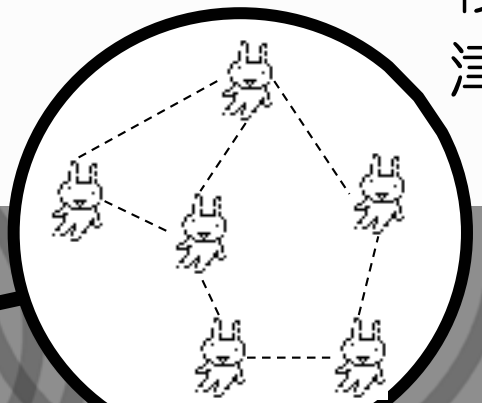
IBM research

Tokyo Research Laboratory

リンク伝播法：リンク予測のための半教師付き学習法

鹿島 久嗣　（IBM東京基礎研究所）
加藤 毅　（お茶の水女子大学）
山西 芳裕　（パリ国立高等鉱業学校）
杉山 将　（東京工業大学）
津田 宏治（マックスプランク研究所）



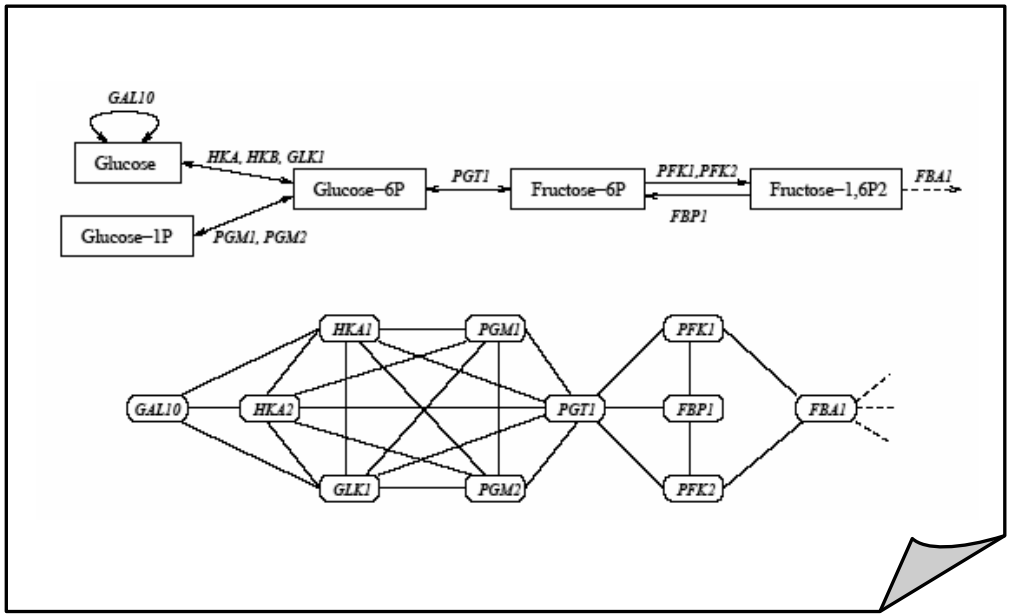
Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama and Koji Tsuda:
Link Propagation: A Supervised Learning Algorithm for Link Prediction, In Proc. 2009 SIAM Conference on Data Mining (SDM), 2009.

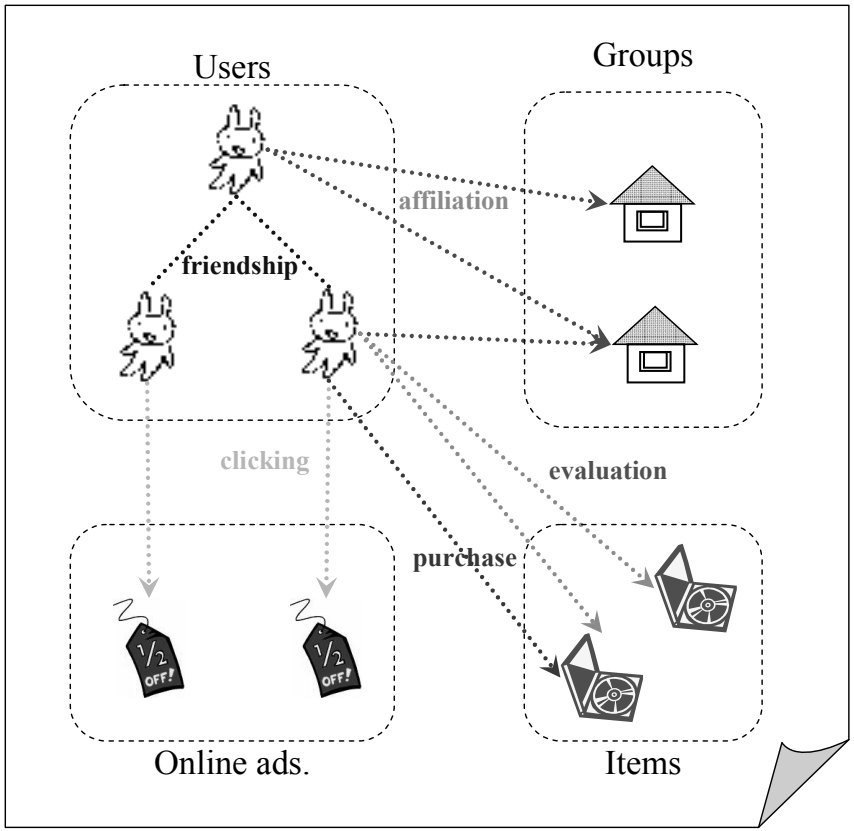
ネットワークのリンク予測のための新しい学習法を提案します

- ネットワーク構造： データ間の「関係」の表現

–例）

- WWW（Webページ同士の関係）
- ソーシャルネットワーク
  （ユーザー間の関係）
- 生体ネットワーク（タンパク質間の関係）



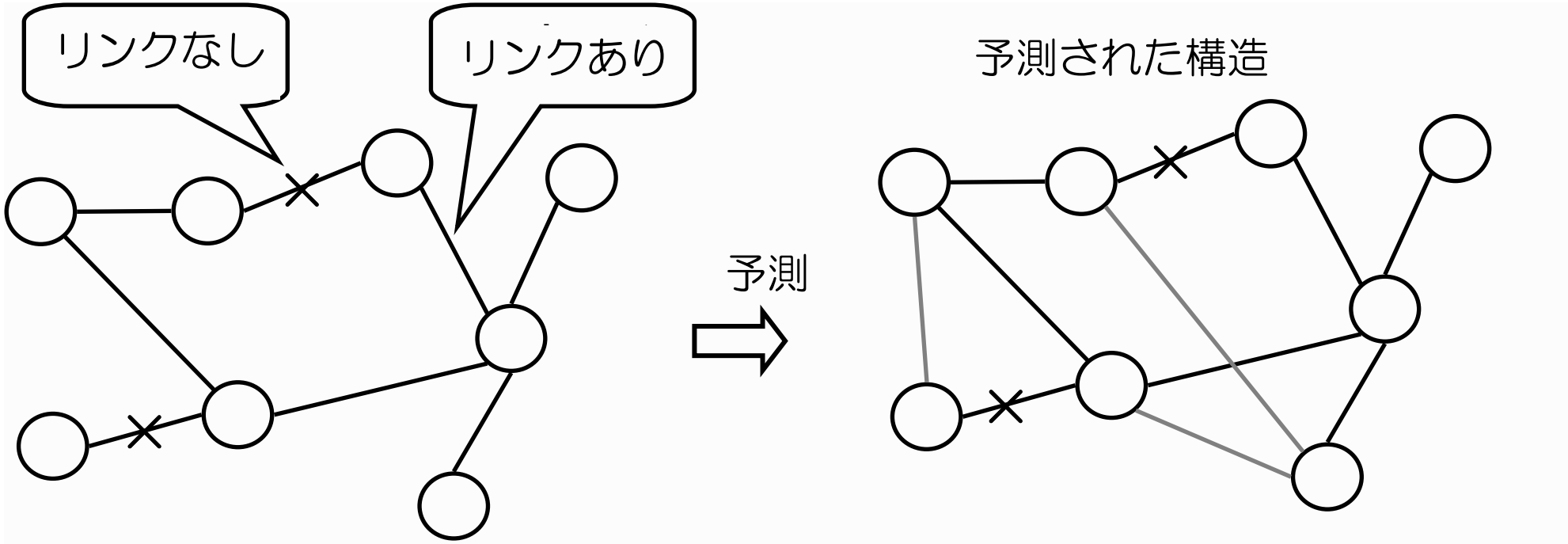


リンク予測問題とは、部分的に観測されているネットワーク構造から、残りの構造を推定する、教師付き予測問題です

–入力: 一部が欠けたネットワーク構造

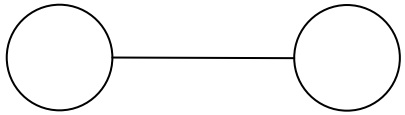
- リンクありのノードペア
- リンクのないノードペア

–出力： リンクの有無が未知のノードペアについてのリンク予測

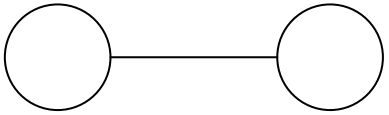


リンク予測問題には、生体ネットワーク予測や、オンラインマーケティングなどの応用があります

- 生体ネットワーク分析： 2つのたんぱく質の相互作用の有無を予測する
  - 相互作用あり＝リンクあり ／ 相互作用なし＝リンクなし
- オンライン・マーケティング： ユーザーと、商品の間の購買関係を予測する
  - 買う＝リンクあり ／ 買わない＝リンクなし



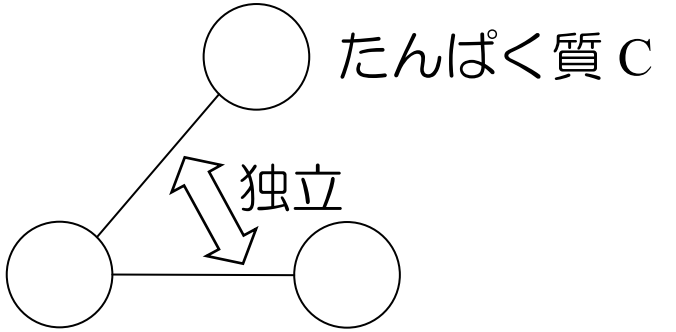
たんぱく質 A　たんぱく質 B
生体ネットワーク予測



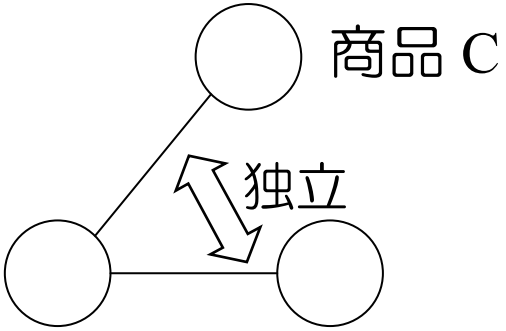
ユーザ A　商品 B
オンライン・マーケティング

リンク予測のひとつの捉え方は「ノードのペアの教師つき分類問題」です

- 問題を単純化するための仮定：
  各リンクの有無は、互いに独立であるとする
- これによって、通常の教師付き分類の枠組みで議論できる
  – ノードのペアを入力として、リンクの有無を出力
- 独立性を仮定しないモデルは、計算量的な困難をもつ
  – e.g. 関係マルコフネットワーク



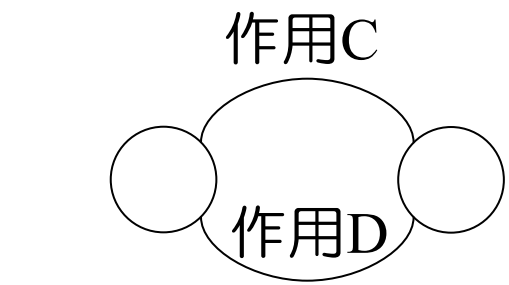
たんぱく質 A　たんぱく質 B
生体ネットワーク予測



ユーザ A　商品 B
オンライン・マーケティング

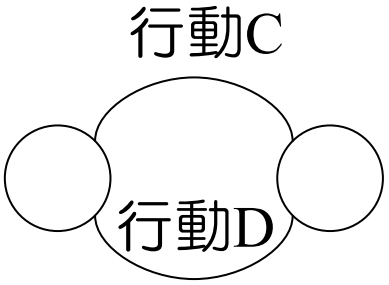
今回扱う「複数タイプリンク予測」は、リンクの種類を考えることで、より詳細なモデル化を可能にします

- 複数タイプリンク予測は、「リンクの種類」を考える
  - 生体ネットワーク分析：作用の種類や、作用が起こる環境、など
  - オンライン・マーケティング：購買、評価、商品情報の閲覧、など
- 異なるタイプのリンク間の相関を利用することで、予測精度があがる可能性がある



たんぱく質 A　たんぱく質 B

生体ネットワーク予測



ユーザ A　商品 B

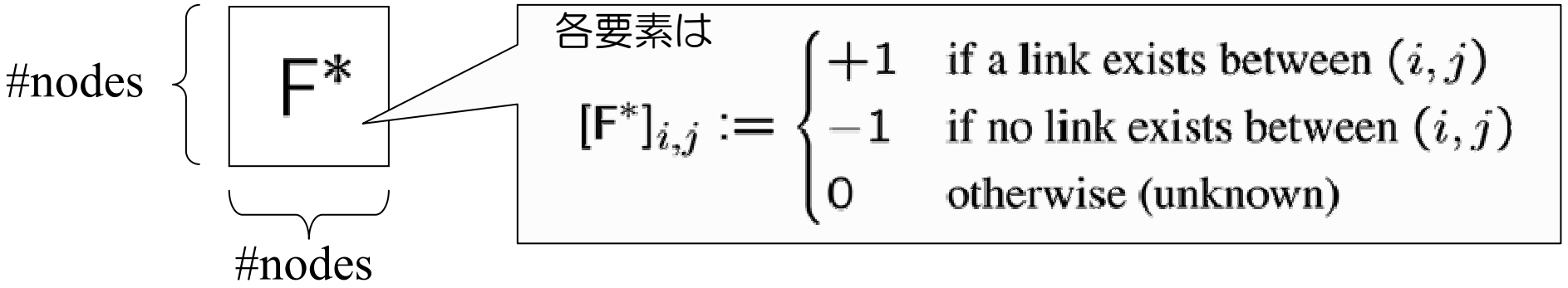
オンライン・マーケティング

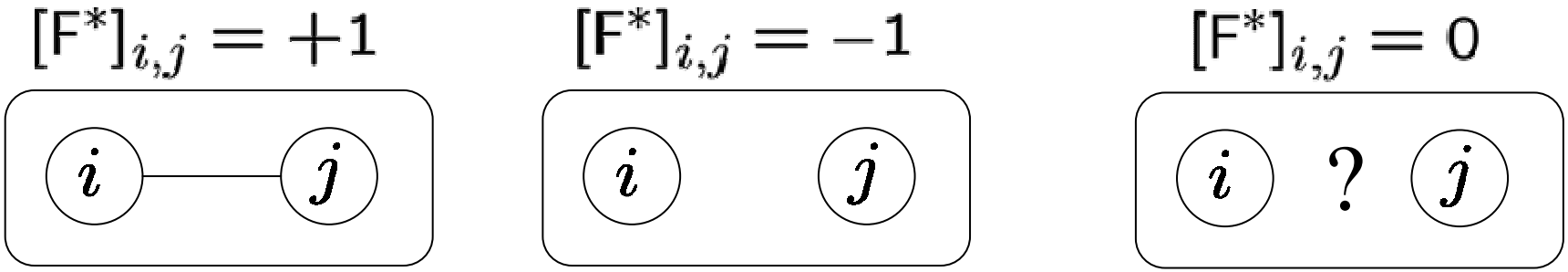
このあとの話の流れ

- 「複数タイプリンク予測」の定式化
  - 類似度情報を用いたテンソル補完の問題として定式化する
- 我々のアプローチ： 半教師付学習「ラベル伝播」の適用
  - 扱いやすい類似度の定義
  - 共役勾配法の高速化による効率的な解法
- 実験結果
  - 複数種リンクの同時予測によって予測精度が上がる
  - 提案手法は、同種の手法と比較して遥かに高速である

まず、単一タイプのリンク予測問題が、行列補完の問題として捉えられることを見ます

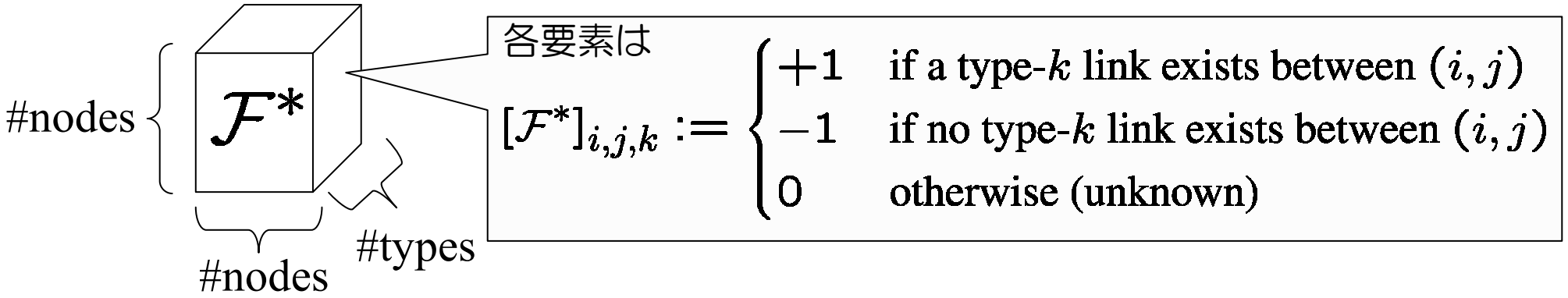
- ネットワーク構造の既知部分が、隣接行列 F^* で与えられている
  - 既知部分は +1 か -1 で、未知部分は 0
- 目標：未知部分 (0) を [-1, +1] の値で（リンクの確信度に応じて）埋める

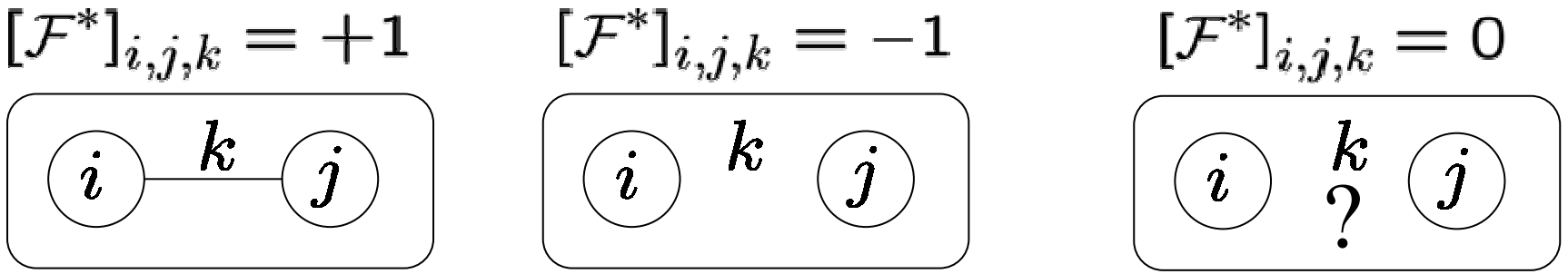




複数タイプリンク予測問題は、3階のテンソルの補完問題として捉えられます

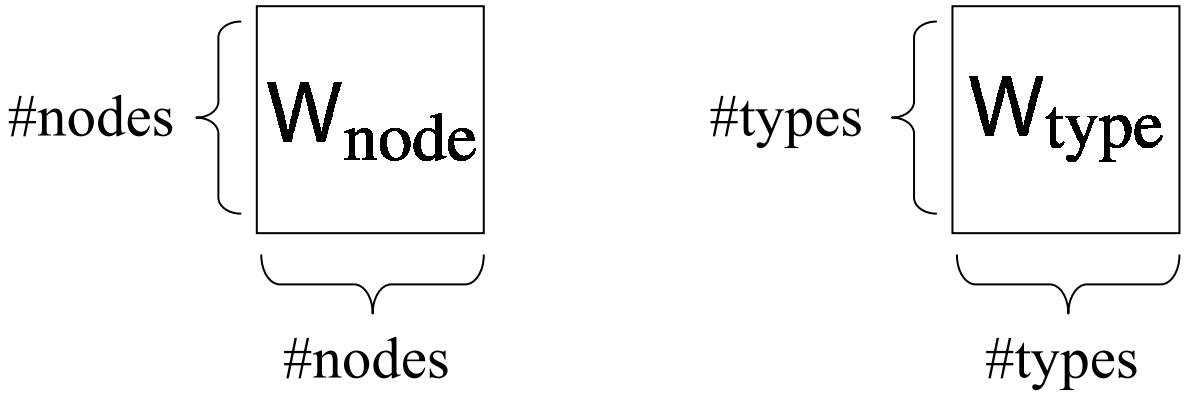
- ネットワーク構造の既知部分が、3階のテンソル $\mathcal{F}^*$ で与えられる
  - 既知部分は +1 か -1 で、未知部分は 0
- 目標：未知部分 (0) を [-1, +1] の値で（リンクの確信度に応じて）埋める





仮定：今回は、補助情報として、ノード間の類似度や、リンクタイプ間の類似度が与えられているとします

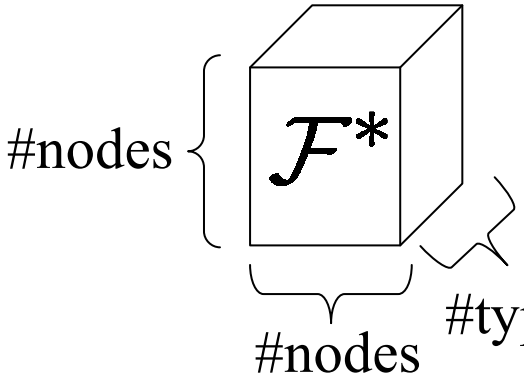
- リンクの有無の情報のほかに、事前知識としての類似度が与えられている場合がしばしばある
- 今回は、ノード間の類似度行列 と リンクタイプ間の類似度行列が与えられているとする
  - カーネル関数に相当



- たとえば、バイオインフォマティクスでは、
  - W_{node} は、タンパク質の配列、遺伝子発現、系統発生、局在部位、などの類似度を
  - W_{type} は、2つのリンクタイプの共起度合い

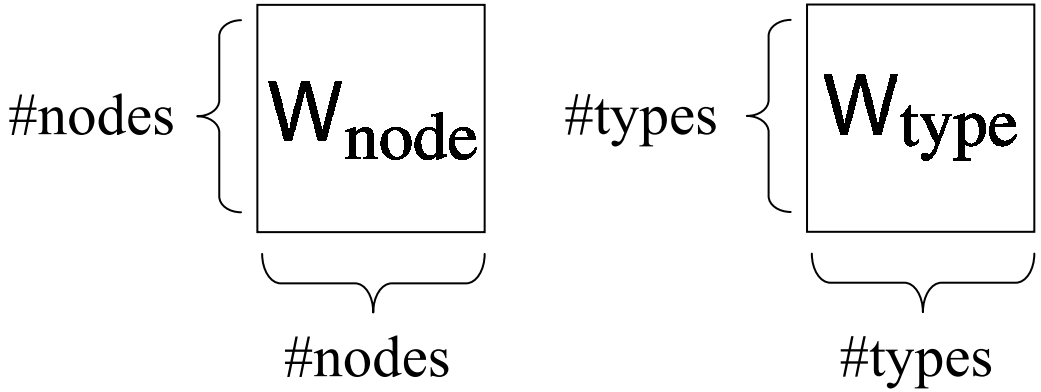
問題のまとめ：複数タイプリンク予測問題

- 入力:
  - ネットワークの既知部分を表した3階テンソル

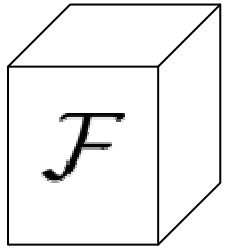


$$[\mathcal{F}^*]_{i,j,k} := \begin{cases} +1 & \text{if a type-}k\text{ link exists between } (i,j) \\ -1 & \text{if no type-}k\text{ link exists between } (i,j) \\ 0 & \text{otherwise (unknown)} \end{cases}$$

  - ノード間、タイプ間の類似度行列

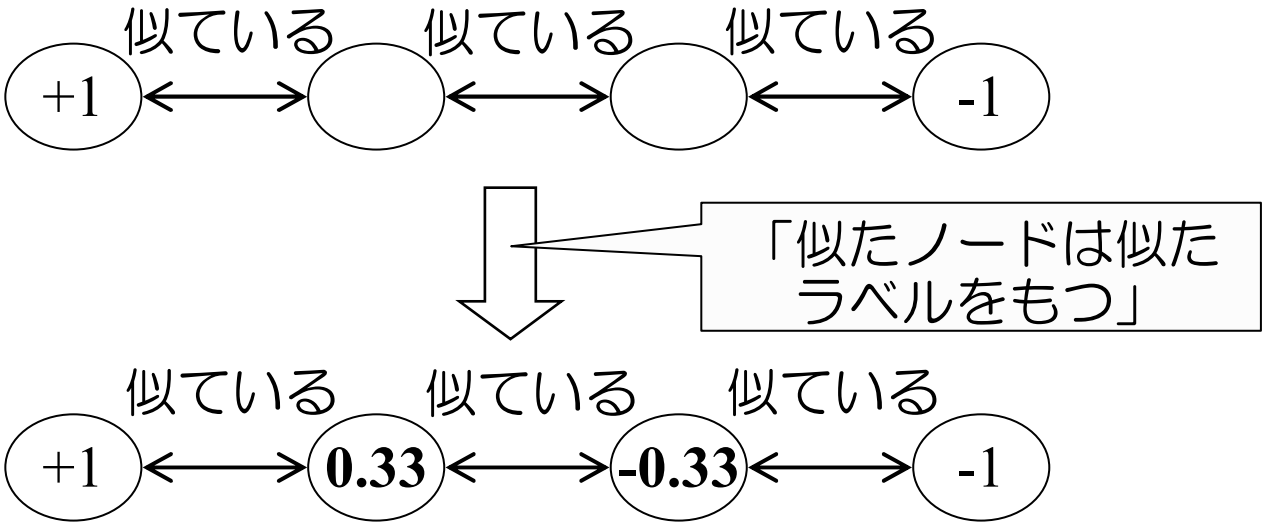


- 出力: 構造が未知の部分のリンク強度を表した3階テンソル



この問題は、半教師つきの予測問題なので、半教師学習の代表的手法「ラベル伝播」を用いることにします

- この問題は、リンク未知のノードペアが予めわかっているので、「半教師つき」の問題として捉えられる
  - テストデータの情報を利用できる
- 半教師つき学習の代表的手法「ラベル伝播」を用いる
  - 元々は、ノードの分類（ノードペアではなく）のための手法
  - 「似たノードは、似たラベルをもつ」の法則を用いる

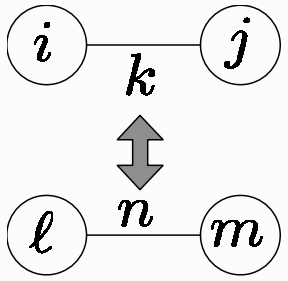


目的関数の定義: ラベル伝播を、ノード「ペア」に適用します

- ラベル伝播を 3つ組み (i, j, k) に対して適用（ノードペアとリンクタイプ）
- 目的関数 $J(\mathcal{F})$ を最小化する、3階テンソル $\mathcal{F}$ を求める
  - 1項目：「似ている3つ組みは、近いリンク強度をもつ」（リンク伝播）
    - $\tilde{w}_{ijk,\ell mn} > 0$ ：2つの 3つ組の間の類似度
  - 2項目：「リンク強度の予測値は、既知の構造に近づける」

$$J(\mathcal{F}) := \frac{\sigma}{2} \sum_{i,j,k,\ell,m,n} \tilde{w}_{ijk,\ell mn}([\mathcal{F}]_{i,j,k} - [\mathcal{F}]_{\ell,m,n})^2 + \frac{1}{2} \sum_{(i,j,k)} ([\mathcal{F}]_{i,j,k} - [\mathcal{F}^*]_{i,j,k})^2$$

2つの3つ組み (i,j,k) と (l,m,n) の間の類似度 (後で定義する)

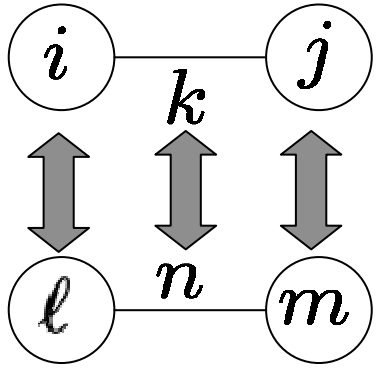


3つ組同士の類似度を、与えられた類似度行列の「クロネッカー積」か「クロネッカー和」によって定義することにします

- 3つ組み同士の類似度を、もともとの類似度行列から定義

1. クロネッカー積類似度 $\tilde{\mathsf{W}} := \mathsf{W}_{\text{type}} \otimes \mathsf{W}_{\text{node}} \otimes \mathsf{W}_{\text{node}}$
   - 「対応するノード同士が全て似ているなら、3つ組み同士も似ている」

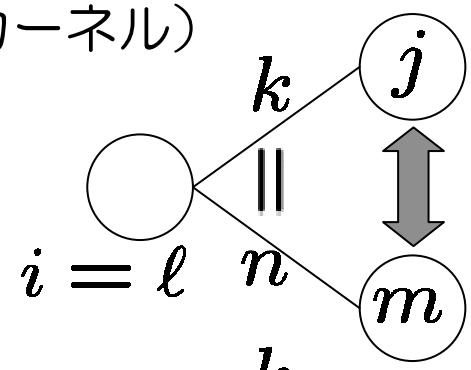
$$\tilde{w}_{ijk,\ell mn} := [\mathsf{W}_{\text{node}}]_{i,\ell} \cdot [\mathsf{W}_{\text{node}}]_{j,m} \cdot [\mathsf{W}_{\text{type}}]_{k,n}$$

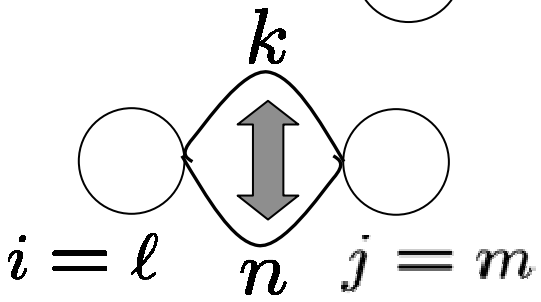


   - 特徴空間のクロネッカー積における内積（ペアワイズカーネル）

2. クロネッカー和類似度 $\tilde{\mathsf{W}} := \mathsf{W}_{\text{type}} \oplus \mathsf{W}_{\text{node}} \oplus \mathsf{W}_{\text{node}}$
   - 「2つが共通で、もう1つが似ているなら、3つ組み同士も似ている」

$$\begin{aligned}\tilde{w}_{ijk,\ell mn} := \quad & [\mathsf{W}_{\text{node}}]_{i,\ell} \cdot \delta(j=m) \cdot \delta(k=n) \\ & + \delta(i=\ell) \cdot [\mathsf{W}_{\text{node}}]_{j,m} \cdot \delta(k=n) \\ & + \delta(i=\ell) \cdot \delta(j=m) \cdot [\mathsf{W}_{\text{type}}]_{k,n}\end{aligned}$$





ラベル伝播を、単に3つ組みに対して適用するだけでは計算量的な問題があります

- 目的関数を行列を使って書き直すと

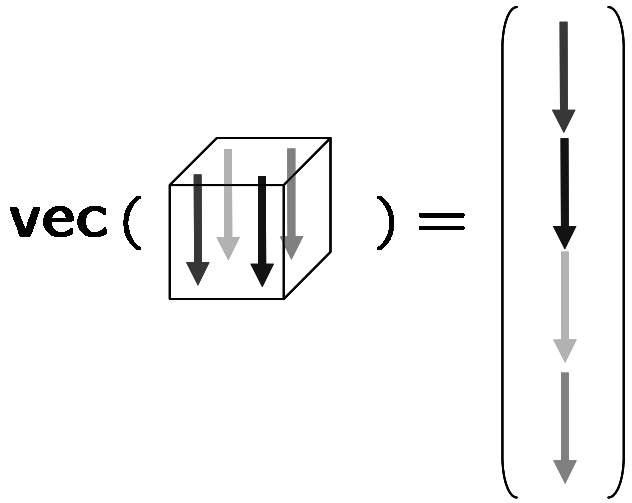
$$J(\mathcal{F}) = \frac{\sigma}{2}\mathbf{vec}(\mathcal{F})^\top \tilde{\mathbb{L}}\mathbf{vec}(\mathcal{F}) + \frac{1}{2} \parallel \mathbf{vec}(\mathcal{F}) - \mathbf{vec}(\mathcal{F}^*) \parallel_2^2$$

  - $\tilde{\mathbb{L}}$ はラプラシアン行列 $\tilde{\mathbb{L}} := \tilde{\mathbb{D}} - \tilde{\mathbb{W}}$ ← $\tilde{w}_{ijk,\ell mn}$ の行列表現

- 結局、以下の連立方程式を解くことによって解が求まる

$$\left(\sigma\tilde{\mathbb{L}} + \mathbb{I}\right)\mathbf{vec}(\mathcal{F}) = \mathbf{vec}(\mathcal{F}^*)$$

(#nodes^2·# types)×(#nodes^2·# types)の巨大な行列



- しかし、連立方程式が大きすぎる...

ひとまず、共役勾配法を適用してみるものの、計算のボトルネックがあります

- テンソル版の共役勾配法
  - 置き換え：$\mathsf{A} = (\sigma\tilde{\mathsf{L}} + \mathsf{I}),\ \mathbf{f} = \mathbf{vec}\,(\mathcal{F})$ and $\mathbf{f}^* = \mathbf{vec}\,(\mathcal{F}^*)$
- $(\sigma\tilde{\mathsf{L}} + \mathsf{I})$ が大きいため、$\mathbf{vec}(\mathrm{Q}(t))$ の計算部分がボトルネック

Conjugate gradient for $\mathsf{A}\mathbf{f} = \mathbf{f}^*$ (standard)

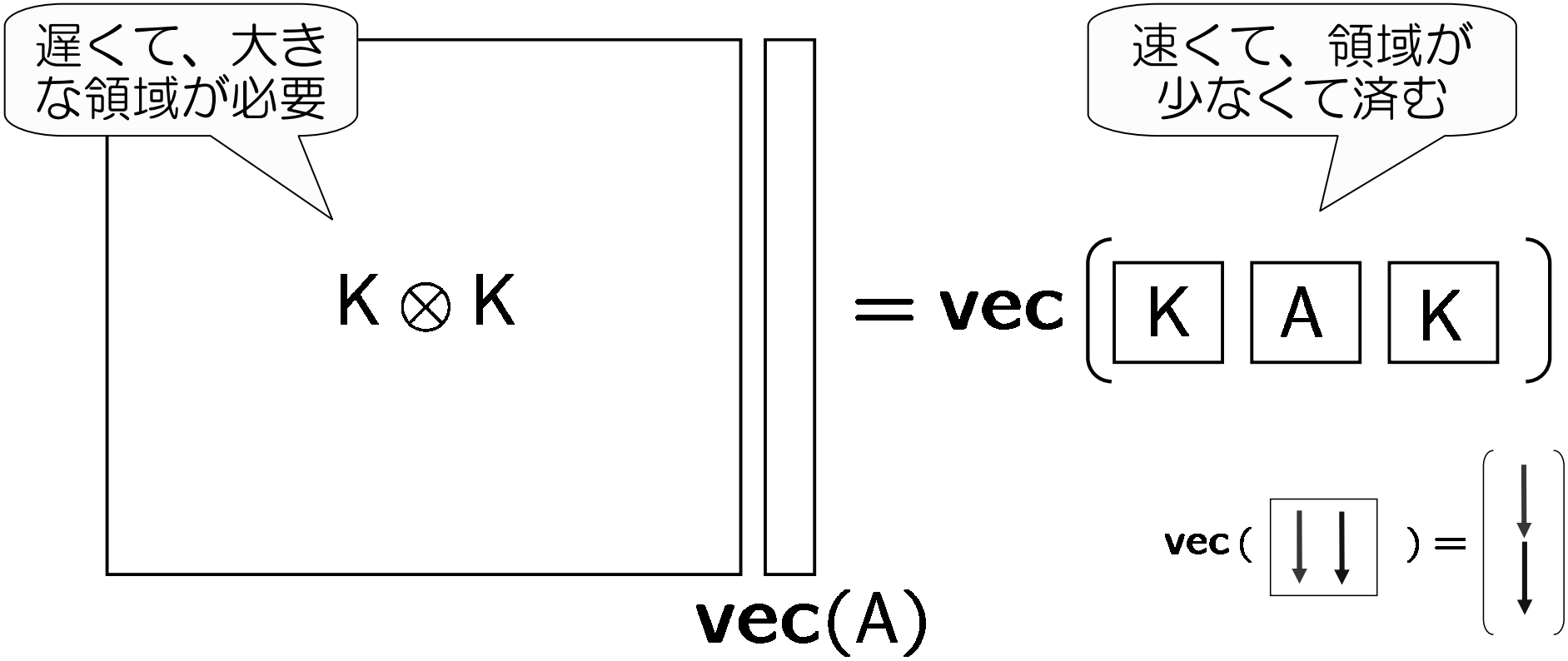
1: $\mathbf{f}(0) := \mathbf{f}^*$
2: $\mathbf{q}(0) := \mathsf{A}\mathbf{f}(0)$
3: $\mathbf{r}(0) := \mathbf{f}^* - \mathbf{q}(0)$, and $\mathbf{p}(0) := \mathbf{r}(0)$
4: **for** $t = 0, 1, 2, \ldots$ **do**
5: $\quad \mathbf{q}(t) := \mathsf{A}\mathbf{p}(t)$
6: $\quad \alpha(t) := \dfrac{\langle \mathbf{r}(t), \mathbf{p}(t) \rangle}{\langle \mathbf{p}(t), \mathbf{q}(t) \rangle}$
7: $\quad \mathbf{f}(t+1) := \mathbf{f}(t) + \alpha(t)\mathbf{p}(t)$
8: $\quad \mathbf{r}(t+1) := \mathbf{r}(t) - \alpha(t)\mathbf{q}(t)$
9: $\quad \beta(t) := \dfrac{\|\mathbf{r}(t+1)\|_2^2}{\|\mathbf{r}(t)\|_2^2}$
10: $\quad$ **if** $\dfrac{\|\mathbf{r}(t+1)\|_2}{\|\mathbf{r}(0)\|_2} < \epsilon$, **return** $\mathbf{f}(t+1)$
11: $\quad \mathbf{p}(t+1) := \mathbf{r}(t+1) + \beta(t)\mathbf{p}(t)$
12: **end for**

Conjugate gradient for $\left(\sigma\tilde{\mathsf{L}} + \mathsf{I}\right)\mathbf{vec}\,(\mathcal{F}) = \mathbf{vec}\,(\mathcal{F}^*)$ (tensorized)

1: $\mathcal{F}(0) := \mathcal{F}^*$
2: $\mathbf{vec}\,(\mathcal{Q}(0)) := (\sigma\tilde{\mathsf{L}} + \mathsf{I})\mathbf{vec}\,(\mathcal{F}(0))$
3: $\mathcal{R}(0) := \mathcal{F}^* - \mathcal{Q}(0)$, and $\mathcal{P}(0) := \mathcal{R}(0)$
4: **for** $t = 0, 1, 2, \ldots$ **do**
5: $\quad \mathbf{vec}\,(\mathcal{Q}(t)) := (\sigma\tilde{\mathsf{L}} + \mathsf{I})\mathbf{vec}\,(\mathcal{P}(t))$
6: $\quad \alpha(k) := \dfrac{\langle \mathcal{R}(k), \mathcal{P}(k) \rangle}{\langle \mathcal{P}(k), \mathcal{Q}(t) \rangle}$
7: $\quad \mathcal{F}(k+1) := \mathcal{F}(k) + \alpha(k)\mathcal{P}(k)$
8: $\quad \mathcal{R}(k+1) := \mathcal{R}(k) - \alpha(k)\mathcal{Q}(k)$
9: $\quad$ **if** $\dfrac{\|\mathcal{R}(k+1)\|_2}{\|\mathcal{R}(0)\|_2} < \epsilon$, **return** $\mathcal{F}(k+1)$
10: $\quad$ **if** $\dfrac{\|\mathbf{r}(t+1)\|_2}{\|\mathbf{r}(0)\|_2} < \epsilon$, **return** $\mathbf{f}(t+1)$
11: $\quad \mathcal{P}(k+1) := \mathcal{R}(k+1) + \beta(k)\mathcal{P}(k)$
12: **end for**

ここで、線形代数の標準的な公式が「非常に」役に立ちます

- 「行列のクロネッカー積」と「ベクトル化された行列」の積についての公式
- 左辺より、右辺がずっと効率的に計算できる



テンソルのモード積によって、共役勾配法を高速化できます

- 計算したいのは、$\mathbf{vec}(\mathcal{Q}(t)) := (\sigma\tilde{\mathsf{L}} + \mathsf{I})\mathbf{vec}(\mathcal{P}(t))$　、ただし
  - クロネッカー積類似度を使う場合

$$\tilde{\mathsf{L}} = \mathsf{D}_{\text{type}} \otimes \mathsf{D}_{\text{node}} \otimes \mathsf{D}_{\text{node}} - \mathsf{W}_{\text{type}} \otimes \mathsf{W}_{\text{node}} \otimes \mathsf{W}_{\text{node}}$$

  - クロネッカー和類似度を使う場合

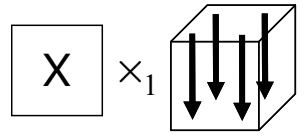
$$\begin{aligned}\tilde{\mathsf{L}} &= \mathsf{D}_{\text{type}} \oplus \mathsf{D}_{\text{node}} \oplus \mathsf{D}_{\text{node}} - \mathsf{W}_{\text{type}} \oplus \mathsf{W}_{\text{node}} \oplus \mathsf{W}_{\text{node}} \\ &= \mathsf{L}_{\text{type}} \oplus \mathsf{L}_{\text{node}} \oplus \mathsf{L}_{\text{node}}\end{aligned}$$

- 前述の公式（のテンソル版）

Tensor multiplication $\times_k$ X multiplies the k-th fiber with a matrix X

$$(\mathsf{X}\otimes\mathsf{Y}\otimes\mathsf{Z})\mathbf{vec}(\mathcal{B}) = \mathbf{vec}(\mathcal{B} \times_1 \mathsf{Z} \times_2 \mathsf{Y} \times_3 \mathsf{X})$$

$$(\mathsf{X}\oplus\mathsf{Y}\oplus\mathsf{Z})\mathbf{vec}(\mathcal{B}) = \mathbf{vec}(\mathcal{B} \times_1 \mathsf{Z} + \mathcal{B} \times_2 \mathsf{Y} + \mathcal{B} \times_3 \mathsf{X})$$



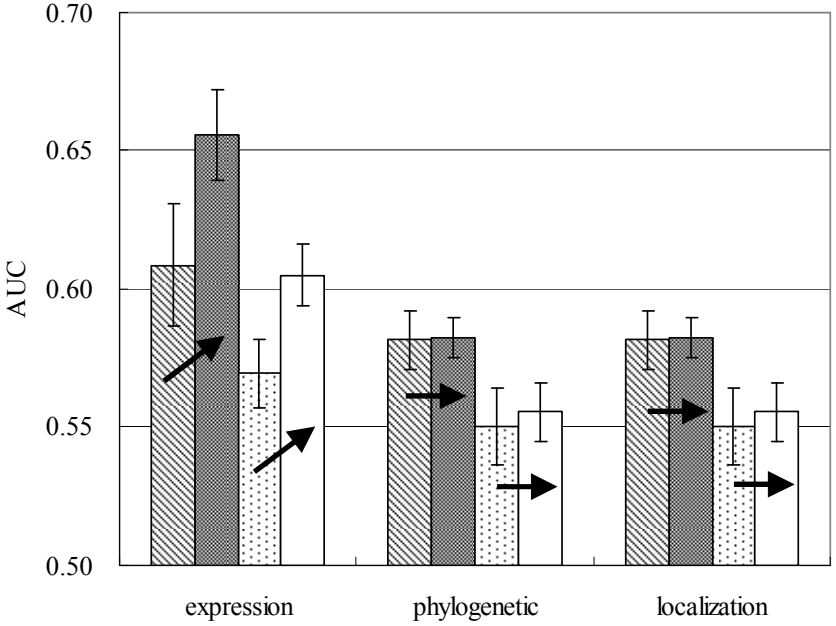
(#nodes^2·# types)×(#nodes^2·# types)の巨大な行列

⇒ #nodes×#nodes×# typesテンソル

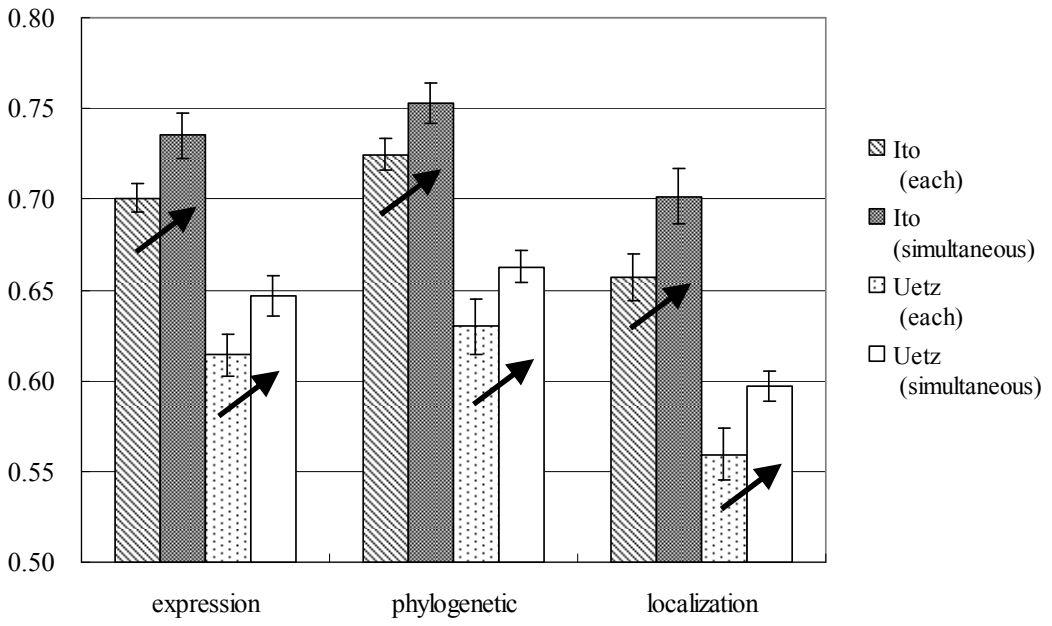
テンソルのモード積で置き換えることで、時間もメモリも大幅に節約できる

2つの生体ネットワークの同時予測では、同時予測によって予測精度が向上しました

- 2つの研究室のタンパク質ネットワークを同時予測
  - ~1,500 ノード、700~900リンク、 ~150リンクを共有
- 同時予測によって性能が向上
- クロネッカー和類似度のほうが全体的に性能がよい



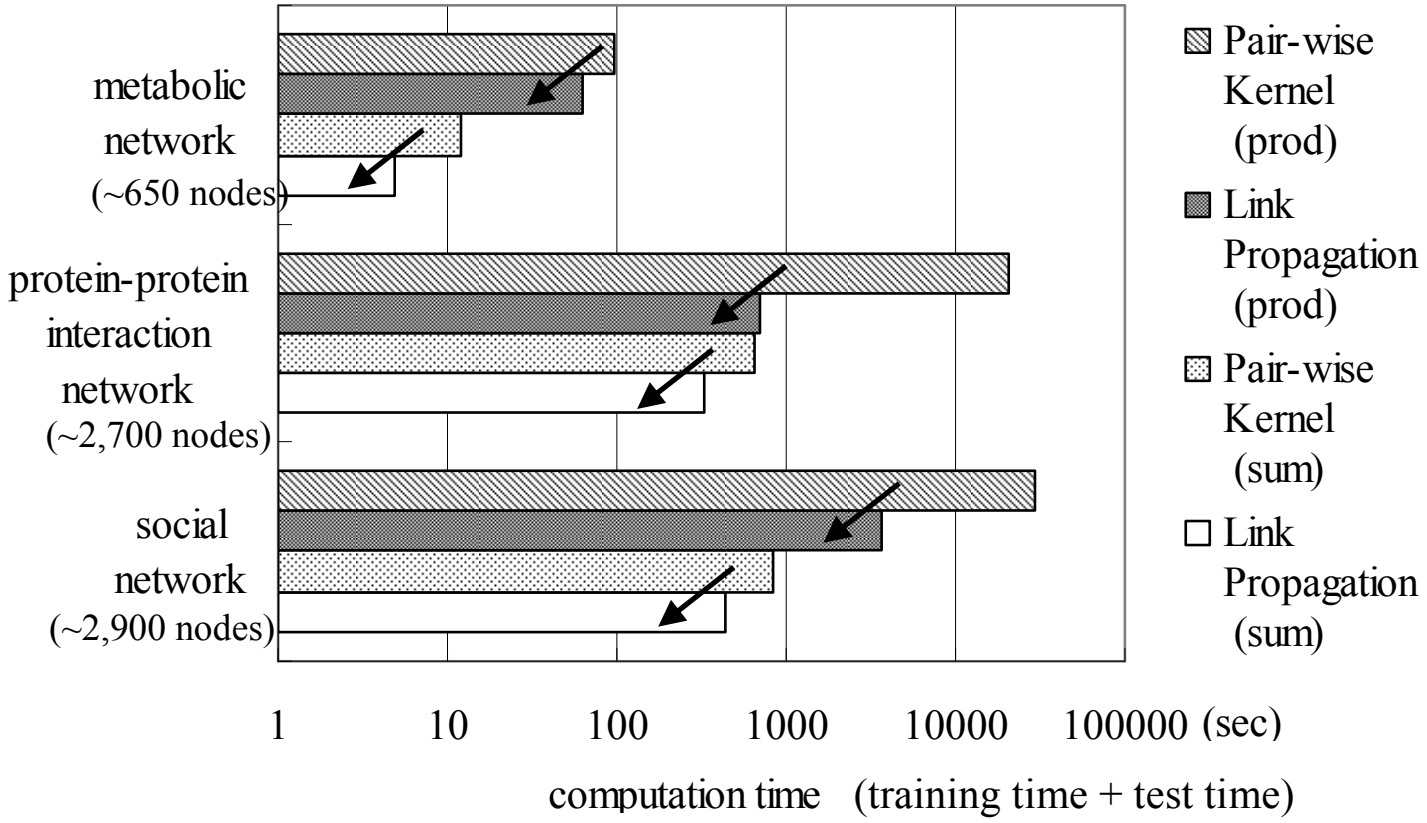
クロネッカー積類似度



クロネッカー和類似度

提案手法は、単一タイプのリンク予測において、既存手法（ペアワイズSVM）よりも遥かに高速です

- 提案手法は、同じ類似度をカーネル関数として用いたオンラインSVM（※）より遥かに高速 （予測精度は、大体同じくらい）
- クロネッカー和のほうがクロネッカー積よりも高速



※ Passive-Aggressiveアルゴリズム（全データ 1周のみでの計測）

まとめ： ネットワークのリンク予測のための、半教師付予測法を提案しました

- 複数種リンクの同時予測： 初めての、複数種類のリンクを扱うことのできるリンク予測法を提案しました
- 半教師付き予測： ラベル伝播法をベースにして、初めての半教師付きリンク予測法を提案しました
- 新しいペアワイズ類似度： ノード類似度のクロネッカー和による、新しいペアワイズ類似度を提案しました
- 効率的な予測アルゴリズム： 共役勾配法を加速することによって、効率的な予測アルゴリズムを提案しました