



IBM Research

ネットワーク構造解析 - 機械学習によるアプローチ -

鹿島 久嗣

IBM東京基礎研究所

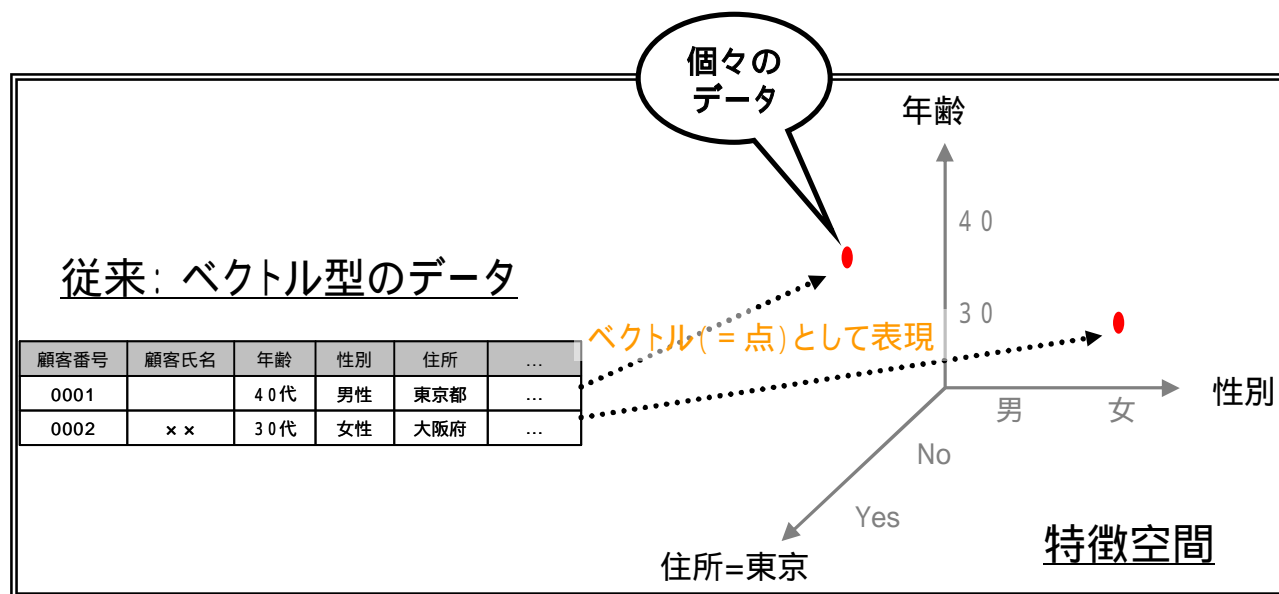
Tokyo Research Laboratory

ネットワーク構造データの解析において、機械学習の立場からどのような考え方に基づいたアプローチが行われているかを紹介します

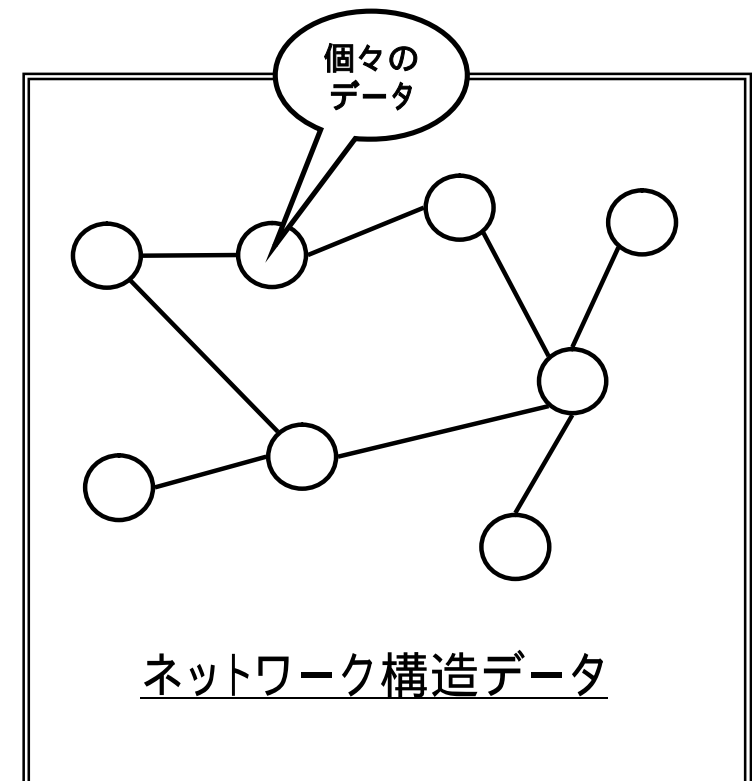
- ここで扱うネットワーク構造データの定義
- なぜ従来の機械学習がネットワーク構造に適用しにくいか
- ネットワーク構造の上でのさまざまな学習タスク
 - ▶ ノードに関するタスク: とくにノード分類
 - ▶ 構造に関するタスク: とくにリンク予測
- 問題へのアプローチに共通の考え方
 - ▶ 具体的な手法
- 統一的なモデル

ネットワーク構造データとは、 データ間の関係がグラフ構造によって表されたものです

- 通常の機械学習において、データが「特徴空間」のベクトル(=点)として表現されるのに対し、ネットワーク構造では「データ間の関係」がグラフ構造で表現されます
- ノードは、個々のデータを表す
- リンクは、データ間の関係を表す



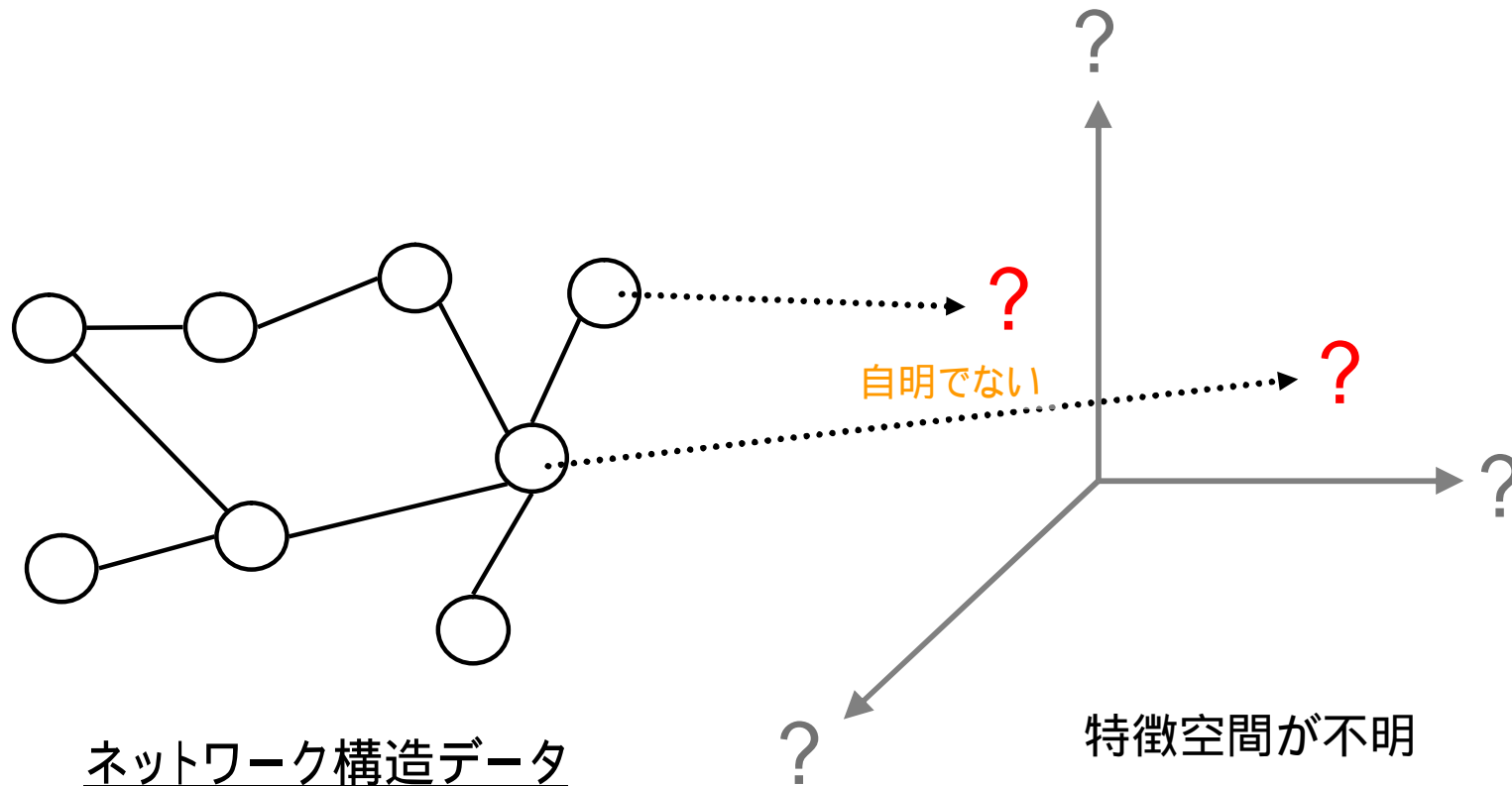
もともとテーブル形式なので、
特徴空間が比較的簡単に構成できる



特徴空間が不明

通常の機械学習手法では、ネットワーク構造を自明に扱うことはできなかつたため、研究が盛んに行われています

- ネットワーク構造の場合、個々のデータに対してどのような特徴空間を定義するかが自明ではない
- 従って、通常の機械学習手法では、ネットワーク構造を自明に扱うことはできない
 - ▶ 通常の機械学習手法は、データ = ベクトルを想定しているため



世の中の色々なデータがネットワーク構造をもっています

- ノードがネットワークの構成要素を、リンクがそれらの間の関係を表します

ネットワーク構造	ノード	リンク
WWW	ページ	ハイパーリンク
SNS	人 コミュニティ	友人関係 所属
生体ネットワーク	遺伝子 タンパク質	制御関係 相互作用関係

- リンクは静的な関係だけでなく、
 - ▶ メールのやりとり
 - ▶ 協力などの動的な関係も表すことがあります

機械学習 / データマイニングの分野では、
ネットワーク構造の解析は「リンク マイニング」という名前で研究が行われています

▪ **Getoor** らによる、リンク マイニングの基本的なタスク

- ▶ ノードに関するタスク
 - ノードのランキング
 - ノードの分類
 - ノードのクラスタリング

- ▶ ネットワーク構造に関するタスク
 - リンク予測
 - 構造パターン発見

今回は、とくに「ノードの分類予測」と「リンク予測」についてのお話です

▪ Getoor らによる、リンクマイニングの基本的なタスク

- ▶ ノードに関するタスク
 - ノードのランキング
 - **ノードの分類**
 - ノードのクラスタリング
- ▶ ネットワーク構造に関するタスク
 - **リンク予測**
 - 構造パターン発見

どちらも「半教師付き分類学習」のような問題として捉えられます

ノードに関する予測タスク: ノードの分類問題

教師付きノード分類タスクの定義:

グラフ上のいくつかのノードのクラスラベルをもとに、残りのノードのクラスラベルを当てる

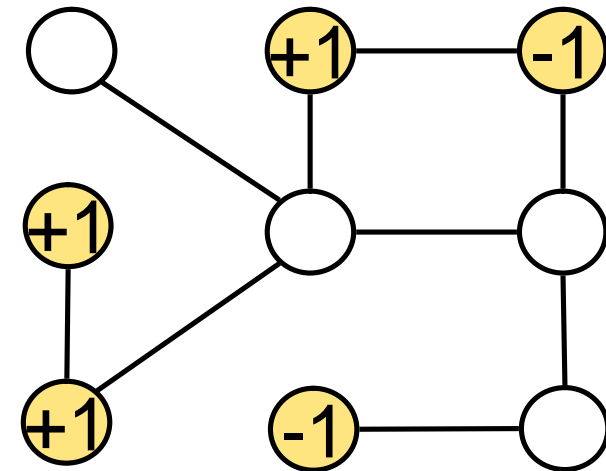
▪ やりたいこと: グラフ構造が与えられ、その上で

- ▶ ラベル付きのノード ●
- ▶ ラベルなしのノード ○

が与えられたとき

- ▶ ラベルなしのノード ○

のラベル(「+1」か「-1」)を当てたい



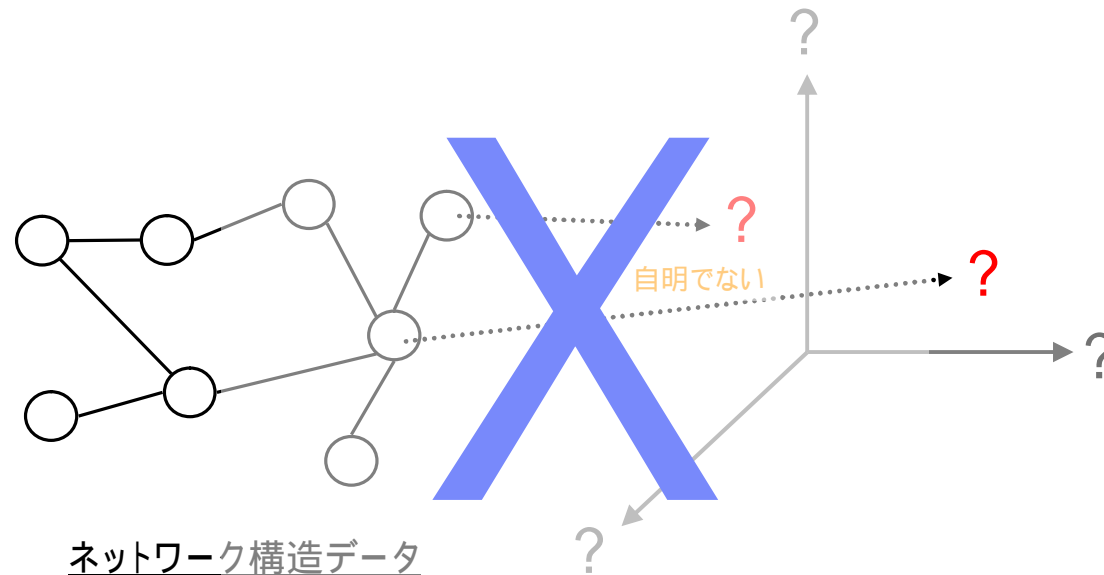
▪ 現実の問題例としては、

- ▶ タンパク質の相互作用ネットワークにおける、各タンパク質の機能予測
- ▶ WWWの各HTMLページのトピック予測

▪ 半教師付きの分類問題、あるいは、トランスダクションの問題として捉えることができる

教師付きノード分類タスクでは、特徴空間を構成するのではなく、直接的に構造情報を利用して問題を解く方法を用います

- 1つ1つのノードを特徴空間の点として表現できれば従来の機械学習アルゴリズムが適用できるが、特徴空間の構成は自明ではない
- ネットワーク構造を「世界の構造の表現」と受け入れ、直接的に構造情報を利用する方法を考える



ネットワーク構造データ

特徴空間を使わず
直接解く

教師付きノード分類の各手法に共通する考え方は「隣同士は似ている」です

- グラフ構造が、データ空間の構造を表しているとする
「隣同士のノードのクラスラベルは似ている」と考えるのは自然
 - ▶ 周りを「+ 1」で囲まれてるノードは、やっぱり「+ 1」だろう…

- さらに、「伝播」を考えることで、隣よりも遠くに「似てる」が伝わる
 - ▶ AさんとBさんが似ている
 - ▶ BさんとCさんが似ている
 - AさんとCさんがそこそこ似ている

具体的な解法はだいたい、
「直接解く」か「グラフの上での類似度を定義して解く」のどちらかです

1. 直接解く

- ▶ 「隣同士のクラスラベルは似ている」を式で書いて、最適化問題を解くことで未知ラベルを当てる

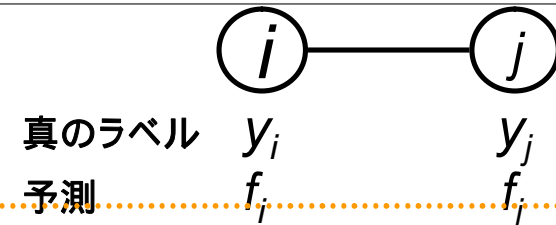
2. グラフの上での類似度を定義して学習アルゴリズムに渡す

- ▶ 「類似度の伝播」に従って、2つのノードの類似度が定義できれば、それをカーネル法などの「類似度ベース学習器」に放り込めばOK

解法 1) 「直接解く」方法: ティコノフ正則化によって 「隣同士のクラスラベルは似てる」を最適化問題として定式化し、解きます

■ 予測が満たすべき2つの条件を式で書いてみる

- ノード i のラベル予測を $\text{sign } f_i$ とする (\mathbf{f} : f_i を並べたベクトル)
- ノード i のラベルを y_i とする (\mathbf{y} : y_i を並べたベクトル)



▶ 条件 1) 隣同士のクラスラベルは似ている

- $F_1 = \sum_{i,j} (\text{ノード } i \text{ のラベル予測} - \text{ノード } j \text{ のラベル予測})^2$ を最小化

キモはここ

$$F_1 = \sum_{i,j} (f_i - f_j)^2 = \mathbf{f}^T L \mathbf{f}$$

Lはグラフラプラシアン

▶ 条件 2) ラベル付きのノードに対する予測が当たる

- $F_2 = \sum_{i:\text{ラベル付きノード}} (\text{ノード } i \text{ のラベル予測} - \text{ノード } i \text{ のラベル})^2$ を最小化

$$F_2 = \sum_{i:\text{labeled data}} (f_i - y_i)^2 = (\mathbf{f} - \mathbf{y})^T I_l (\mathbf{f} - \mathbf{y})$$

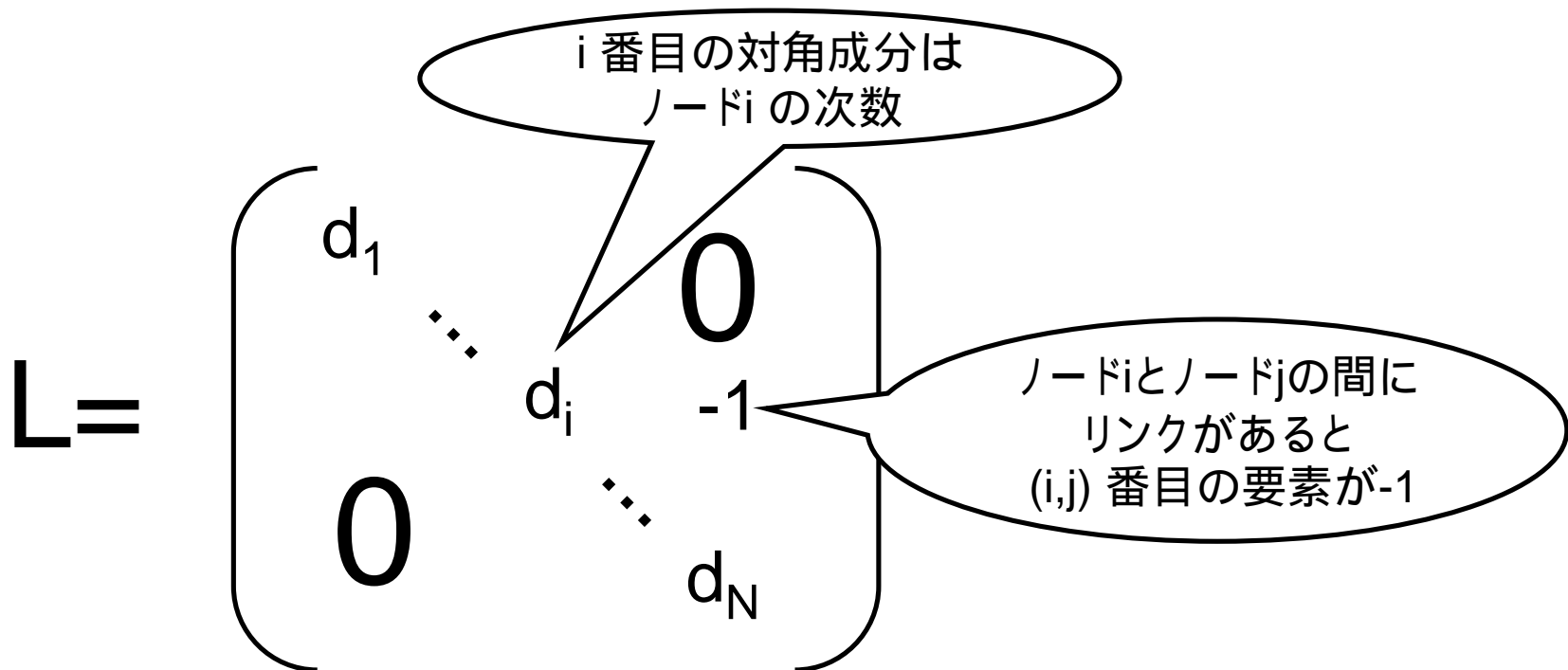
■ F_1 と F_2 の線形結合を目的関数として最小化することで予測が求まる

- ▶ $F_1 + C F_2$ を最小化する (C は定数)
- ▶ 凸で制約ナシ

- ▶ この場合、closed form の解が求まる $\mathbf{f} = (L + C I_l)^{-1} I_l \mathbf{y}$

参考: グラフラプラシアンLは、グラフの構造を行列で表現したものです

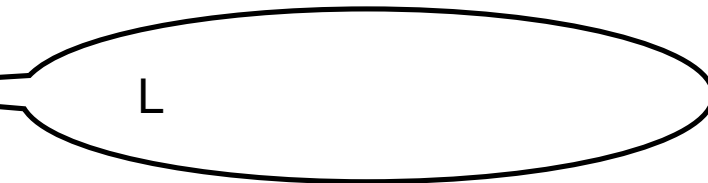
- ノードの次数をもつ対角行列から、隣接行列を引いたもの
 - ▶ 対角成分は、ノードの次数を表す
 - ▶ 対角以外の成分は、グラフのリンク情報を表す(リンクがあるときに-1)
 - ▶ 対称行列



- これをつかって $F_1 = \sum_{i,j} (f_i - f_j)^2 = \mathbf{f}^\top L \mathbf{f}$ と書ける

解法 2) 「グラフの上での類似度を定義して解く」方法: 拡散カーネルによって隣よりも離れたノード間の類似度を求め 類似度ベースの学習機に渡します

- 類似度ベースの機械学習法 (= SVMなどのカーネル法など) に渡すために、グラフの上での2ノード間の類似度を定義する方向で考える
- 「隣同士の類似度を1、そうでないものの類似度を0」では、心許ない
「隣の隣」、「隣の隣の隣」、「隣の隣の隣の隣」、... も考えよう

- そこで、拡散カーネル $K = \exp(-tL)$


▶ (i,j) 要素 k_{ij} は、ノード i と ノード j とのカーネル (= 類似度)

▶ t は拡散の時間を表すパラメータ

- 大きいほど、長い時間「類似度」が拡散することになる

▶
$$K = \exp(-tL) = I + (-tL) + \frac{(-tL)^2}{2!} + \frac{(-tL)^3}{3!} + \dots$$

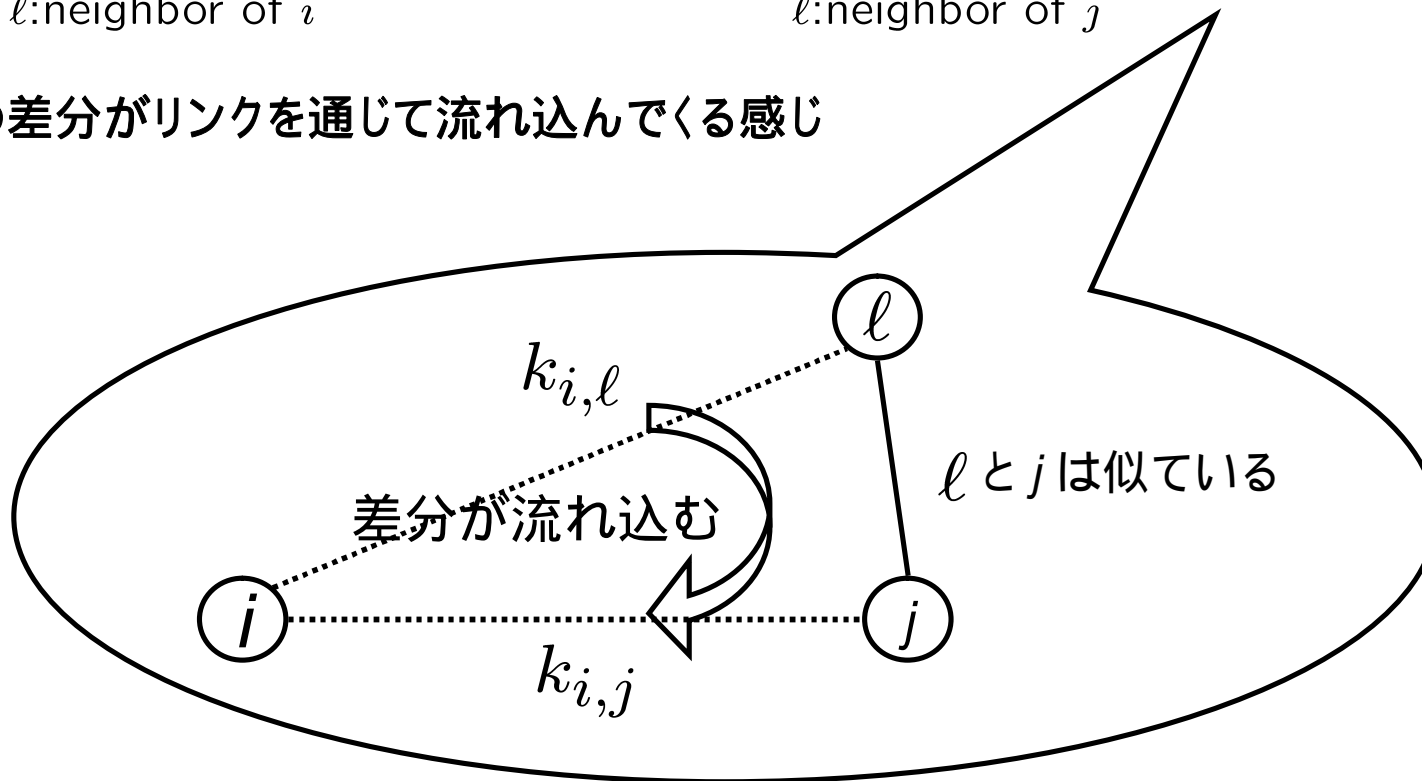
であるから、「隣」、「隣の隣」、「隣の隣の隣」、「隣の隣の隣の隣」、...を重みを小さくしながら足しこんでいっているイメージ

拡散カーネルの解釈：類似度がリンクを介して流れると考えます

- 拡散カーネルの定義 $K = \exp(-tL)$
 - ▶ (i,j) 要素 k_{ij} は、ノード i と ノード j とのカーネル (= 類似度)
- 拡散カーネルは以下の微分方程式の解になっている

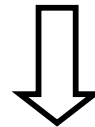
$$\frac{\partial k_{i,j}(t)}{\partial t} = \frac{1}{2} \left(\sum_{\ell: \text{neighbor of } i} (k_{\ell,j}(t) - k_{i,j}(t)) + \sum_{\ell: \text{neighbor of } j} (k_{i,\ell}(t) - k_{i,j}(t)) \right)$$

類似度の差分がリンクを通じて流れ込んでくる感じ



現実的な問題設定では、 構造の情報だけでなく、個々のノードの情報も利用できます

- ここまでは、ネットワーク構造情報のみを用いていたが、本当はノードの持つ情報も使える
 - ▶ タンパク質ネットワークの場合、個々のタンパク質はそれぞれ配列情報などの情報を持っている
 - ▶ SNSの場合、個々の人はそれぞれの個人情報を持っている
- ノード情報は、ベクトル表現可能であるから、通常の機械学習手法で扱える



- 構造情報を用いた「隣は似てる」という考え方に基づく学習と、ノード情報に基づく従来の学習を組み合わせたい

個々のノード情報

+

「隣は似ている」
ティコノフ正則化

+

「類似度を定義する」
拡散カーネル

組み合わせ 1) 個々のノード自身の情報 + 「隣は似ている」ティコノフ正則化: 多様体正則化 (manifold regularization) によって「隣同士の～」制約を入れます

- ノード情報に基づく予測器のパラメータを「隣同士の予測が近くなるような制約」を入れて学習する
- ノード情報に基づく予測器を $\text{sign } f_{\mathbf{w}}(\mathbf{x})$ とする
 - ▶ \mathbf{w} はノード情報に対するパラメータ
 - ▶ \mathbf{x} はノード情報
- 通常は、正しいラベル $y^{(i)}$ を $\text{sign } f_{\mathbf{w}}(\mathbf{x}^{(i)})$ と予測してしまう誤りの損失 $\text{Loss}(y^{(i)}, f_{\mathbf{w}}(\mathbf{x}^{(i)}))$ の和を最小化するようにパラメータを決定する

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \sum_i \text{Loss}(y^{(i)}, f_{\mathbf{w}}(\mathbf{x}^{(i)}))$$

- ここに「隣同士～」の制約をいれると...
 - ▶ グラフラプラシアンをつかった正則化項を入れる

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \sum_i \text{Loss}(y^{(i)}, f_{\mathbf{w}}(\mathbf{x}^{(i)})) + C \mathbf{f}^{\top} L \mathbf{f}$$

$$\sum_{i,j} (f_{\mathbf{w}}(\mathbf{x}^{(i)}) - f_{\mathbf{w}}(\mathbf{x}^{(j)}))^2$$

組み合わせ 2) 個々のノード自身の情報 + 「類似度を定義する」拡散カーネル: カーネルを組み合わせることによって簡単に実現できます

- 拡散カーネルによって、ネットワーク構造から、2つのノード i と j の間のカーネル関数 $K^D(i, j)$ を導くことができた
- 一方、ノード自身の情報から、ノード i と j の間のカーネル関数 $K^N(i, j)$ を定義する
- 従って、2つを組み合わせ、両方の情報をもったカーネル関数を作ることができる
 - ▶ 「カーネル関数の線形和はカーネル関数」なので
$$K^{NEW}(i, j) = \alpha K^D(i, j) + \beta K^N(i, j)$$
によってカーネル関数を合成できる (α と β は定数)
 - ▶ ちなみに「最良の」係数を求める問題は半正定値計画 (SDP) として記述できる

Bach et al.: *Multiple Kernel Learning, Conic Duality, and the SMO Algorithm*, ICML 2004

Lanckriet et al.: *Kernel-Based Data Fusion and Its Application to Protein Function Prediction in Yeast*, PSB 2004

まとめ：ノードの分類問題では 「隣同士は似ている」というネットワーク構造上の制約をうまく使います

- ネットワーク構造上のノード分類問題に対する2つの異なったアプローチを紹介しました
 - ▶ 「直接解く」ティコノフ正則化
 - ▶ 「ノードの類似度を定義する」拡散カーネル

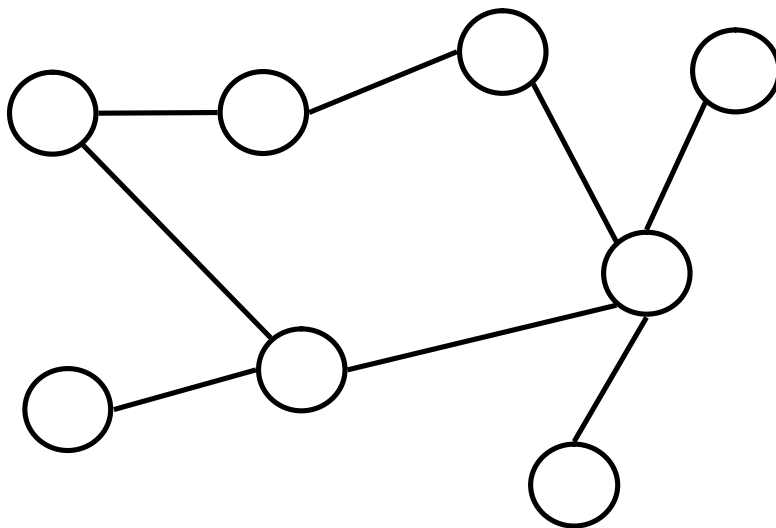
これらを従来のノード情報のみを扱う手法と組み合わせる方法を紹介しました

- ▶ 多様体正則化
 - ▶ カーネル合成
-
- 実際には、色々な手法が山ほど提案されています。
紹介した手法が代表的とは限りませんが、ベースとなる考え方は大体同じです。

構造に関する予測タスク: リンク予測

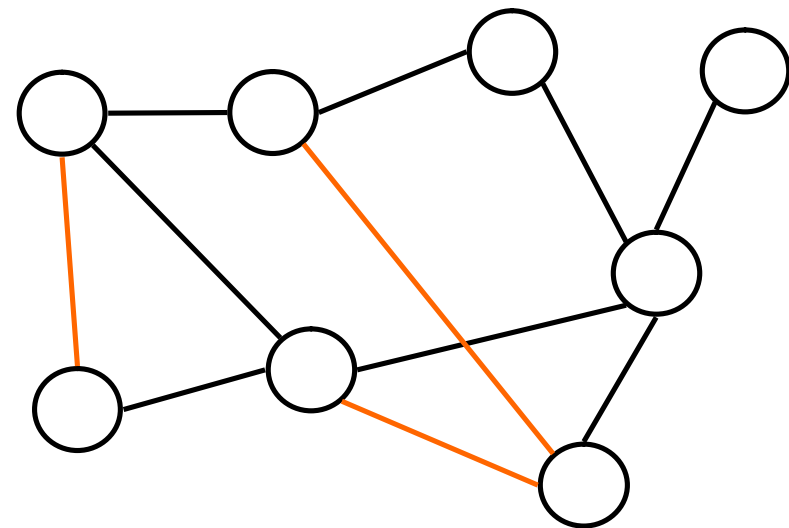
リンク予測問題は、部分的に観測されているネットワーク構造から、残りの構造を推定する問題です

- 例えば、
 - ▶ 生体ネットワークの構造予測
 - ▶ SNSにおける友人の推薦
- 特にバイオインフォマティクス分野で著しく発展している
 - ▶ 理由は、たぶん一番役に立ちそうだから



部分的に観測される構造

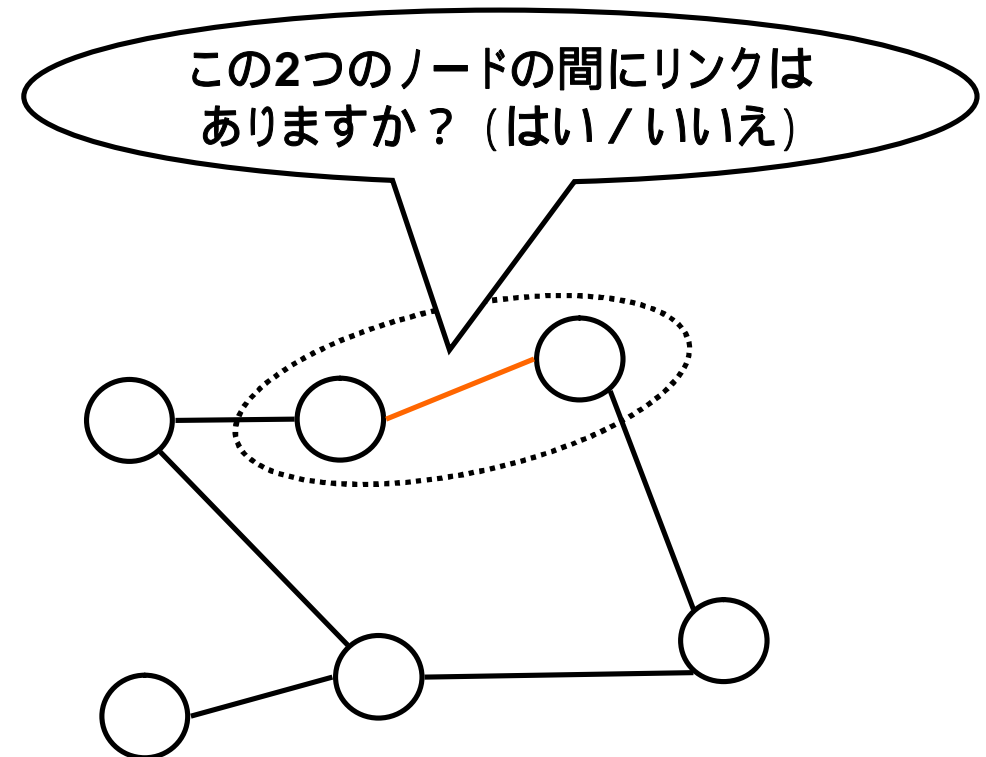
補完
⇒



予測される構造

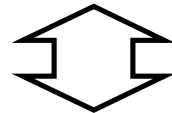
リンク予測の問題は、ノードペアの分類問題として捉えることができます

- リンク予測の問題は、2つのノードに注目して、その間に
 - ▶ リンクがあるかないか (分類問題)
 - ▶ リンクのありそう具合 (ランキング問題)を予測する問題と考えることができる
- リンクの存在はお互いに独立に決まる (i.i.d.) と仮定する



リンク予測は、ノードペアの特徴ベクトルに基づき分類を行います

- 普通の分類は、データ i の特徴ベクトル $\mathbf{x}^{(i)}$ に基づき、データ i の分類を行う



- リンク予測では、ノードペア (i, j) の特徴ベクトル $(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ から、ノードペアの特徴ベクトル $\mathbf{z}^{(i,j)}$ を構成し、これに基づきノードペア (i, j) の分類を行う
- もっとも単純な $\mathbf{z}^{(i,j)}$ の定義の仕方は、要素ごとの和や積によって定義する方法

$$\mathbf{z}^{(i,j)} = \left(x_1^{(i)} \cdot x_1^{(j)}, x_2^{(i)} \cdot x_2^{(j)}, x_3^{(i)} \cdot x_3^{(j)}, \dots \right)$$

でもこれではちょっと不十分。

- ▶ 「片方にある特徴があって、もう片方にそれに対応する別の特徴がある」が表現できないから

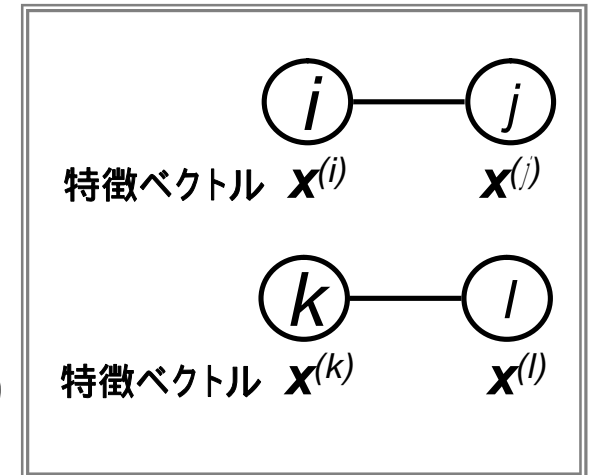
ノードペアの特徴ベクトルは、ノードの特徴ベクトルのテンソル積で定義できます
また、ノードペアの特徴ベクトル同士のカーネル関数も容易に計算できます

- 2つのノードの特徴ベクトルのテンソル積によってノードペアの特徴ベクトルを定義する

$$\mathbf{z}^{(i,j)} = \mathbf{x}^{(i)} \otimes \mathbf{x}^{(j)}$$

ノードペアの
特徴ベクトル

$$= \left(\begin{array}{l} x_1^{(i)} \cdot x_1^{(j)}, x_1^{(i)} \cdot x_2^{(j)}, x_1^{(i)} \cdot x_3^{(j)}, \dots \\ x_2^{(i)} \cdot x_1^{(j)}, x_2^{(i)} \cdot x_2^{(j)}, x_2^{(i)} \cdot x_3^{(j)}, \dots, \\ x_3^{(i)} \cdot x_1^{(j)}, x_3^{(i)} \cdot x_2^{(j)}, x_3^{(i)} \cdot x_3^{(j)}, \dots \end{array} \right)$$



によってノードペアの特徴ベクトルを定義する

- カーネル関数も簡単: 2組のノードペア (i, j) と (k, l) の特徴ベクトル $\mathbf{z}^{(i,j)}$, $\mathbf{z}^{(k,l)}$ 間の内積は、

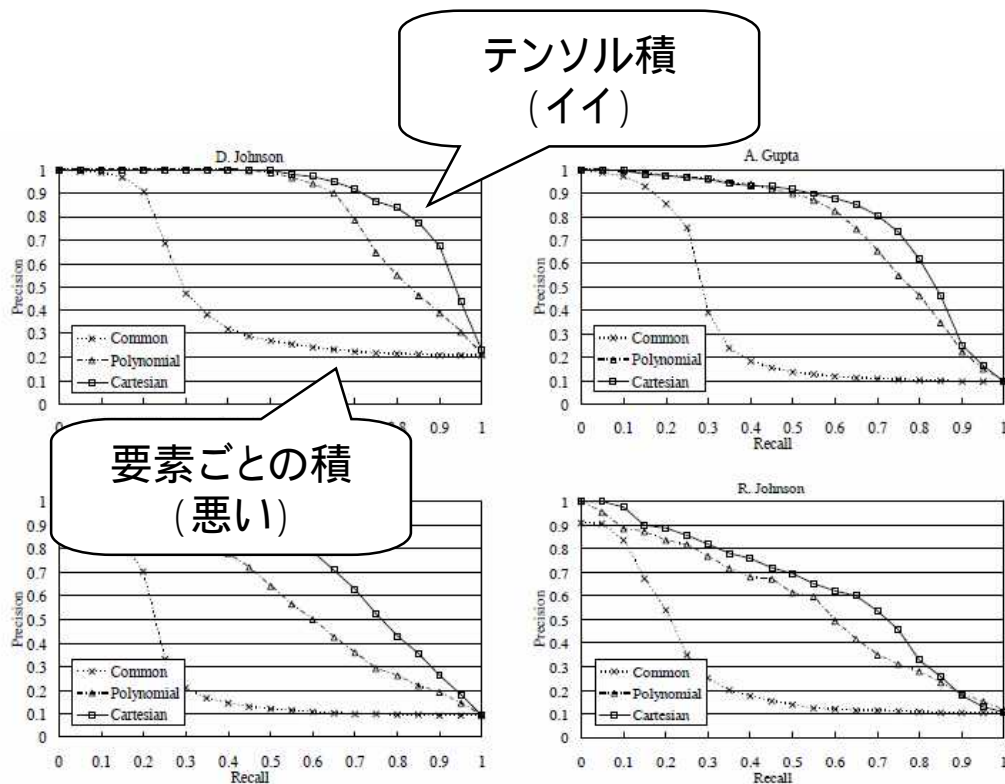
$$\langle \mathbf{z}^{(i,j)}, \mathbf{z}^{(k,l)} \rangle = \langle \mathbf{x}^{(i)}, \mathbf{x}^{(k)} \rangle \cdot \langle \mathbf{x}^{(j)}, \mathbf{x}^{(l)} \rangle$$

のように計算できる

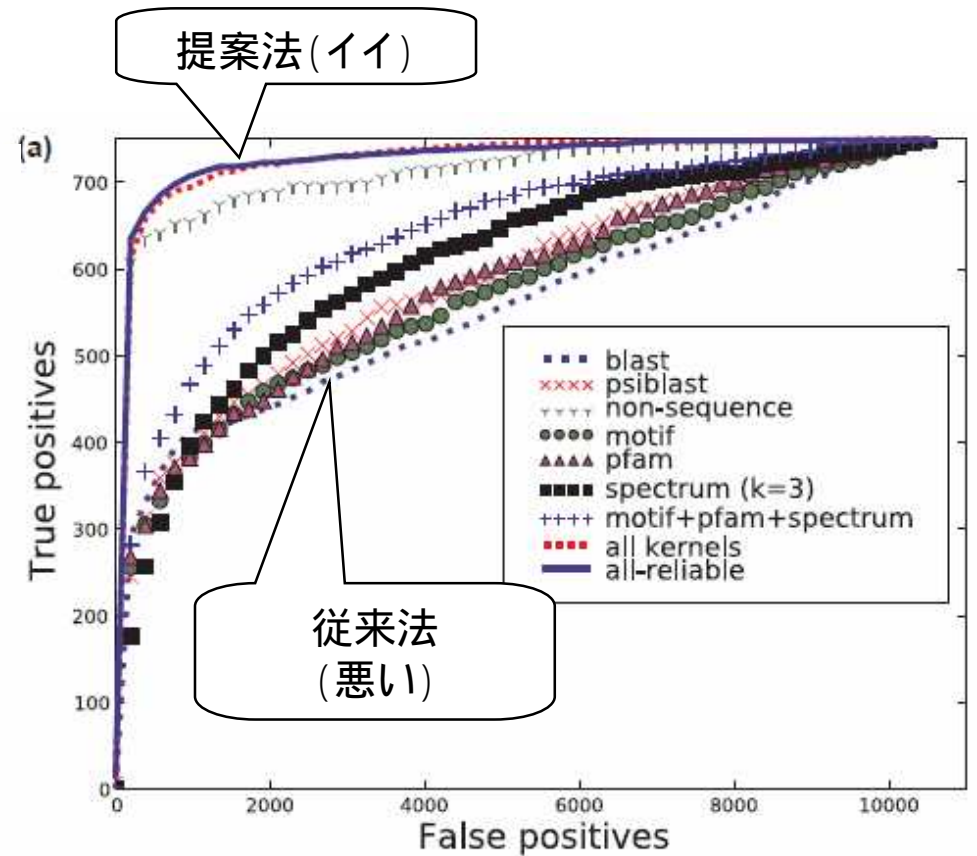
- ノードの特徴ベクトルの次元の計算量で計算できる (ただしノード数の2乗になる)
- ペアの組み合わせに向きがない場合には対称化して使う $\langle \mathbf{z}^{(i,j)}, \mathbf{z}^{(k,l)} \rangle + \langle \mathbf{z}^{(i,k)}, \mathbf{z}^{(j,l)} \rangle$
- あとは SVM などに投げ込めば解決

性能はイイみたいです

- 同一著者の同定
([Oyama & Manning] より拝借)

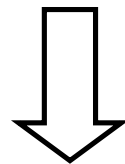


- タンパク質の機能予測
([Ben-Hur & Noble] より拝借)



ノード分類同様、リンク予測においても、ノードのもつ情報と構造の持つ情報(=リンク指標)の2種類を組み合わせて用いることができます

- **ノードの持つ情報**: ノード単体の情報をノードペア情報にまとめ上げて使う (前述)
 - ▶ タンパク質ネットワークの場合、個々のタンパク質はそれぞれ配列情報などの情報を持っている
 - ▶ SNSの場合、個々の人はそれぞれの個人情報を持っている
- **構造の持つ情報 (=リンク指標)**: 大体もともとノードペアに対して定義される
 - ▶ 計量社会学における社会ネットワーク分析の文脈で古くから用いられている各種のリンク指標を用いることができる
 - ▶ 情報検索でも色々提案されている
 - ▶ ノード分類のときに使った指標(例えば、拡散カーネル)を使うのもアリ



2種類をまとめて分類器に渡せば解決

構造的なリンク指標の多くはネットワーク構造の遷移モデルに基づいています

- 社会ネットワーク研究やスケールフリーネットワーク研究においていくつかの指標が提案されている
- これらは、ネットワークの構造遷移モデルに基づいている

- ▶ 共通の隣接ノードが多いほど、リンクが張られやすいとするモデル「友達の友達は友達」

$$\text{common neighbors} := |\Gamma(i) \cap \Gamma(j)|$$

$\Gamma(i)$ はノード i の隣接ノード集合

- ▶ common neighbors の重み付きバージョン「友達が少ない人ほど付き合いは深い」

$$\text{Adamic/Adar} := \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log |\Gamma(k)|}$$

$$\text{Jaccard's coefficient} := \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}$$

(こちらは情報検索で使われる指標)

- ▶ 遠距離の影響もとりにれた common neighbors

$$\text{Katz}_\beta := \sum_{l=1}^{\infty} \beta^l |\text{paths}_{i,j}^{(l)}|$$

$\text{path}_{i,j}^{(l)}$ はノード i から j への長さ l のパスの集合

- これは、ほぼ拡散カーネル

- ▶ バラバシの preferential attachment モデル (友人が多いほど、より多くの友人を得る)

$$\text{preferential attachment} := |\Gamma(i)| \cdot |\Gamma(j)|$$

- 単純に、指標の大きい順にリンクがあると予測する方法もあり

参考：各指標の相対的な関係

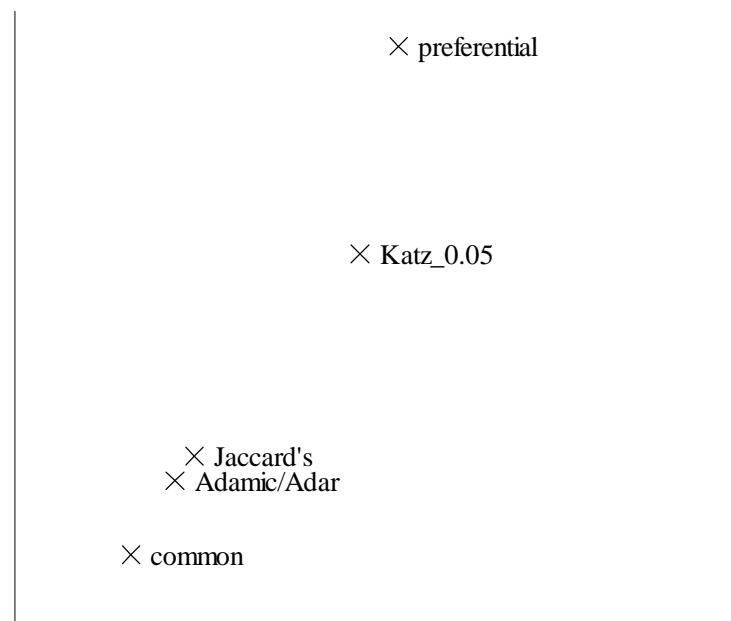
■ 代謝ネットワークデータに対する各手法の予測の Spearman 相関 (順序の相関)

- ▶ 大きいほど予測が似ている

	common	Jaccard's	Adamic/Adar	preferential	Katz _{0.05}
common	1	0.92	0.94	0.31	0.61
Jaccard's	0.92	1	0.97	0.53	0.75
Adamic/Adar	0.94	0.97	1	0.49	0.70
preferential	0.31	0.53	0.49	1	0.84
Katz _{0.05}	0.61	0.75	0.70	0.84	1

- ▶ 相関のMDSによる視覚化

- preferential attachmentと
common neighborsが一番遠い



【手前味噌】「構造情報だけで、できるところまで頑張ろう」

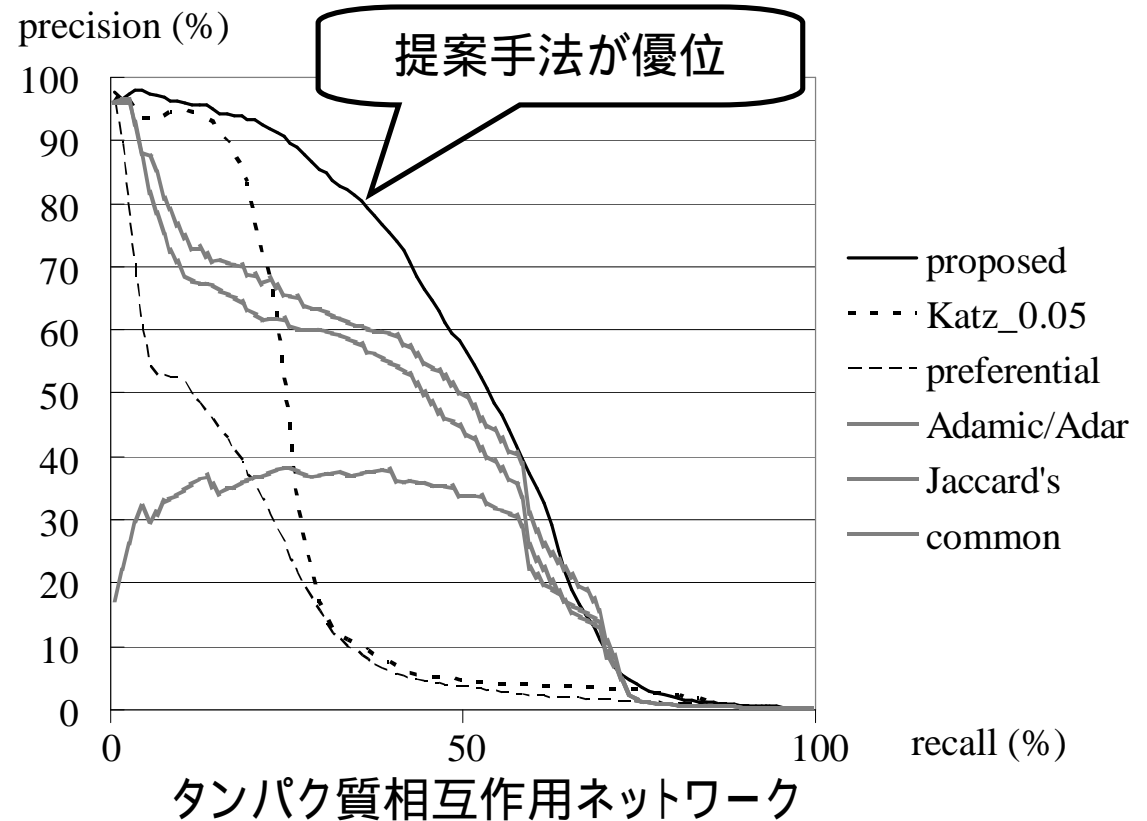
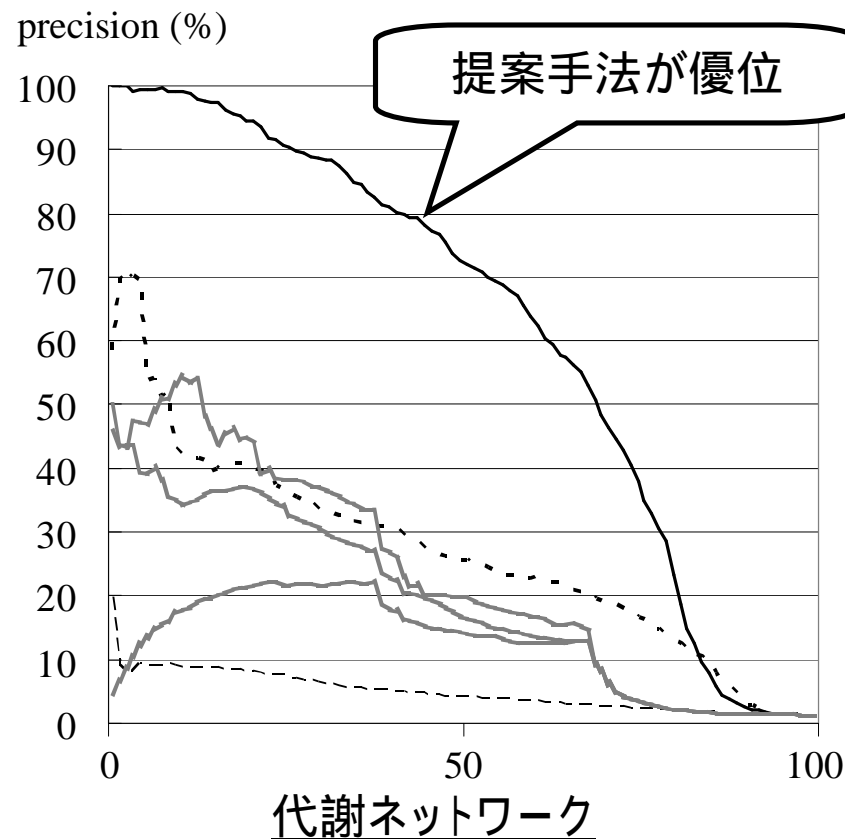
リンク指標をパラメタライズすることで予測精度を高めようという試みを行っています

- 個々のリンク指標に注目すると、これらのパラメタライズされた版を考えることができる
- 例えば、ノードからノードへの枝の「コピー & ペースト」が確率的におこるノードコピーモデル
 - ▶ もともと Kleinberg らによって、Webの構造進化のモデルとして提案された
 - ▶ タンパク質のネットワークの成長過程のモデルとしても有力
- ノード間のコピー確率をモデルパラメータとして、パラメータを学習、残りの部分を予測する
- 観測されるネットワークは、この時間変化モデルの定常状態であると仮定して、分かっている部分を使ってパラメータを最尤推定、そして残りの部分を予測する



リンク指標をパラメタライズすることによって予測精度は向上します

- 提案されているいくつかのリンク指標と比べて性能が改善された
- 同様に、他の指標のパラメタライズも考えられるはず...



まとめ：リンク予測の問題は、ノードペアについての分類問題と考えます

- ノードの特徴から、ノードペアの特徴を構成する方法を紹介しました
- ネットワーク構造に基づくいくつかのリンク指標を紹介しました
 - ▶ 手前味噌として、リンク指標のパラメタライズの試みを紹介しました

統一的フレームワーク？：ネットワーク構造全体のモデル

ネットワーク構造全体の一般的モデルを考えることで、問題をより一般的に捉えなおします

▪ ここまでで疑問

▶ ノード分類で「隣同士は似ている」というけれど...

- 「隣同士は似ていない」
- 「隣が似ているもの同士は似ている」

などは扱えないの？

▶ リンク指標は恣意的、どれを使うの？ 全部使うの？ そもそもこれで全部なの？

▪ また、これまではノードあるいはリンクの独立性を仮定し、分類問題に帰着してきたが、ネットワーク構造の大局的な整合性は無視している

▪ ネットワーク構造全体のより一般的なモデルが考えられないか？

マルコフネットワークは、ネットワーク全体の確率モデルを定義します
たぶんモデルとしてはファイナルアンサーに近いですが、計算は大変です

▪ ネットワーク構造 G の特徴ベクトル $\Phi(G)$ を定義

- ▶ $G = \{X, Y\}$ は与えられている部分 X と、予測したい部分からなる
 - ノード分類の場合: X はノード特徴とリンク構造、 Y がノードラベル
 - リンク予測の場合: X はノード特徴と部分的なリンク、 Y が残りのリンク
- ▶ 各特徴 ($\Phi(G)$ の各要素) はネットワーク構成部品 (テンプレート) のマッチ回数

▪ Relational Markov Network

$$P(Y|X) = \frac{\exp(\langle \mathbf{w}, \Phi(G) \rangle)}{Z(X)}$$

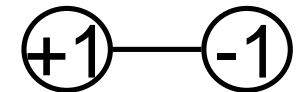
各部品の
重要度

$$Z(X) = \sum_Y \exp(\langle \mathbf{w}, \Phi(G) \rangle)$$

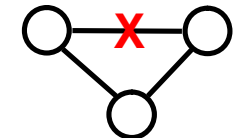
- ▶ $Z(X)$ の計算は大変。サンプリングなどによる近似が必要。
- ▶ 予測が大変

$$\hat{Y} = \operatorname{argmax}_Y P(Y|X) = \operatorname{argmax}_Y \langle \mathbf{w}, \Phi(G) \rangle$$

Taskar et al.: *Label and Link Prediction in Relational Data*, IJCAI Workshop on Learning Statistical Models from Relational Data, 2003



「+1の隣は-1」な特徴
(ノード分類)



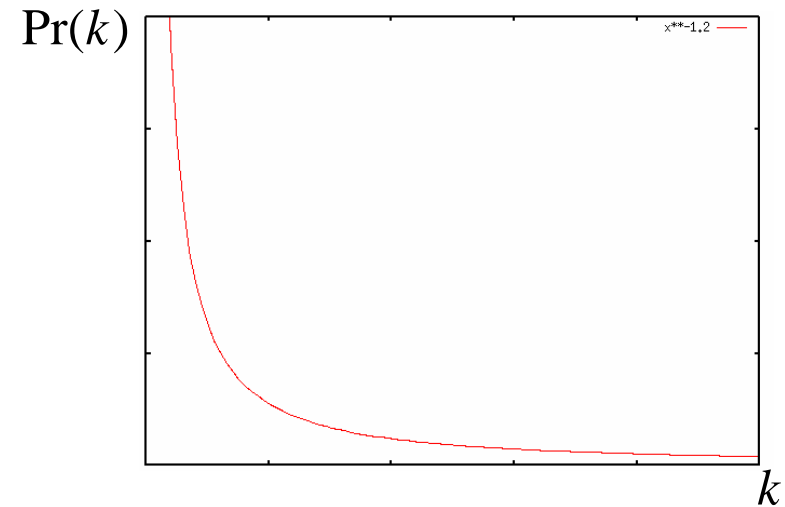
「友達の友達は友達
なんかじゃない」な特徴
(リンク予測)

ネットワーク構成部品
(テンプレート) の例

...なので、大規模なデータには適当な中間的なモデルを考えていくのがよいでしょう
例えば、ネットワークのスケールフリー性を積極的に予測に用いていくという試みもあります

- 現実世界のネットワーク構造は、しばしばスケールフリーネットワークになっている
 - ▶ つまり次数 k がべき分布に従う $\Pr(k) \propto k^{-\gamma}$ (γ は定数)
- ネットワーク構造に「スケールフリーネットワークになるような制約」を入れて予測を行うことで、より現実のネットワーク構造に近いものが構成できるはず
 - ▶ ネットワーク構造の事後分布に、スケールフリー性をもつ事前分布を入れる

(ネットワーク構造の事後分布) = (ノードペアからのリンク確率) \times (スケールフリーな事前分布)



ネットワーク構造データの解析において、機械学習の立場からどのような考え方に 基づいたアプローチが行われているかを紹介しました

- リンクマイニングのタスクのうち「ノード分類」と「リンク予測」を紹介しました
- 「隣は似ている」という考えに基づくアプローチを紹介しました
- ネットワーク構造の遷移モデルから出てくるいくつかのリンク指標を紹介しました
- これらの制約を除いたより一般的なモデルを紹介しました