2K2-3

Itemset Factorization Machines

石畠 正和　　マチュー・ブロンデル
Masakazu Ishihata　　Mathieu Blondel

北海道大学　　NTT コミュニケーション科学基礎研究所
Hokkaido University　　NTT Comunication Science Laboratories

We propose a new polynomial regression model called Itemset Factorization Machine (IFM), which is a generalization of Higher-order Factorization Machines (HOFMs). IFMs can consider an arbitrary subset of combinations of features. We also propose an efficient algorithm for evaluating and learning IFMs.

1. はじめに

回帰問題は人工知能・機械学習分野における中心的な問題の一つである．回帰問題の目標は，特徴ベクトル $\boldsymbol{x}$ と目標変数 y のペアが観測データとして複数与えられたとき，新たな $\boldsymbol{x}$ に対して，対応する y を予測することである．一般的に $\boldsymbol{x}$ から y の予測値を計算する関数を回帰モデルと呼び，この回帰モデルのパラメータを学習することで予測を達成する．予測対象が複雑な関係になれば，回帰モデルもより複雑である必要があるが，複雑なモデルに対するパラメータの学習は非常に困難である．例えば，予測対象の関係が，$\boldsymbol{x}$ の複数の特徴の組合せに依存するとき，そのパラメータ数は $\boldsymbol{x}$ の次元に対して指数的に増加する．そこでより少ないパラメータで，特徴の高次組合せを考慮した予測を行うことが，複雑な関係を学習するために重要となる．

特徴の高次組合せを考慮可能である回帰モデルとして Higher-order Factorization Machine (HOFM) [1] が提案されている．ℓ 次の HOFM は，$k \leq \ell$ 次のすべての特徴の組合せを考慮可能である．HOFM に対する効率的な学習アルゴリズムが提案されているが，ℓ が大きくなれば考慮すべき値の組合せは指数的に増加するため，多くの特徴の組合せはデータには現れない．よって全ての組合せを考慮するのは冗長である場合がある．

本稿では HOFM を任意の組合せ集合族を考慮できるように拡張した Itemset Factorization Machine (IFM) を提案する．HOFM は ℓ 次以下のすべての特徴組合せを考慮するのに対し，IFM は与えられた特徴組合せ集合のみを考慮する．与えられる特徴組合せの次数に制限はなく，また，一部の特徴組合せのみを考慮するため，HOFM よりもより高次な特徴組合せを効率的に扱うことが可能となる．本稿では IFM を効率的に評価・学習するアルゴリズムを提案する．提案手法は ZDD と呼ばれる組合せ集合族を効率的に扱うデータ構造を利用することにより，ZDD に比例する時間で IFM の評価・学習を可能にする．

2. 準備

2.1 回帰問題

ここではまず，本稿で扱う回帰問題を定義する．$\boldsymbol{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ $(d \in \mathbb{N})$ を d 次元の特徴変数ベクトルとし，$y \in \mathbb{R}$ を目標変数とする．特徴変数ベクトル $\boldsymbol{x}$ と目標変数 y には何らかの関係があるとし，$\boldsymbol{x}$ に対応する y を $y(\boldsymbol{x})$ と表す．$\boldsymbol{x}$ と $y(\boldsymbol{x})$ の実現値を T 組観測したとし，それらを $(\boldsymbol{x}^{(t)}, y^{(t)})$ $(t \in [T])$ と表す．

回帰問題とは，観測データ $D = \{(\boldsymbol{x}^{(t)}, y^{(t)}) \mid t \in [T]\}$ より，$\boldsymbol{x}$ と y の未知の関係を学習することである．観測データ D より学習した $\boldsymbol{x}$ と y の関係を関数 $f: \mathbb{R}^d \to \mathbb{R}$ で表し，これを回帰モデルと呼ぶ．回帰モデル $f(\boldsymbol{x})$ の精度を測る基準として，損失関数 $L: \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ を考える．損失関数 $L(y, y')$ の元での回帰モデル $f(\boldsymbol{x})$ の観測データ D に対する経験損失 $R(f; D)$ は以下である．

$$R(f; D) \equiv \frac{1}{T} \sum_{t \in [T]} L\Big(f(\boldsymbol{x}^{(t)}), y^{(t)}\Big) \tag{1}$$

本稿では L は μ-smooth な凸関数とし，L' を L の導関数とする．

回帰モデル $f(\boldsymbol{x})$ は一般的には調節可能なモデルパラメータを持ち，そのパラメータを調整することで未知の関係を学習する．一般的には上記の経験損失とモデルパラメータの正則化項の重み付き和を最小化するようにパラメータを最適化することで学習を実現する．以降，回帰モデルの例として線形モデルと Higher-order Factorization Machine について述べ，それらの最適化の目的関数を示す．

2.2 線形モデル

最も素朴な回帰モデル $f(\boldsymbol{x})$ として線形モデル $f_{\mathrm{LM}}(\boldsymbol{x}; \boldsymbol{w})$ が挙げられる．$\boldsymbol{w} = (w_1, \ldots, w_d) \in \mathbb{R}^d$ を線形モデルのパラメータベクトルと呼ぶ．このとき $f_{\mathrm{LM}}(\boldsymbol{x}; \boldsymbol{w})$ は以下の式により y を予測する．

$$f_{\mathrm{LM}}(\boldsymbol{x}; \boldsymbol{w}) \equiv \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{i \in [d]} w_i x_i \tag{2}$$

ここで，$\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ は d 次元ベクトルの内積である．観測データ D が与えられたとき，線形モデルでは以下の目的関数を最小化するようにパラメータベクトル $\boldsymbol{w}$ を学習する．

$$F_{\mathrm{LM}}(\boldsymbol{w}; D) \equiv R(f_{\mathrm{LM}}(\boldsymbol{x}; \boldsymbol{w})\,; D) + \frac{\alpha}{2} \|\boldsymbol{w}\|^2 \tag{3}$$

ここで第二項は正則化項と呼ばれ，$\alpha > 0$ は正則化の強さをコントロールするハイパーパラメータである．線形モデル $f_{\mathrm{LM}}(\boldsymbol{x}; \boldsymbol{w})$ は各特徴 x_i の重み w_i を調整することで，回帰モデルを学習する．しかし対象とするデータが複雑である場合，1つの特徴の重みを学習するだけでは $\boldsymbol{x}$ から y を予測することが不可能な場合がある．

連絡先: 石畠正和 ishihata.masakazu@ist.hokudai.ac.jp

2.3 Higher-Order Factorization Machine

特徴の組合せ効果を考慮した識別器として Factorization Machine (FM) [7, 8] が提案されている．(Second-order) FM では，全属性の 2 つ組 (ペア) に関して重みを導入し，その重み行列を低ランクの行列で近似することで，効率よく特徴の組合せ効果を考慮可能にする．

Higher-Order Factorization Machine (HOFM) [1] は FM をより高次の特徴の組合せを考慮できるように拡張したモデルである．$\ell \in \mathbb{N}$ を HOFM が考慮する特徴組合せの最大次数，$m_k \in \mathbb{N}$ $(k \in [\ell])$ をハイパーパラメータ (ランクパラメータ)，$\mathrm{P}^{(k)} \in \mathbb{R}^{d \times m_k}$ をパラメータ行列とする．$p_{i,j}^{(k)}$ を $\mathrm{P}^{(k)}$ の i 行 j 列目の要素，$\boldsymbol{p}_{*,j}^{(k)}$ と $\boldsymbol{p}_{i,*}^{(k)}$ をそれぞれ $\mathrm{P}^{(k)}$ の i 行ベクトル，j 列ベクトルとする．このとき, パラメータ $\mathbf{P} \equiv \{\mathrm{P}^{(k)} \mid k \in [\ell]\}$ を持つ HOFM $f_{\mathrm{HOFM}}(\boldsymbol{x}; \mathbf{P})$ は ANOVA Kernel $K_{\mathrm{A}}(\boldsymbol{x}, \boldsymbol{x}'; k)$ を用いて以下のように定義される [2].

$$f_{\mathrm{HOFM}}(\boldsymbol{x}; \mathbf{P}) \equiv \sum_{k \in [\ell]} \sum_{j \in [m_k]} K_{\mathrm{A}}\left(\boldsymbol{p}_{*,j}^{(k)}, \boldsymbol{x}; k\right), \tag{4}$$

$$K_{\mathrm{A}}(\boldsymbol{p}, \boldsymbol{x}; k) \equiv \sum_{S \in \binom{[d]}{k}} \prod_{i \in S} p_i x_i, \tag{5}$$

$$\binom{[d]}{k} \equiv \{S \subseteq [d] : |S| = k\} \tag{6}$$

ここで ANOVA Kernel は以下を満たす.

$$K_{\mathrm{A}}(\boldsymbol{p}, \boldsymbol{x}; 1) = \langle \boldsymbol{p}, \boldsymbol{x} \rangle \tag{7}$$

つまり ANOVA Kernel は通常のベクトルの内積をより高次の組合せに拡張したものと考えることができる．本来の HOFM [1] では $m_1 = 1$ であるが，本稿では表記を簡潔にするために上記のより一般的な定義を用いる．

HOFM $f_{\mathrm{HOFM}}(\boldsymbol{x}; \mathbf{P})$ では, データ・セット D に対して以下の目的関数を最小化するようにパラメータ P を学習する．

$$F_{\mathrm{HOFM}}(\mathbf{P}; D) \equiv R(f_{\mathrm{HOFM}}(\boldsymbol{x}; \mathbf{P}); D) + \sum_{k \in [\ell]} \frac{\alpha_k}{2} \|\mathrm{P}^{(k)}\|^2 \tag{8}$$

ここで $\alpha_k > 0$ は正則化の強さを調整するハイパーパラメータである．

HOFM は $k \in [\ell]$ 次組合せ特徴をすべて考慮した学習が可能である．しかし，k が大きくなるに連れ，その計算量は増加するため，より高次な組合せ特徴を扱うためには工夫が必要である．そこで本稿では，すべての k 次組合せ特徴を扱うのではなく，特定の組合せ特徴集合のみを考慮したモデルを提案する．提案手法は，特徴の組合せ集合の族 $\mathrm{S}^{(k)} \subseteq 2^{[d]}$ $(k \in [\ell])$ をハイパーパラメータとし，$\mathrm{S}^{(k)}$ に含まれる組合せ特徴 $S \in \mathrm{S}^{(k)}$ のみを考慮する．また次節で述べる，$\mathrm{S}^{(k)}$ を効率よく扱うデータ構造である ZDD を用いることで効率的な評価・学習を可能にする．

2.4 Zero-suppressed Binary Decision Diagram (ZDD)

本稿では HOFM をより一般的な特徴の組合せを扱えるように拡張する．ここでは提案手法が用いる組合せ集合族の圧縮表現である ZDD [5] について述べる．本稿では紙面の都合上，ZDD の構築法については割愛し，ZDD の満たす性質のみを述べる．

組合せ集合 $S \subseteq [d]$ をアイテムセットと呼び， 組合せ集合族 $\mathrm{S} \subseteq 2^{[d]}$ をアイテムセット集合と呼ぶ．ZDD は S を根付き非循環有向グラフ (DAG) により圧縮表現する手法である．$\Delta_{\mathrm{S}} = \langle V, r, E_0, E_1 \rangle$ を S を表現する ZDD とする．$V = \{0, 1, \ldots, |\mathcal{N}| - 1\}$ は ZDD の頂点集合であり， 0 と 1 を特に 0-terminal, 1-terminal と呼ぶ. b-terminal $(b \in \{0, 1\})$ は出次数が 0 であり，terminal 以外の節点の出次数は丁度 2 である．terminal 以外の節点を内点と呼ぶ．各内点 $v \in V \backslash \{0, 1\}$ はラベル $\mathrm{label}(v) \in [d]$ を持つ．$\mathcal{E}_b \subseteq \mathcal{N} \times \mathcal{N}$ は有向辺の集合であり， $e \equiv (v, u) \in \mathcal{E}_b$ を v の b-edge と呼び， u を v の b-child と呼ぶ．各内点 v は 0-edge と 1-edge を丁度 1 本ずつ持つ．v の b-child を v_b と表す．$r \in V$ は Δ_{S} の根である．内点 $v \in V \backslash \{0, 1\}$ とその子孫 $v' \in V$ において，本稿では一般性を失うことなく $v > v'$ かつ $\mathrm{label}(v) < \mathrm{label}(v')$ と仮定する．Δ_{S} 上で v から u に至る有向パス (辺の集合) の族を $\mathcal{P}(v, u) \subseteq 2^{E_0 \cup E_1}$ とする．このとき $S(p)$ $(p \in \mathcal{P}(r, 1))$ と $\mathrm{S}(v)$ を以下のように定める．

$$S(p) \equiv \{\mathrm{label}(v) : (v, u) \in p \cap E_1\}, \tag{9}$$

$$\mathrm{S}(v) \equiv \{S(p) : p \in \mathcal{P}(v, 1)\} \tag{10}$$

$S(p)$ は有向パス p に含まれる 1-edge の始点のラベル集合であり，これは p の表現するアイテムセットである．$\mathrm{S}(v)$ は v から 1-terminal に至るパス $p \in \mathcal{P}(v, 1)$ が表現するアイテムセットの集合である．また，定義より以下が成り立つ．

$$\mathrm{S}(v) = \{\{\mathrm{label}(v)\} \cup S : S \in \mathrm{S}(v_1)\} \cup \mathrm{S}(v_0), \tag{11}$$

$$\mathrm{S}(r) = \mathrm{S} \tag{12}$$

つまり Δ_{S} は，S を有向パスの集合として表現する．アイテムセット集合 S に対して $\mathrm{S}_i, \mathrm{S}_{-i}$ をそれぞれ以下のように定める．

$$\mathrm{S}_i \equiv \{S \backslash \{i\} : S \in \mathrm{S}, i \in S\}, \tag{13}$$

$$\mathrm{S}_{-i} \equiv \{S : S \in \mathrm{S}, i \notin S\} \tag{14}$$

よって $i = \mathrm{label}(v)$ とすれば $\mathrm{S}(v), \mathrm{S}(v_0), \mathrm{S}(v_1)$ の間には以下が成り立つ．

$$\mathrm{S}(v_1) = \mathrm{S}_i(v), \tag{15}$$

$$\mathrm{S}(v_0) = \mathrm{S}_{-i}(v) \tag{16}$$

つまり ZDD は S を上式に従い再帰的に分解し，同型の部分グラフを共有することで S を効率的に表現する．

3. 提案手法

HOFM $f_{\mathrm{HOFM}}(\boldsymbol{x}; \mathrm{P})$ は特徴の高次組合せを考慮できるが，次数 ℓ が大きくなると効率的に学習することが困難である．そこで本稿では，全ての高次特徴組合せではなく，与えられた特定の特徴組合せのみを考慮する，Itemset Factorization Machine (IFM) を提案する．

3.1 Itemset Factorization Machine (IFM)

アイテムセット集合 $\mathrm{S} \subseteq 2^{[d]}$ が与えられたとき，*Itemset Kernel* $K_{\mathrm{I}}(\boldsymbol{x}, \boldsymbol{x}'; \mathrm{S})$ を以下のように定義する．

$$K_{\mathrm{I}}(\boldsymbol{p}, \boldsymbol{x}; \mathrm{S}) \equiv \sum_{S \in \mathrm{S}} \prod_{i \in S} p_i x_i \tag{17}$$

ただし $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\{\})=1$ とする．Itemset Kernel と ANOVA Kernel の間には以下の関係が成り立つ．

$$K_\mathrm{I}\left(\boldsymbol{p},\boldsymbol{x};\binom{[d]}{k}\right)=K_\mathrm{A}(\boldsymbol{p},\boldsymbol{x};k) \tag{18}$$

つまり Itemset Kernel は ANOVA Kernel の任意の組合せ集合への一般化である．

アイテムセット集合の族 $\mathbf{S}\equiv\{\mathrm{S}^{(k)}\subseteq 2^{[d]}\mid k\in[\ell]\}$ が与えられたとき， パラメータ $\mathbf{P}$ を持つ IFM $f_\mathrm{IFM}(\boldsymbol{x};\mathbf{S},\mathbf{P})$ を以下のように定義する．

$$f_\mathrm{IFM}(\boldsymbol{x};\mathbf{S},\mathbf{P})\equiv\sum_{k\in[\ell]}\sum_{j\in[m_k]}K_\mathrm{I}\left(\boldsymbol{p}^{(k)}_{*,j},\boldsymbol{x};\mathrm{S}^{(k)}\right) \tag{19}$$

IFM は HOFM の ANOVA Kernel を Itemset Kernel に置き換えたものである．つまり IFM は HOFM の任意の組合せ集合族を扱えるように拡張したモデルである．

$f_\mathrm{IFM}(\boldsymbol{x};\mathrm{P})$ では, 観測データ D とアイテムセット集合族 $\mathbf{S}$ に対して以下の目的関数を最小化するようにパラメータ $\mathbf{P}$ を学習する．

$$F_\mathrm{IFM}(\mathbf{P};\mathbf{S},D)\equiv R(f_\mathrm{IFM}(\boldsymbol{x};\mathbf{S},\mathbf{P})\,;D)+\sum_{k\in[\ell]}\frac{\alpha_k}{2}\|\mathrm{P}^{(k)}\|^2 \tag{20}$$

3.2 IFM の評価

アイテムセット集合 $\mathrm{S}\subseteq 2^{[d]}$ のサイズは一般的に $O(2^d)$ であるため，Itemset Kernel $K_\mathrm{I}(\boldsymbol{x},\boldsymbol{x}';\mathrm{S})$ を式 (17) に従って評価した場合，その計算量は $O(2^d)$ である．つまり IFM の評価も一般的には指数的な時間が必要となる．本稿では，S を表現する ZDD Δ_S を用いて効率よく $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ を評価する手法を提案する．

$K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ は式 (13)(14) の $\mathrm{S}_i,\mathrm{S}_{-i}$ を用いて以下のように分解可能である．

$$K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})=p_ix_iK_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S}_i)+K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S}_{-i}) \tag{21}$$

よって $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ は上記の分解を繰り返すことで計算可能である．式 (10)(16)(15) を上記の分解に適用すれば，Δ_S を用いて $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ を評価可能である．Δ_S が与えられたとき，$v\in V$ に対して F_v と B_v をそれぞれ以下のように定める．

$$F_v=\begin{cases}1 & v=r\\ \displaystyle\sum_{(u,v)\in E_1}p_jx_jF_u & \text{otherwise}\end{cases} \tag{22}$$

$$B_v=\begin{cases}0 & v=0\\ 1 & v=1\\ p_ix_iB_{v_1}+B_{v_0} & \text{otherwise}\end{cases} \tag{23}$$

ここで $i=\mathrm{label}(v)\,,j=\mathrm{label}(u)$ である．F_v と B_v をそれぞれ v の Forward Weight, Backward Weight と呼ぶ．このとき $F_1=B_r=K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ が成り立つ．よって B_v を $v=0,1,\ldots,r$ まで順に計算することで $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})$ を計算可能であり，その計算量は Δ_S のサイズに比例する．つまり IFM $f_\mathrm{IFM}(\boldsymbol{x};\mathbf{S},\mathbf{P})$ の評価は, $\Delta_{\mathrm{S}^{(k)}}$ $(k\in[\ell])$ のサイズに比例する時間で計算可能である．

3.3 IFM の学習

HOFM の学習では式 (8) を最小化する．しかし， この関数は一般的には非凸関数であり，直接最小化することは困難である．式 (8) の局所解を発見する手法として，HOFM に対する Coordinate Decent (CD) 法が提案されている [1]．本稿では，これを式 (20) で定義される IFM の目的関数に適用することで， IFM に対する CD 法を構成する．IFM に対する CD 法は，以下の更新式により各パラメータを逐次的に更新する．

$$p^{(k)}_{i,j}\leftarrow p^{(k)}_{i,j}-(\eta^{(k)}_{i,j})^{-1}\frac{\partial}{\partial p^{(k)}_{i,j}}F_\mathrm{IFM}(\mathbf{P};\mathbf{S},D) \tag{24}$$

ここで $\eta^{(k)}_{i,j}$ は以下である．

$$\eta^{(k)}_{i,j}=\frac{\mu}{T}\sum_{t\in[T]}\frac{\partial\hat{y}^{(t)}}{\partial p^{(k)}_{i,j}}+\alpha_k \tag{25}$$

$$\hat{y}^{(t)}=f_\mathrm{IFM}\left(\boldsymbol{x}^{(t)};\mathbf{S},\mathbf{P}\right) \tag{26}$$

また $F_\mathrm{IFM}(\mathbf{P};\mathbf{S},D)$ の $p^{(t)}_{i,j}$ に関する偏微分は以下のように計算できる．

$$\frac{\partial}{\partial p^{(k)}_{i,j}}F_\mathrm{IFM}(\mathbf{P};\mathbf{S},D)=\sum_{t\in[T]}L'(y^{(t)},\hat{y}^{(t)})\frac{\partial\hat{y}^{(t)}}{\partial p^{(k)}_{i,j}}+\alpha_kp^{(k)}_{i,j} \tag{27}$$

$$\frac{\partial\hat{y}^{(t)}}{\partial p^{(k)}_{i,j}}=\frac{\partial}{\partial p^{(k)}_{i,j}}K_\mathrm{I}\left(\boldsymbol{p}^{(k)}_{*,j},\boldsymbol{x}^{(t)};\mathrm{S}^{(k)}\right) \tag{28}$$

よって Itemset Kernel の偏微分が評価できれば，CD 法を実行できる．

式 (21) の両辺を p_i で偏微分すると以下を得る．

$$\frac{\partial}{\partial p_i}K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S})=x_iK_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S}_i) \tag{29}$$

ここで $K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S}_i)$ は 式 (22)(23) で定義される F_v と B_v を用いて，以下のように計算できる．

$$K_\mathrm{I}(\boldsymbol{p},\boldsymbol{x};\mathrm{S}_i)=\sum_{(v,u)\in E_1:\mathrm{label}(v)=i}F_vB_u \tag{30}$$

よって式 (29) は $\Delta_{\mathrm{S}^{(k)}}$ のサイズに比例する時間で評価可能である．

これらのアルゴリズムは BDD/ZDD を用いた確率計算・学習アルゴリズム [3, 4] の Itemset Kernel への適用である．

3.4 スパースデータに対する IFM の効率化

$\boldsymbol{x}^{(t)}$ がスパースであるとき，IFM の評価と学習はより効率的に実行可能である．$\mathrm{support}\left(\boldsymbol{x}^{(t)}\right)=\{i\in[d]:x^{(t)}_i\neq 0\}$ とする．$|\mathrm{support}\left(\boldsymbol{x}^{(t)}\right)|\ll d$ であるとき， $\boldsymbol{x}^{(t)}$ はスパースであるという．式 (23) で定義される B_v において， $\mathrm{label}(v)=i$ かつ $x_i=0$ であるとき，$B_v=B_{v_0}$ が成り立つ．よって B_{v_1} を評価する必要はない．このような冗長な計算を回避するため，$\mathrm{S}^{(t,k)}$ を以下のように定義する．

$$\mathrm{S}^{(t,k)}\equiv\left\{S\in\mathrm{S}^{(k)}:S\subseteq\mathrm{support}\left(x^{(t)}\right)\right\} \tag{31}$$

$\mathrm{S}^{(t,k)}$ は $\mathrm{S}^{(k)}$ のうち $\mathrm{support}\left(x^{(t)}\right)$ に含まれるアイテムのみで構成されたアイテムセットの集合であり，一般的に $\mathrm{S}^{(k)}\leq\mathrm{S}^{(t,k)}$

である．また，$\mathrm{S}^{(k)}$ と $\mathrm{S}^{(t,k)}$ の間には以下の関係が成り立つ．

$$K_{\mathrm{I}}\Big(\boldsymbol{p},\boldsymbol{x}^{(t)};\mathrm{S}^{(k)}\Big)=K_{\mathrm{I}}\Big(\boldsymbol{p},\boldsymbol{x}^{(t)};\mathrm{S}^{(t,k)}\Big) \tag{32}$$

$|\mathrm{support}\Big(\boldsymbol{x}^{(t)}\Big)|\ll d$ であるとき， $\Delta_{\mathrm{S}^{(t,k)}}$ は $\Delta_{\mathrm{S}^{(k)}}$ と比べて非常に小さくなることが期待される．よって $\mathrm{S}^{(t,k)}$ を事前に構築することで，IFM の評価・学習の計算量の削減が期待できる．$\Delta_{\mathrm{S}^{(k)}}$ が与えられているとき，$\Delta_{\mathrm{S}^{(t,k)}}$ の構築は，ZDD がサポートしている演算を利用することで効率的に構築可能である [5]．

3.5 S の決定

IFM は与えられた組合せ集合の族 **S** に含まれる特徴組合せを考慮した回帰モデルの学習を可能にする．観測データ D から適切な **S** を選択することは IFM の重要な課題である．本稿では D から **S** を構成する方法の一例として頻出パターンマイニングを用いる方法を述べる．また，特徴間の関係がグラフ構造として与えられるとき，それを利用して **S** を構成する方法も述べる．

3.5.1 Frequent Pattern Mining for IFM

$\boldsymbol{x}^{(t)}$ がスパースであるとき， **S** を D から自動的に構成する方法として Frequent Pattern Mining (FPM) を用いる方法が考えられる．アイテムセット $S\subseteq[d]$ に対して support$(S;D)$ を以下のように定義する．

$$\mathrm{support}(S;D)\equiv|\{t\in[T]:S\subseteq\mathrm{support}\Big(x^{(t)}\Big)\}| \tag{33}$$

閾値 $\sigma\in\mathbb{N}$ が与えられたとき， 以下を Frequent Pattern と呼ぶ．

$$\mathrm{FP}(\sigma;D)=\{S\subseteq[n];\mathrm{support}(S;D)>\sigma\} \tag{34}$$

$\mathrm{FP}(\sigma;D)$ を $\mathrm{S}^{(k)}$ として採用すれば，観測データ D に含まれる頻出の特徴組合せを考慮した回帰モデルの学習が可能になる．FPM を高速に行う手法として LCM [9] が知られており， $\mathrm{FP}(\sigma;D)$ をそのサイズに比例する時間で出力可能である．また $\mathrm{FP}(\sigma;D)$ を表現する ZDD を直接出力する LCM over ZDD [6] が提案されており，これを用いることで効率的に $\Delta_{\mathrm{S}^{(k)}}$ を構成可能である．

3.5.2 Structural Information for IFM

構造正則化で利用される特徴の関係を表現したグラフを用いて **S** を構成可能である．例えば無向グラフ $G=\langle[d],E\rangle$ が事前知識として与えられたとする．G における辺 $(i,j)\in E\subseteq[d]\times[d]$ は特徴 x_i と x_j に何らかの類似関係があることを意味する．このようなグラフ構造 G が与えられたとき，G 中の特定の性質を満たす部分グラフを求め，そのノード集合族を $\mathrm{S}^{(k)}$ とすることで，事前知識を反映した **S** を構成できる．例えば $\mathrm{S}^{(k)}$ を G に含まれる極大クリークの集合とすれば，関連度の高い特徴の組合せを考慮する事が可能である．G 中の特定の性質を満たす部分グラフを列挙する手法としてフロンティア法が知られている [10]．フロンティア法は，G 中の特定の部分集合の族を表現する ZDD を G の構造を利用して Top-Down に構築する手法である．多くのフロンティア法は，部分グラフを辺の部分集合として表現するが，それらを頂点の集合に変換することで IFM に適用可能となる．また，幾つかの問題においては部分グラフを頂点の部分集合として表現するほうが効率的であることが知られており，これらの部分グラフに関しては，その出力 ZDD を直接 IFM の入力として利用することが可能である．

4. おわりに

本稿では HOFM を任意の特徴の組合せを考慮できるように拡張した IFM を提案した．IFM はアイテムセット集合族 **S** をハイパーパラメータとし，**S** に含まれる特徴の組合せを考慮した回帰モデルの学習を可能にする．また，$\mathrm{S}^{(k)}\in\mathbf{S}$ を表現する ZDD を用いた IFM の効率的な評価・学習アルゴリズムを提案した．IFM の評価と学習は ZDD のサイズに比例する時間で実行可能である．また，観測データがスパースであるときに，IFM の評価・学習を効率的に行う方法についても述べた．IFM では **S** をハイパーパラメータとするが，これを構成する方法についても述べた．

今後の課題は，本稿で提案した IFM や **S** の構成方法を実データに適用し， HOFM とその識別精度を比較する．IFM は，学習によって得られたパラメータから，回帰に大きく寄与する特徴の組合せを計算可能である．実データに対し，このような特徴組合せを計算することで，解釈可能な回帰が行えているかも確認したい．

参考文献

[1] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. Higher-Order Factorization Machines. In *NIPS 2016*, pages 3351–3359, 2016.

[2] Mathieu Blondel, Masakazu Ishihata, Akinori Fujino, and Naonori Ueda. Polynomial networks and factorization machines: New insights and efficient training algorithms. In *ICML 2016*, pages 850–858, 2016.

[3] Masakazu Ishihata, Yoshitaka Kameya, Taisuke Sato, and Shin-ichi Minato. Propositionalizing the em algorithm by bdds. In *ILP 2008*, pages 44–49, 2008.

[4] Masakazu Ishihata, Yoshitaka Kameya, Taisuke Sato, and Shin-ichi Minato. An EM algorithm on bdds with order encoding for logic-based probabilistic models. In *ACML 2010*, pages 161–176, 2010.

[5] Shin-ichi Minato. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In *DAC*, pages 272–277, 1993.

[6] Shin-ichi Minato, Takeaki Uno, and Hiroki Arimura. LCM over zbdds: Fast generation of very large-scale frequent itemsets using a compact graph-based representation. In *PAKDD 2008*, pages 234–246, 2008.

[7] Steffen Rendle. Factorization Machines. In *ICDM 2010*, pages 995–1000, 2010.

[8] Steffen Rendle. Factorization Machines with libFM. *ACM TIST*, 3(3):57:1–57:22, 2012.

[9] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. LCM: an efficient algorithm for enumerating frequent closed item sets. In *FIMI '03*, 2003.

[10] 川原純．フロンティア法 - 組合せ問題の解を列挙索引化する ZDD 構築アルゴリズム．`http://www-lsm.naist.jp/~jkawahara/frontier/frontier_lec.pdf`.