}^{T} + c(\eta)\ln N$$



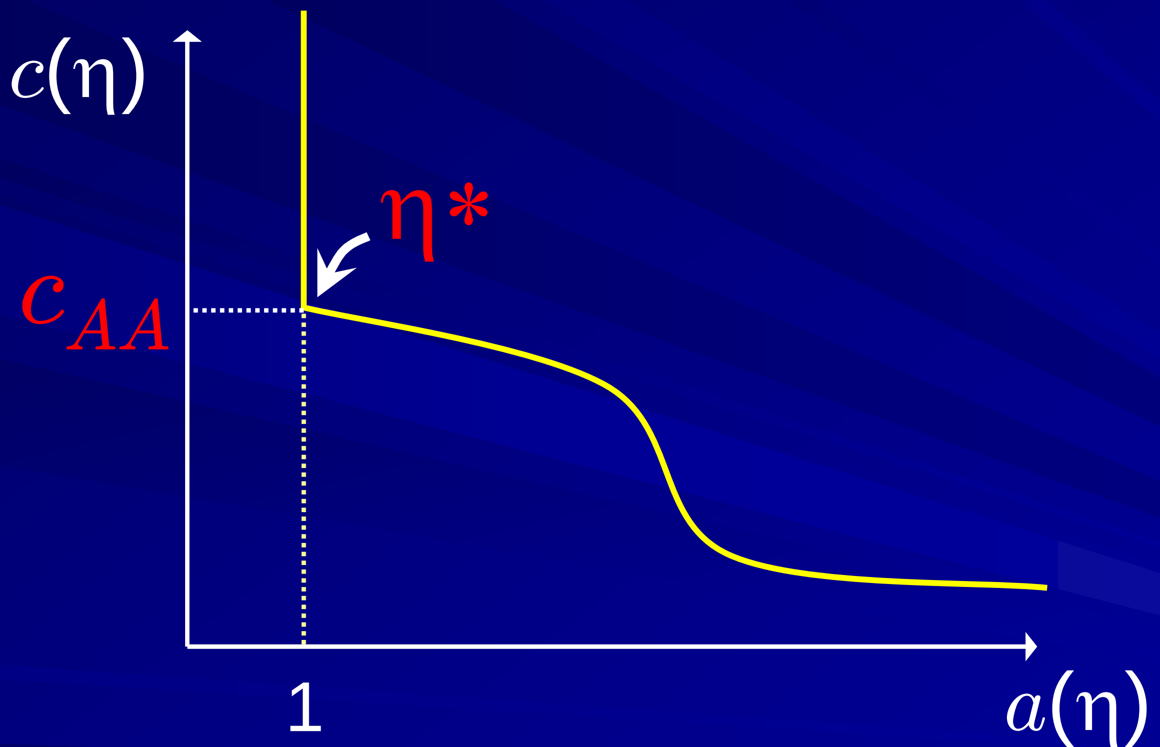
- 定理

問題 (Y, Z, L) が“自然”ならば,
∀A, ∃エキスパート, ∃環境

$$\mathrm{Loss}_{A}^{T} > a\mathrm{Loss}_{i^*}^{T} + c\ln N$$

凸損失関数に対する性質

L(y,z) が z に関して下に凸のとき
（２次損失，相対エントロピー損失，Hellinger損失，．．．）



$$\mathrm{Loss}_{AA}^{T} \le \mathrm{Loss}_{i^*}^{T} + c_{AA} \ln N$$

Weighted Averaging Algorithm (WAA)

- 簡便なアルゴリズム（AAの近似）
- Y = Z = [0,1], L：凸損失関数と仮定
- 予測の統合

$$z_t = \sum_{i=1}^{N} v_{t,i} x_{t,i}$$

- 定理

$$\mathrm{Loss}_{WAA}^{T} \leq \mathrm{Loss}_{i^*}^{T} + c_{WAA} \ln N$$

$$(c_{AA} \leq c_{WAA})$$

さまざまな凸損失関数と c_{AA}, c_{WAA}

$$\mathrm{Loss}^T_{AA} \le \mathrm{Loss}^T_{i^*} + c_{AA} \ln N$$

$$\mathrm{Loss}^T_{WAA} \le \mathrm{Loss}^T_{i^*} + c_{WAA} \ln N$$

損失関数	$L(y, z)$	c_{AA}	c_{WAA}
2次損失	$(y - z)^2$	$1/2$	2
相対エントロピー損失	$(1-y)\ln\frac{1-y}{1-z} + y\ln\frac{y}{z}$	1	1
Hellinger損失	$\frac{1}{2}((\sqrt{1-y} - \sqrt{1-z})^2 + (\sqrt{y} - \sqrt{z})^2)$	$1/\sqrt{2}$	1
絶対値損失	$\lvert y - z \rvert$	凸関数ではない	

ダイバージェンスを用いた導出

- KLダイバージェンス（相対エントロピー）
  確率ベクトル v, u に対し,
  $$\mathrm{RE}(v \,\|\, u) = \sum_i v_i \ln (v_i \,/\, u_i)$$

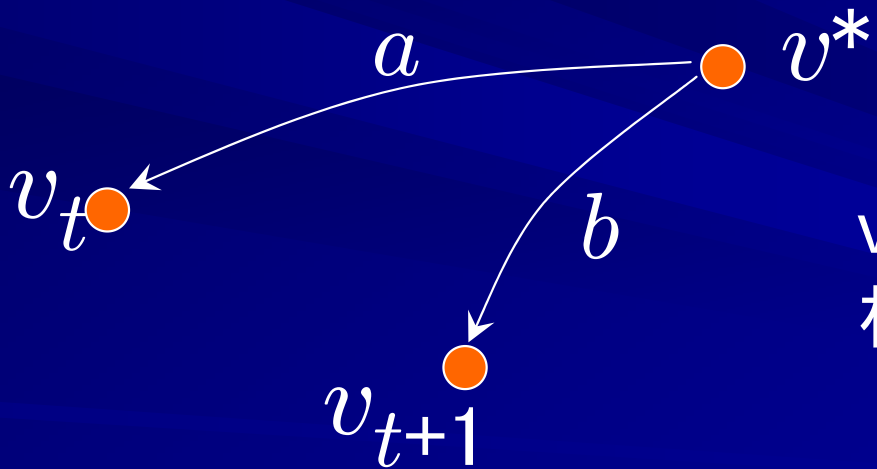
- 重み更新の導出
  $$v_{t+1} = \mathrm{argmin}_v \left(\eta \sum_i v_i L(y_t, x_{t,i}) + \mathrm{RE}(v \,\|\, v_t)\right)$$

  $$= \frac{v_{t,i} e^{-\eta L(y_t, x_{t,i})}}{\text{規格化係数}}$$

上界の導出

- $v^* = (0, \ldots, 1, \ldots., 0) \in [0,1]^N$ （第 i^* 成分のみ1）

$$\boxed{\begin{aligned} &\mathrm{L}(y_t, z_t) - \mathrm{L}(y_t, x_{t,i^*}) \\ &\qquad \le c\,\big(\mathrm{RE}(v^* \,\|\, v_t) - \mathrm{RE}(v^* \,\|\, v_{t+1})\big) \end{aligned}}$$



v_t が v^* に“時刻 t における”相対損失分以上近づく

- t=1, ..., T で和を取ると,

相対損失 $\le$ c $\mathrm{RE}(v^* \,\|\, v_1)$ = c ln N

オンライン予測の応用

オンラインアルゴリズムの統合

- オンライン問題を解くアルゴリズム $A_1, \ldots, A_N$
- 入力系列 $S = y_1, \ldots, y_T$

$\max_S \min_i A_i(S)$ の損失 $\leq \min_i \max_S A_i(S)$の損失

AA の損失 $\leq \min_i A_i$の損失 $+ c_{AA} O(\ln N)$

確率モデルの統合と情報圧縮

・・・ so she was considering in her ___?___

↓

言語モデル θ_i

↓

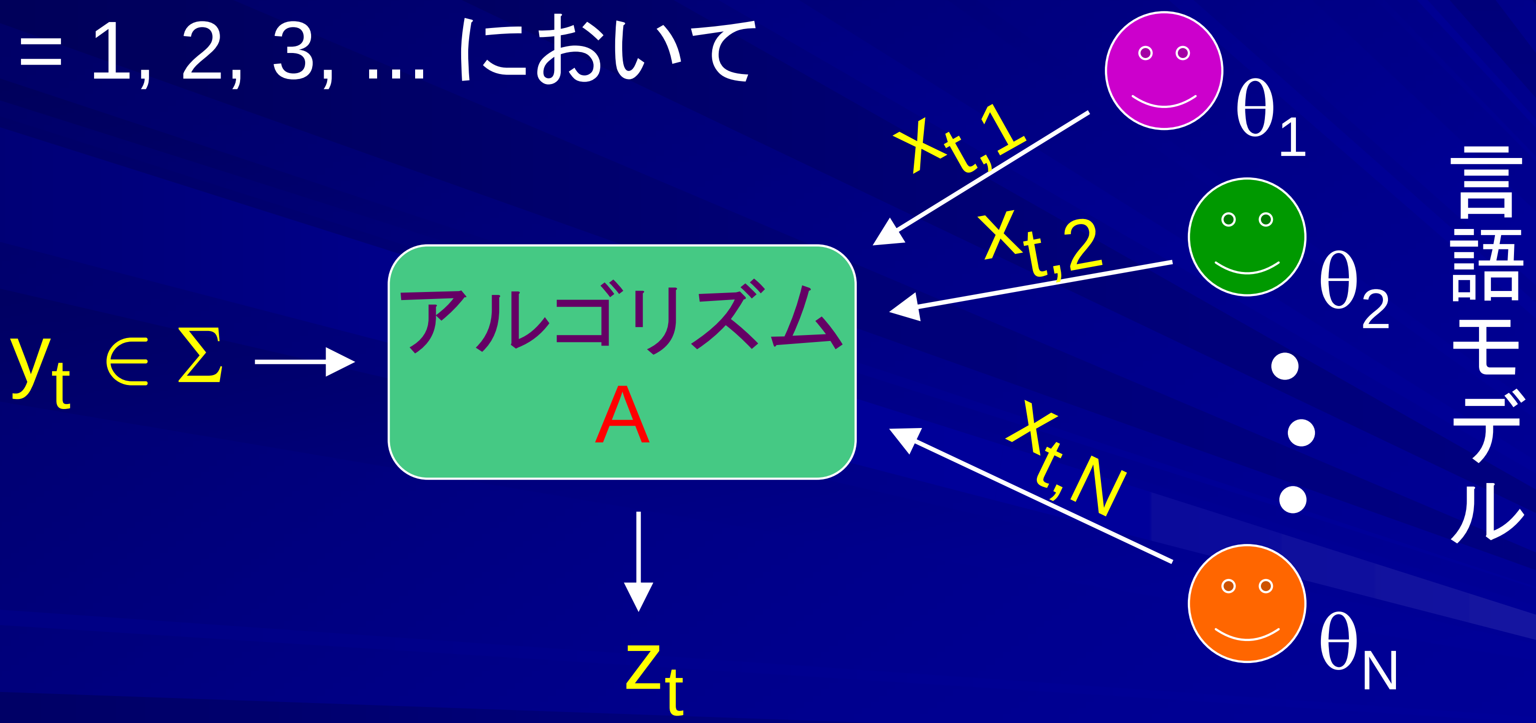
	Pr($\cdot$ | context, θ_i)
mind	0.232
thought	0.183
own	0.129
office	0.073
:	:

確率モデルの統合モデル

$\Sigma = \{1, ..., K\}$：アルファベット

$y_1, y_2, ..., y_{t-1} \in \Sigma \longrightarrow \boxed{\theta_i} \longrightarrow x_{t,i}(\cdot) = \Pr(\,\cdot \mid y_1, ..., y_{t-1}, \theta_i)$

各時刻 t = 1, 2, 3, ... において

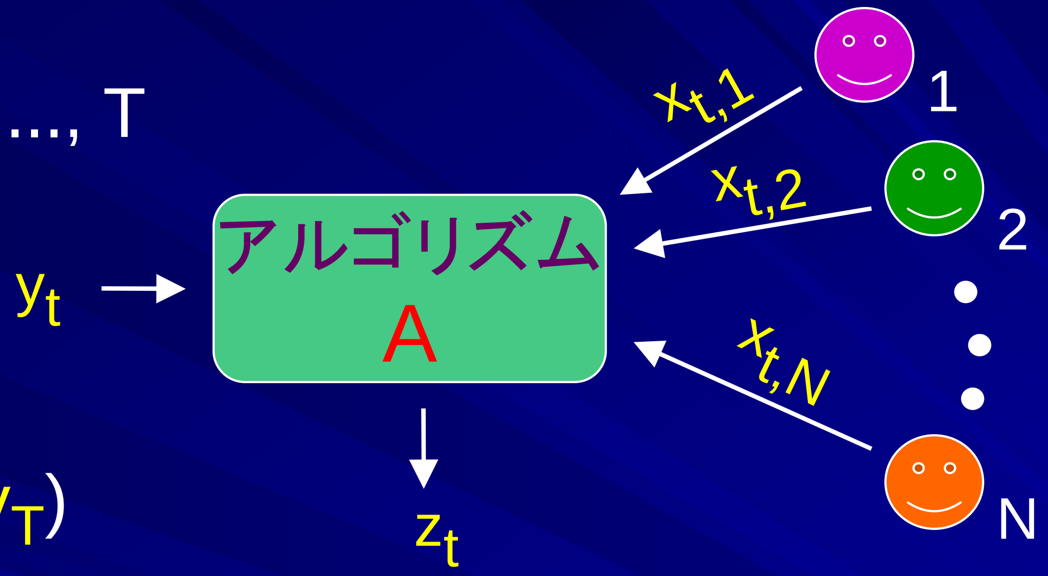


アルゴリズムの尤度: $z_t(y_t) = \Pr(y_t \mid y_1, ..., y_{t-1}, A)$

モデル θ_i の尤度: $x_{t,i}(y_t) = \Pr(y_t \mid y_1, ..., y_{t-1}, \theta_i)$

ユニバーサル符号と冗長度

t = 1, 2, ..., T



系列 S = (y_1, ..., y_T)

- Aの尤度: P(S | A) = $\prod_t$ P(y_t | $y_{1..t\text{-}1}$, A) = $\prod_t$ z_t(y_t)
- θ_i の尤度: P(S | θ_i) = $\prod_t$ $x_{t,i}$(y_t)
- A + ユニバーサル符号 ⇒ 符号長 = – $\log_2$ P(S | A)
  θ_i + ユニバーサル符号 ⇒ 符号長 = – $\log_2$ P(S | θ_i)
- 冗長度 = – $\log_2$ P(S | A) – $\min_i$ (– $\log_2$ P(S | θ_i))

AAの適用とベイズ混合

以下の問題 (Y, Z, L) と等価

- Y = Σ = {1, ..., K}
- Z = { z ∈ $[0,1]^K$: Σ 上の分布 }
- L(y, z) = −ln z(y) ・・・ 対数損失
  ⇒ $\mathrm{Loss}^T{}_A$ = − ln P(S | A) ・・・ 符号長
- AA の出力 = WAA の出力 = ベイズ混合

$$
\begin{aligned}
z_t(j) &= \sum_{i=1}^{N} v_{t,i} x_{t,i}(j) \\
&= \sum_{i=1}^{N} \mathrm{Pr}(\theta_i \mid y_{1..t-1}) \, \mathrm{Pr}(j \mid y_{1..t-1}, \theta_i)
\end{aligned}
$$

冗長度 = 相対損失 = $\mathrm{Loss}^T{}_A - \mathrm{Loss}^T{}_{i^*} \leq \ln N$

無限のエキスパートの統合

- $Y = \{0,1\}$
- エキスパート $\theta \in [0,1]$

  $x_{t,\theta}(1) = \Pr(1 \mid y_{1..t-1}, \theta) = \theta$

  $x_{t,\theta}(0) = \Pr(0 \mid y_{1..t-1}, \theta) = 1 - \theta$
- AAの出力 = KT-Estimator

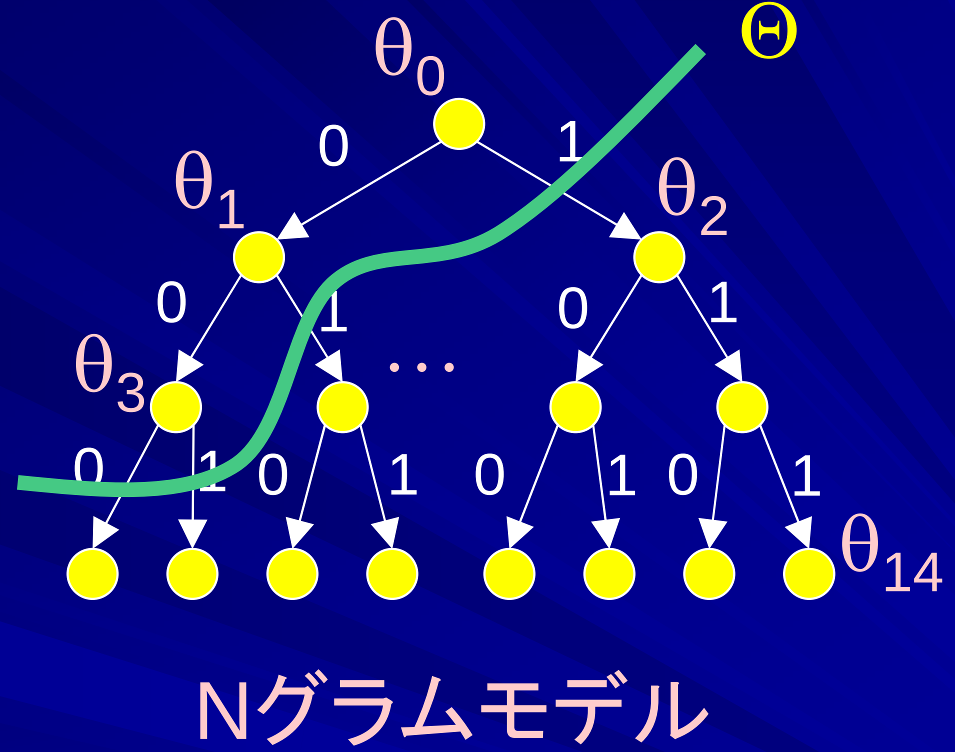
$$
\begin{aligned}
z_t(y) &= \int v_{t,\theta} x_{t,\theta}(y) d\theta \\
&= \frac{\sum_{s=1}^{t-1} y_s + 1/2}{t}
\end{aligned}
$$

- 相対損失 = O(log T)

Context Tree Weighting

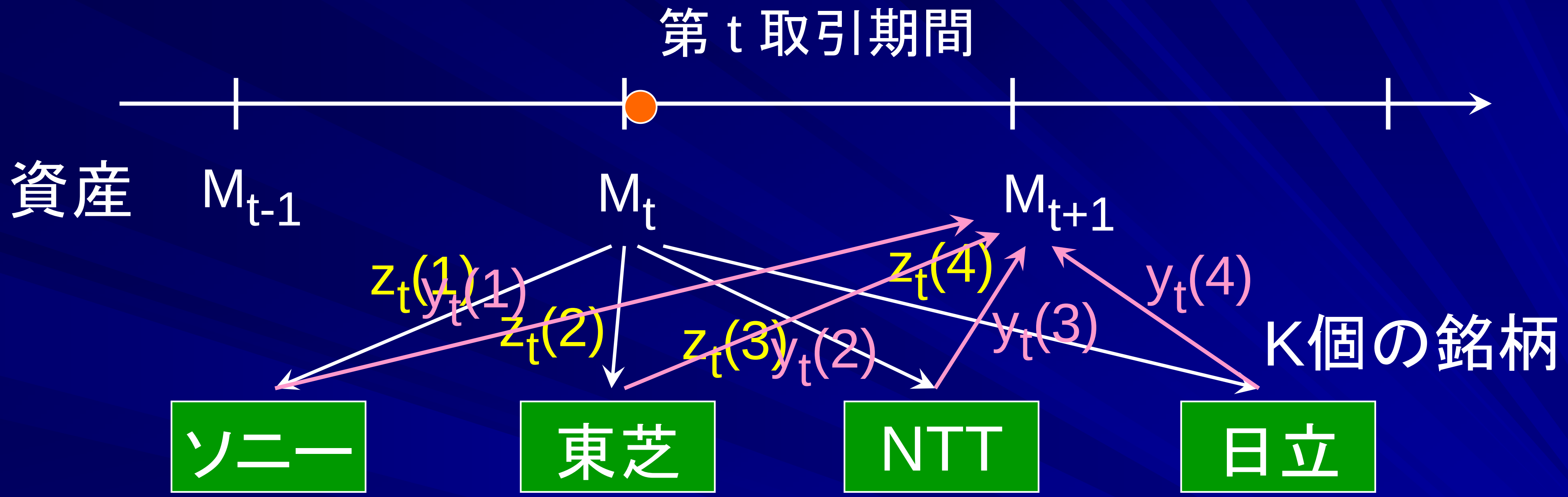
各枝刈り Θ：モデル

Pr(y | ･･･001, Θ) = θ_2
Pr(y | ･･･010, Θ) = θ_4



符号長 = 最適な Θ の符号長 + O(Θ の葉の数)

ポートフォリオ



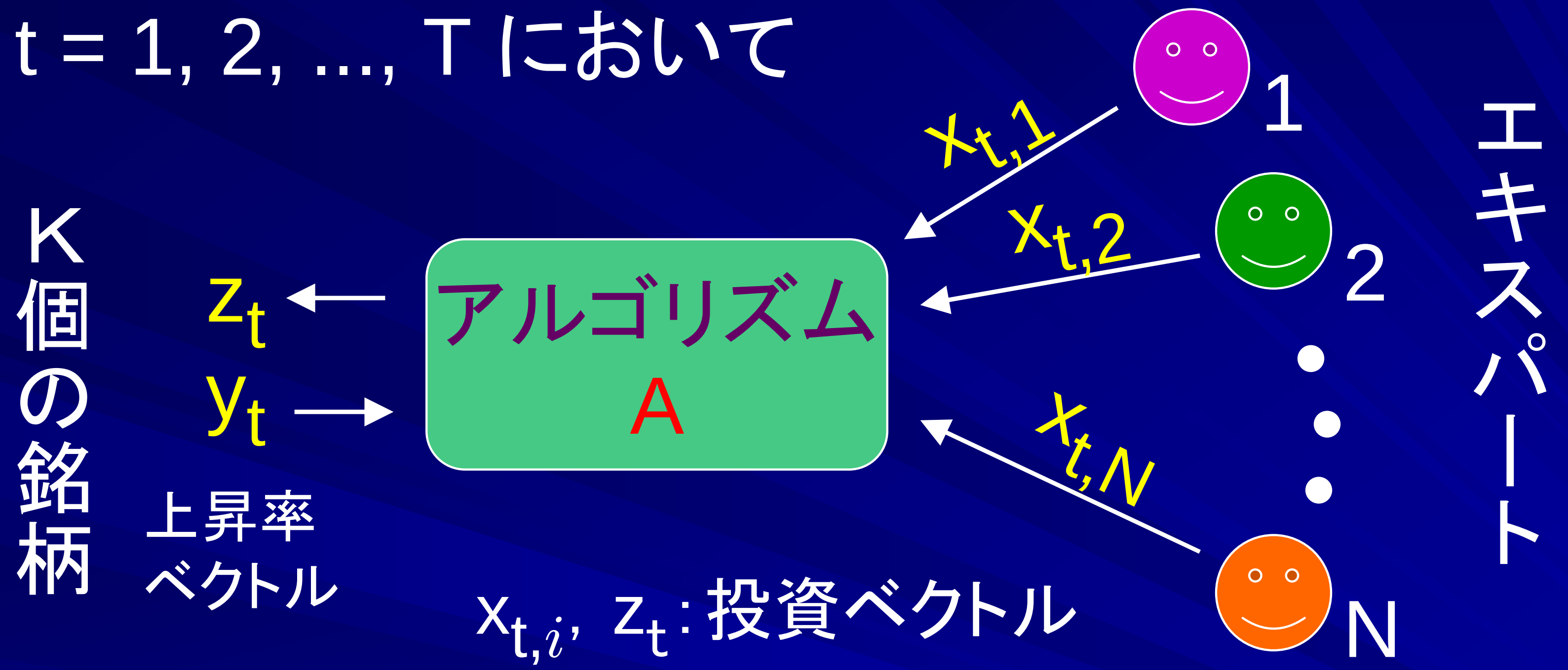
- 全額投資. 銘柄 j の株を $M_t\ z_t(j)$ の金額で購入
  ($z_t(1) + \cdots + z_t(K) = 1$)

- 株をすべて売却. 銘柄 j の株価の上昇率を $y_t(j)$ とする

新しい資産 $M_{t+1} = M_t \sum_j y_t(j)\ z_t(j) = M_t\ (y_t \cdot z_t)$

投資アドバイスの統合

各時刻 t = 1, 2, ..., T において



- Aの資産: $M_{T+1} = M_1 \prod_t z_t \cdot y_t$
- エキスパート i のアドバイスに従ったときの資産:

$$M_{T+1,i} = M_1 \prod_t x_{t,i} \cdot y_t$$

- 競合比: $M_{T+1} / \max_i M_{T+1,i}$

AAの適用

以下の問題 (Y, Z, L) と等価

- $Y = [0, \infty]^K$
- $Z = \{ z \in [0,1]^K \mid z(1) + \cdots + z(K) = 1 \}$
- $L(y, z) = -\ln (y \cdot z)$ ・・・ ポートフォリオ損失
- $\mathrm{Loss}^T_A = -\ln M_{T+1}$
- AA の出力 = WAA の出力

$$z_t = \sum_{i=1}^{N} v_{t,i} x_{t,i}$$

$$\boxed{\text{競合比} = e^{-\text{相対損失}} \geq 1 / N}$$

オンライン投資

- ポートフォリオ問題において，毎時刻全額投資するかわりに，定額（例えば1万円）投資
- 株価の上昇率に上限 B > 0 を仮定（相対損失を有限に抑えるため）

以下の問題 (Y, Z, L) と等価

- $Y = [0, B]^K$
- $Z = \{ z \in [0,1]^K \mid z(1) + \cdots + z(K) = 1 \}$
- $L(y, z) = y \cdot z$　　・・・内積損失

WAAの適用

- 内積損失 L(y, z) = y・z は凸ではない
- 適当な η に対して

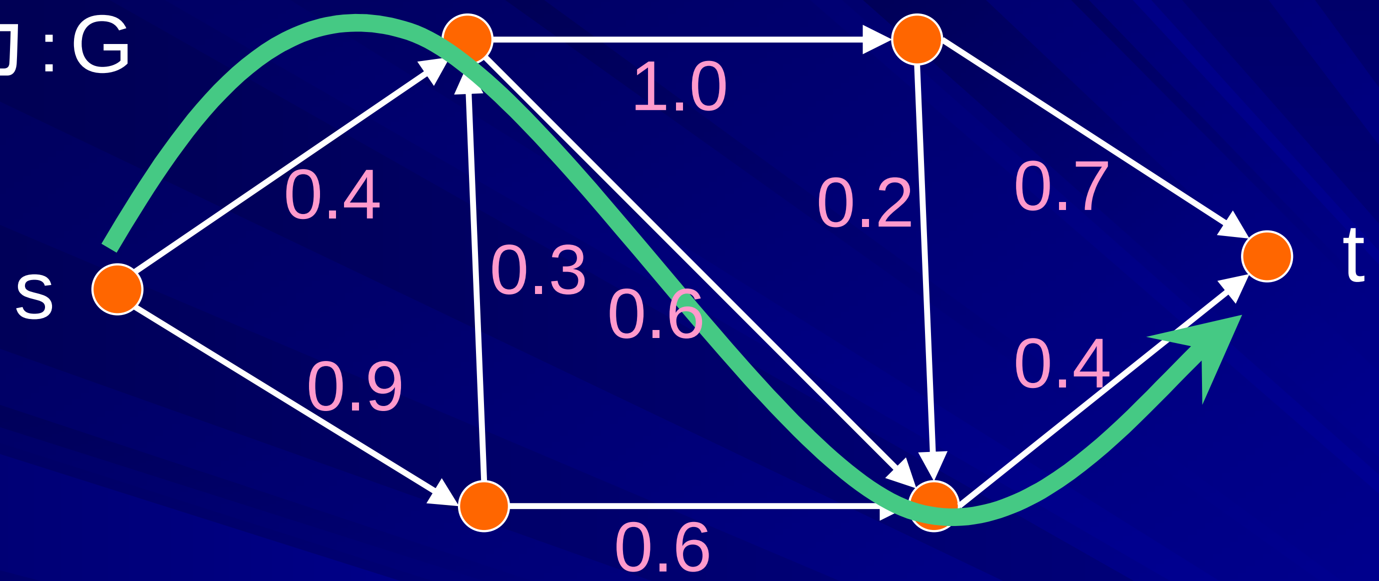
相対損失 = $O\left(\sqrt{2BL^* \ln N} + B \ln N\right)$

L* = 最適なエキスパートの損失

Path Kernel 法

オンライン最短路問題

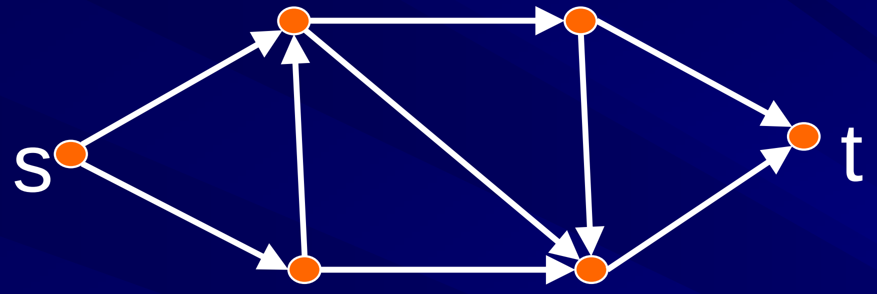
入力：G



1. あるs-t 道 P を(確率的に)選ぶ. この確率を $z_t(P)$ とする.
2. 各辺の遅延時間 $d_t(e) \in [0,1]$ が明らかとなる.
   道Pに沿って s から t に到達する時間は $y_t(P) = \sum_{e \in P} d_t(e)$
3. 到達時間の期待値 $\sum_P y_t(P)\, z_t(P) = y_t \cdot z_t$

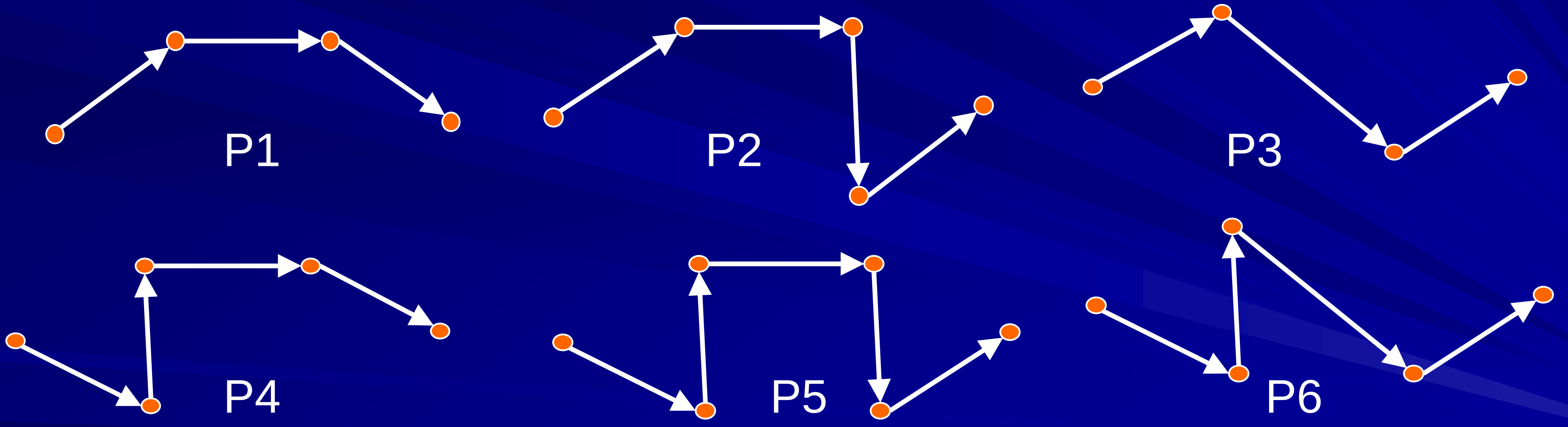
目標: $\sum_t y_t \cdot z_t - \min_P \sum_t y_t(P) \rightarrow$ 小

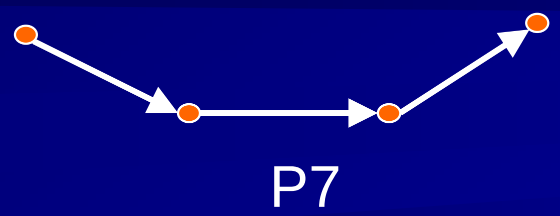
オンライン投資への還元



- $B = \max_P |P|$
- $Y = [0,B]^N$
- Z = s-t道上の分布の集合
- $L(y, z) = y \cdot z$

エキスパート

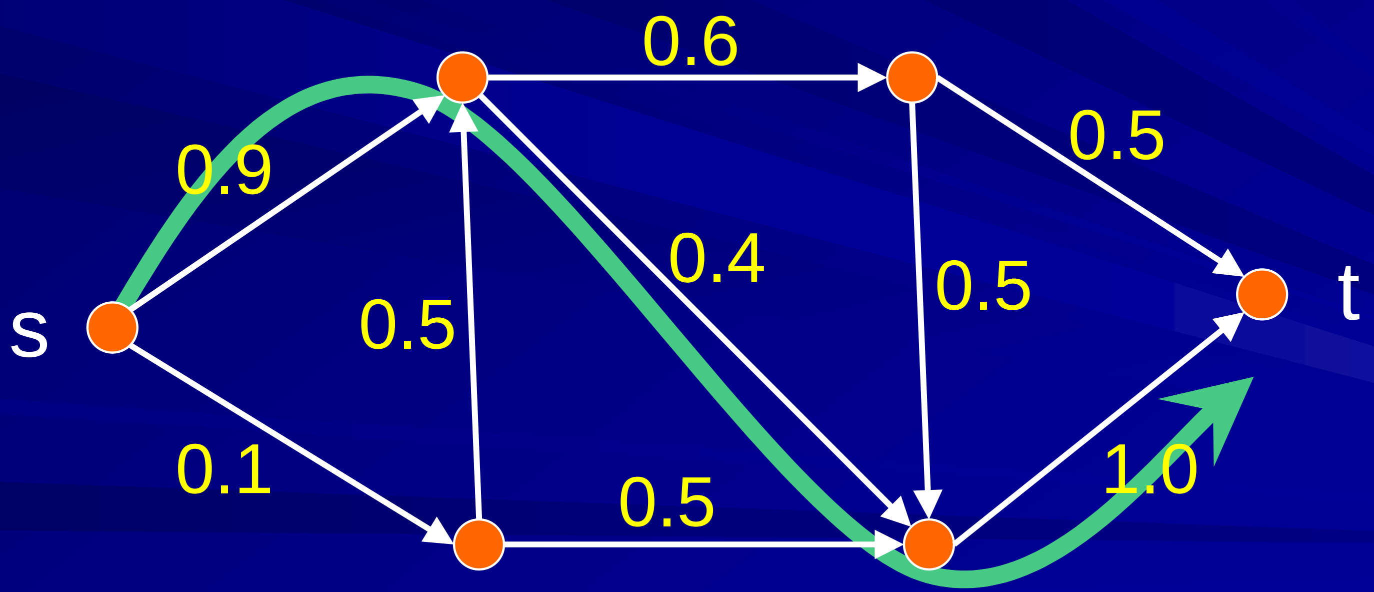




- エキスパートi は道 Pi を確率1で選ぶ
- N = s-t 道の総数

Path Kernel 法

- N(道の総数)は指数的になりうる
- すべての道 P に対し，重み $v_{t,P}$ を保持できない
- その代わり，各辺 e に対し，遷移確率 $a_t(e)$ を保持
- 関係式 $v_{t,P} = \prod_{e \in P} a_t(e)$ によって間接的に $v_{t,P}$ を表現



$v_{t,P} = 0.9 * 0.4 * 1.0 = 0.36$

重み更新の模倣

条件:

（更新前）　$v_{t,P} = \Pi_{e \in P}\, a_t(e)$

$$v_{t+1,P} = \frac{v_{t,P}\, e^{-\eta y_t(P)}}{\text{normalize}}$$

（更新後）　$v_{t+1,P} = \Pi_{e \in P}\, a_{t+1}(e)$

を満たすような辺上の重み更新

$$a_t \to a_{t+1}$$

を行う必要がある

重み更新の模倣

- 任意の P に対し，次の式を満たす a_{t+1} を求めればよい

$$v_{t+1,P} = \prod_{e\in P} a_{t+1}(e) = \frac{\prod_{e\in P} a_t(e) b_t(e)}{\sum_P \prod_{e\in P} a_t(e) b_t(e)}$$

ただし，$b_t(e) = \exp(-\eta\ d_t(e))$

道 P の損失 = 辺の損失の和
$(L(y_t, x_{t,P}) = \sum_{e\in P} d_t(e))$

Path Kernels

- 辺の重みベクトル: **a** = (a(e_1), ..., a(e_m))
- パスの重みベクトル: **v** = (v_{P1}, ..., v_{PN}),,

$$\forall P, v_P = \prod_{e \in P} a(e)$$

この関係を, **v = Φ(a)** と表す.

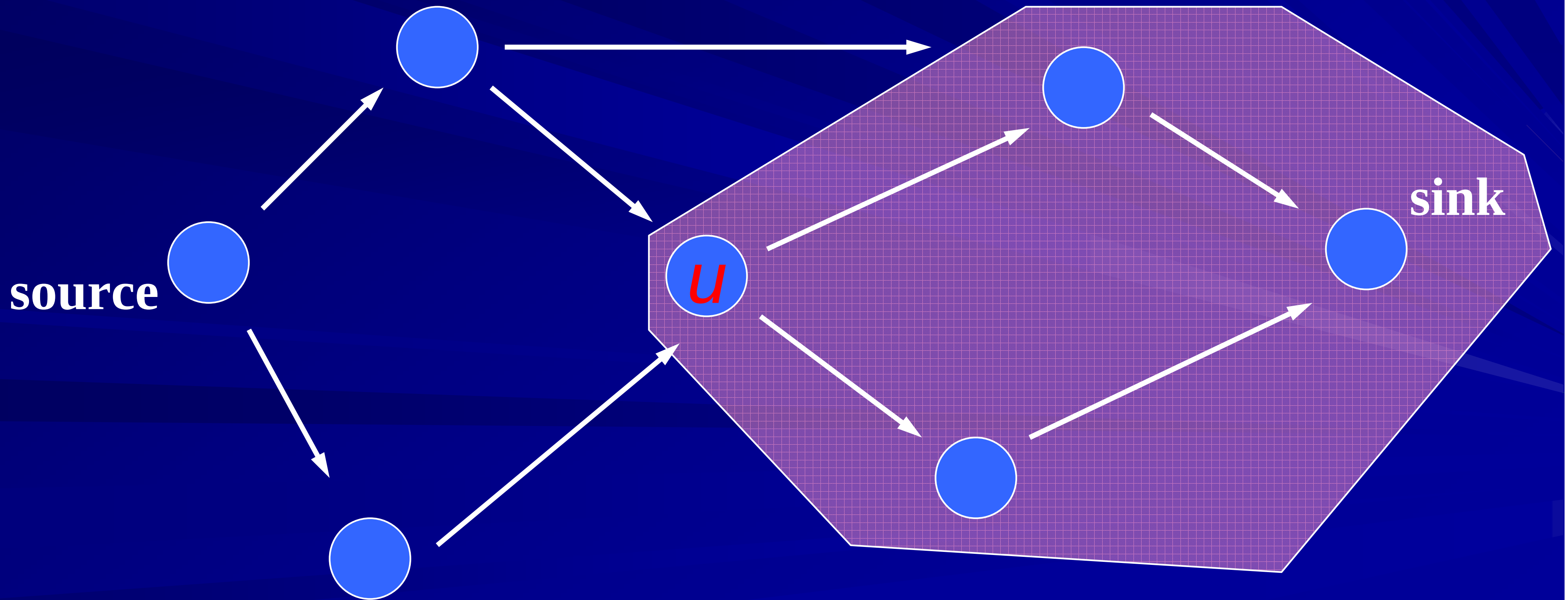
- 係数ベクトル : **b** = (b(e_1), ..., b(e_m))

Path Kernel

$$K(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{\Phi}(\boldsymbol{a}) \cdot \boldsymbol{\Phi}(\boldsymbol{b}) = \frac{\sum_P \prod_{e \in P} a(e) b(e)}{\text{normalization factor}}$$

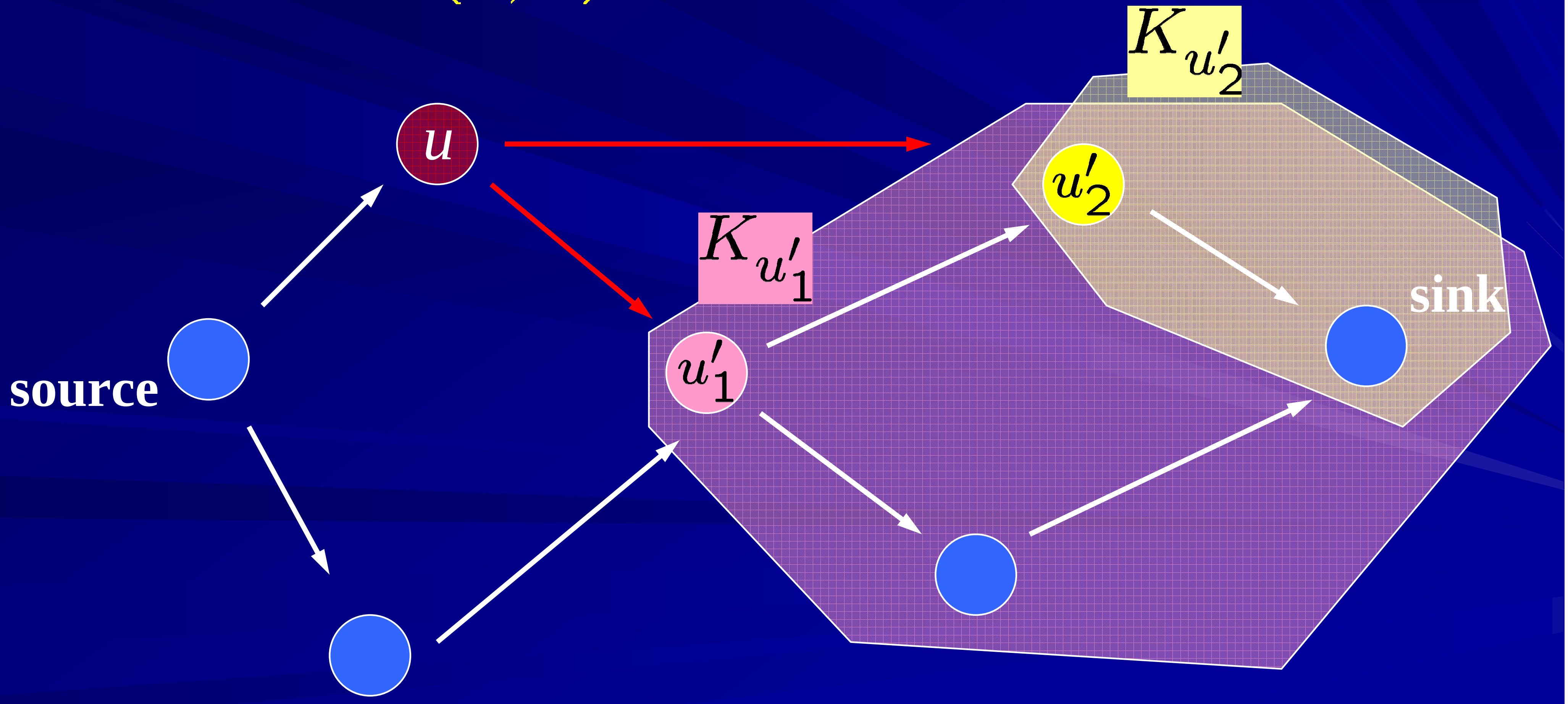
Path Kernel の計算

$$K_u = \sum_{P:\text{from } u} \prod_{e \in P} a(e)b(e)$$



Path Kernel の計算

- $K_{\mathsf{sink}} = 1$
- $K_u = \displaystyle\sum_{u' : (u,u') \in E} v(u,u') b(u,u') K_{u'}$

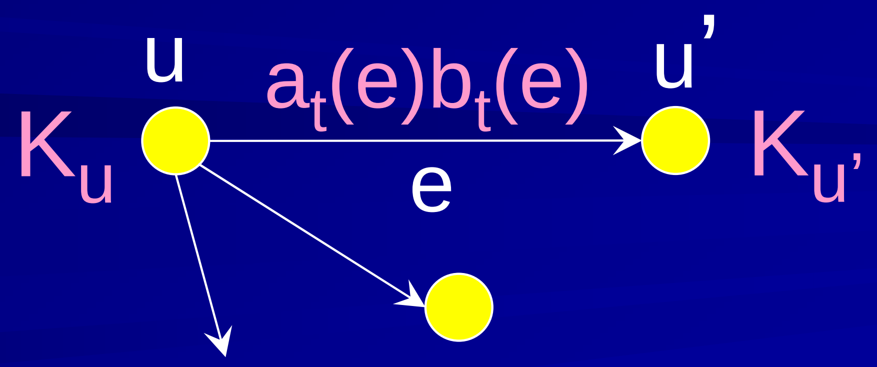


Weight Pushing Algorithm

- 目標 $\forall P, \prod_{e \in P} a_{t+1}(e) = \frac{\prod_{e \in P} a_t(e) b_t(e)}{K_{\mathrm{source}}(\boldsymbol{a}_t, \boldsymbol{b}_t)}$
- 漸化式 $K_u = \sum_{u' : (u,u') \in E} a_t(u,u') b_t(u,u') K_{u'}$

$$\boxed{a_{t+1}(e) = \frac{a_t(e) b_t(e) K_{u'}}{K_u}}$$

目標達成！



まとめ

- さまざまなオンライン最適化問題が，共通の統合スキーム（AA, WAA）を用いて解ける
- 一般の最適化問題への適用
  - 各時刻 t における損失が，それまでの履歴に依存する
  - オフラインで最適な解を求めるのが困難
- 構造的エキスパートの統合の可能性
  - 決定リストの効率の良いオンライン学習
- モデルのさらなる拡張
  - リスク情報を用いた統合