九州大学学術情報リポジトリ
Kyushu University Institutional Repository

SVMによる2部ランキング学習を用いたコンピュータ将棋における評価関数の学習

末廣, 大貴
九州大学大学院システム情報科学府

畑埜, 晃平
九州大学大学院システム情報科学府

坂内, 英夫
九州大学大学院システム情報科学府

瀧本, 英二
九州大学大学院システム情報科学府

他

https://hdl.handle.net/2324/1505699

出版情報：The IEICE transactions on information and systems. J97-D (3), pp.593-600, 2014-03.
The Institute of Electronics, Information and Communication Engineers
バージョン：
権利関係：©一般社団法人電子情報通信学会 2014

論　文

SVMによる2部ランキング学習を用いたコンピュータ将棋における評価関数の学習*

末廣　大貴† 畑埜　晃平† 坂内　英夫† 瀧本　英二† 竹田　正幸†

Learning Evaluation Functions for Shogi Using SVM-Based Bipartite Ranking Learning*

Daiki SUEHIRO†, Kohei HATANO†, Hideo BANNAI†, Eiji TAKIMOTO†, and Masayuki TAKEDA†

あらまし　近年，将棋の評価関数の設計においては，機械学習を応用してパラメータの自動調整を行う手法が主流となっている．ただし，評価項目（特徴）は作成者の棋力，感覚に基づいて用意されることが多く，これまで，複数の駒同士の関係など，複雑な特徴が数多く考案されてきた．本研究では，明示的に用意する特徴としては局面を表す基本的で単純なもののみとし，多項式カーネルとサポートベクターマシン（SVM）を用いて評価関数の学習を行う手法を提案する．多項式カーネルを用いることにより，単項式で表現できる特徴間の n 項関係を，全て高次の特徴として利用することができる．また，評価関数の学習問題を，合法手後の局面を順位づける2部ランキングの問題と捉え，SVMを用いて学習を行う手法（ランキングSVM法）を提案する．対局や棋譜との一致率を調べる実験結果，及び駒組みの観察等から，ランキングSVM法の有効性を示す．

キーワード　評価関数，SVM，多項式カーネル，2部ランキング，ランキング学習

1. まえがき

人工知能分野において，チェス，将棋，囲碁など様々な思考ゲームの研究がなされている．将棋は，初手からの組み合わせが10の220乗ほども言われており，必勝手順をコンピュータがしらみ潰しに求めることは不可能と考えられている．しかし，コンピュータ将棋は近年急速に進歩しており，いまやプロ棋士とも互角な勝負をするまでに至っている．その大きな要因の一つとして，保木によるBonanza Methodと呼ばれる手法[13]の登場がある．コンピュータ将棋を成す重要なファクターとして，局面の優劣を判定する「評価関数」がある．従来，評価関数は人間の手作業により作成されており，それには大変な労力がかかるという問題と，作成者の棋力，感覚に依存してしまうという問題があった．Bonanza Methodは，プロ棋士の棋譜を用いて，各局面においてプロ棋士の指した手を学習の教師とし，その手が最大の評価値を与えるように評価関数の学習を行う手法である．Bonanza Methodの成功以来，多くの強豪プログラムで，評価関数の作成に機械学習を用いた手法が取り入れられている．

しかし，この手法にも依然問題は残されている．局面評価にどのような評価項目（特徴）を用意すべきかということについては確立した手法がなく，依然として作成者の棋力，感覚に依存しているという点である．例えば，有用な特徴として，複数の駒同士の位置関係や，玉の周囲の利きの数などがあげられるが，多くの強豪プログラムでは，このような複雑な特徴が作成者自身により多数用意され，結果として膨大な次元数の特徴ベクトルが用いられている．しかし，有用な関係を網羅しているかどうかは不明であり，その組み合わせの数を増やすと学習に要する計算時間も増大してしまう．

† 九州大学大学院システム情報科学府，福岡市
Graduate School/Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan

* 本論文はシステム開発論文である．

ⓒ一般社団法人電子情報通信学会 2014

本論文ではこの問題を解決する手段として，明示的に用意する評価項目は局面を表す基本的で単純なもの（低次の特徴）のみとし，多項式カーネルと SVM（サポートベクターマシン）を用いて学習を行う手法を提案する(注1)．次数 n の多項式カーネルを用いることにより，低次の特徴の高々n 次の単項式で表現できる特徴を全て網羅することができる．また，カーネルの計算量は次数 n に依存しないため，n を増やしても全体の計算量は変化しない．

SVM は 2 クラス分類に用いられる線形学習機械として，非常に優れた性能を示す．したがって，棋譜データに存在する局面を正例，存在しない局面を負例として学習サンプルを構成し，これを SVM に与えて得られる関数 f_{SVM} を評価関数として用いると，局面 $\boldsymbol{x}$ に対する評価関数の値 $f_{\mathrm{SVM}}(\boldsymbol{x})$ が正ならば $\boldsymbol{x}$ はプロ棋士が指しそうな手，負ならばそうではない手という合理的な判定を下すことができる．しかし，評価関数の値そのもの $f_{\mathrm{SVM}}(\boldsymbol{x})$ は，局面 $\boldsymbol{x}$ の優劣を表しているわけではない．したがって，例えば次局面の評価値が全て負である場合，その中で最も評価値の高い局面を最善と考える根拠がない．局面の評価関数として求められている性質は，その値が局面の優劣を表すこと，すなわち，局面集合に対して正しい順位付け（ランキング）を行うことである．そこで，評価関数の学習を，2 値ラベルのついた学習サンプルからランキング関数を学習する，いわゆる 2 部ランキング学習の問題 [7] と捉え，ランキング学習の手法を用いる．

ランキング学習は，情報検索，推薦システム，バイオインフォマティクスなど様々な分野に応用をもつため，非常に多くの研究が行われている [1]〜[7], [9]〜[11]．2 部ランキング学習の目標は，正例と負例のランダムな対に対し，高い確率で，正例の評価値が負例の評価値よりも大きくなるようなランキング関数を求めることである．ランキング学習の素朴な方法として，ランキング問題を 2 クラス分類問題に還元する方法がある [2], [5], [7], [10], [11]．したがって，この問題に対して，SVM などの任意の 2 クラス分類学習アルゴリズムを適用することができる [2], [5], [7]．しかし，通常，この還元では，与えられた学習サンプルのサイズの 2 乗の大きさのサンプルを新たに構成する必要があるため，効率が悪い．本論文では，この還元を，合法手により得られる次局面集合からなる部分サンプルにのみ適用することにより，必要メモリの大幅な削減及び，計算時間の大幅な削減を図る．以降では，この提案アルゴリズムをランキング SVM 法と呼ぶ．

本論文は 5 節から成り，**2.** では SVM とカーネル法に関する基本的な説明，**3.** ではランキング SVM の説明と，ランキング SVM 法を用いた評価関数の学習手法について述べる．**4.** では対局や棋譜との一致率の実験を行い，カーネル法及びランキング SVM 法がコンピュータ将棋に有効であることを示し，**5.** ではランキング SVM 法の今後の課題について述べる．

2. SVM による評価関数の学習

本節では，評価関数の学習問題を 2 値分類学習器の問題と捉え，SVM を適用する標準的な手法について述べる．

2.1 SVM

まず，SVM による学習の一般的な枠組みについて述べる．事例空間を $X \subseteq R^n$，ラベル空間を $Y = \{-1, 1\}$ とし，学習サンプルを $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \subseteq X \times Y$ とする．$\boldsymbol{x}_i$ を事例，y_i を事例 $\boldsymbol{x}_i$ のラベルと呼ぶ．m は事例の数を表す．SVM は，入力としてサンプル S が与えられると，（ソフト）マージン最大化の原理に基づき，カーネルによって定義される特徴空間における線形関数 $f : X \to \mathbb{R}$ を求めるものである．通常，f は任意の事例 $\boldsymbol{x} \in X$ に対し，$f(\boldsymbol{x})$ の符号に基づいて 2 クラス分類を行う仮説として用いられる．関数 f は，カーネル $K : X^2 \to \mathbb{R}$ を用いて

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}) + b^*$$

と表すことができる．ここで，α_i^* $(i = 1, 2, \ldots, m)$ は，以下の 2 次計画問題の最適解である．

Maximize

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(\boldsymbol{x_i}, \boldsymbol{x_j})$$

subject to

$$\sum_{i=1}^{m} y_i \alpha_i = 0,$$
$$C \geq \alpha_i \geq 0 \quad (i = 1, \ldots, m).$$

(注1)：カーネル法に基づく手法については，[14] にて一部発表済みである．

また，b^* は，$\alpha_i^* \neq 0$ かつ $\alpha_i^* \neq C$ を満たす全ての i に対し，$y_i f(\boldsymbol{x}_i) = 1$ を満たす実数，C は適当なパラメータである．

2.2 局面ベクトル

本研究では，事例ベクトル $\boldsymbol{x}$ の各成分を表す特徴（低次の特徴）として，局面を表現するための基本的で単純な特徴のみを用いる．以下では，事例ベクトルを局面ベクトルと呼ぶ．局面ベクトルの各成分の内訳は，

(1) 盤上にある各駒の数

(2) 持ち駒の種類とそれぞれの数

(3) 盤上の各枡に位置している駒の種類

(4) 盤上の各枡に対する利きの有無とその駒の種類

である．例えば初期局面であれば，先手の歩が盤上に 9 枚あること，先手の金が盤上に 2 枚あること，1 三に後手の歩があること，1 四に後手の歩が利いていること等が，局面ベクトルの成分となる．ただし，実際の局面ベクトルは，次のように冗長性をもたせることにより，全ての成分が 0 か 1 の値を取るものとなる．項目 (1) の場合，各駒

$$p \in P \equiv \{\text{先手歩}, \text{後手歩}, \ldots, \text{先手馬}, \text{後手馬}\}$$

と各整数 $i\,(0 \leq i \leq k_p)$ に対し，「駒 p が盤上に i 枚存在するか否か」を 1 か 0 で符号化することにより，$\sum_{p \in P}(k_p + 1)$ 次元分のベクトル成分で表す．k_p は，駒 p が盤上に存在し得る数の上限であり，例えば，$p =$「先手歩」の場合 $k_p = 9$，$p =$「後手と金」の場合 $k_p = 18$ となり，項目 (1) の次元数は 130 となる．また，項目 (2) の次元数は 84 となり，項目 (3) は,「各枡に駒 p が存在するか否か」がそれぞれ特徴となり,（ただし，歩，桂，香はルール上位置することができない枡がある）次元数は 2196 となる．項目 (4) は「各枡に駒 p の利きが存在するか否か」がそれぞれ特徴となり，利きが存在し得ない駒と枡の組を省き，次元数は 2358 となる．

こうして，局面ベクトルの次元数 n は 4768 となる．

2.3 多項式カーネル

多くの強豪プログラムでは，こうした低次の特徴の他に，作成者の感覚により有用と判断された複雑な特徴が多数追加され，膨大な次元数の局面ベクトルが用いられている．例えば，玉と金の位置関係や，玉の周囲の利きの数などがあげられる．本研究では，こうした特徴の多くが低次の特徴の d 項関係として表すことができる点に着目し，多項式カーネルを用いる．特に，d 次の多項式カーネルは

$$K(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a} \cdot \boldsymbol{b} + 1)^d$$

で定義される関数で，対応する特徴空間 $\mathbb{R}^N$，$N = \binom{n+d}{d}$ の各次元が，高々d 次の単項式に対応する．すなわち，この特徴空間は，低次の特徴の k 項関係（$k \leq d$）で，単項式として表すことができるものを全て特徴として含んでいる．すなわち，複数の駒同士の位置関係等や，玉の周囲の利きの数といった特徴も，この低次の特徴（駒の位置，玉の位置，駒の利き等）の k 項関係として非明示的に表現されることとなる．また，カーネルの計算量は次数 d に依存しないことから，特徴空間の次元数 N がどれだけ大きくとも，学習に要する計算量は増大しない．

2.4 多項式カーネルの優位性

現在のコンピュータ将棋ソフトの多くでは，評価関数に用いられる特徴は人間の知識，感覚により手作業で用意されている．例えばコンピュータ将棋ソフト「Bonanza」では，

(a) 駒の価値

(b) 王，他の駒二つの位置

(c) 隣接しあった駒二つの位置関係

(d) 竜馬飛角桂香の利き上にいる駒の種類

(e) 竜馬飛角香が動ける枡の数

(f) ピンされている駒の種類，方向，王との距離

(g) 角と同じ色の枡にいる味方の歩の数

(h) 歩桂銀が前進できるか

(i) 竜飛香の前後の歩

(j) 王の周囲 25 枡の利きの配置

が特徴として用いられていた [13]．これらの特徴は，全て **2.2** で用意した基本的な特徴の積和形論理式で表現可能である．例えば，(c) の特徴の一つとして,「銀の隣に金が存在する」という特徴 A は以下のように表現できる．盤上の各枡を $l(1 \leq l \leq 81)$ とおき，**2.2** の項目 (3) の特徴を，駒 p が枡 l に存在するかどうか

を表す論理変数 $x_{p,l}$ で表現したとき，

$$A = (x_{\text{金},\,1} \wedge x_{\text{銀},\,2}) \vee \cdots \vee (x_{\text{金},\,81} \wedge x_{\text{銀},\,80})$$

のように表現できる．ここで，各項が排他的であることに注意すると，A は 2 次の多項式

$$A = x_{\text{金},\,1}x_{\text{銀},\,2} + \cdots + x_{\text{金},\,81}x_{\text{銀},\,80}$$

と表現することもできる．同様にして，(a)∼(j) は基本的な特徴の d 項関係，すなわち d 次多項式で表すことができる．多項式カーネルを用いた特徴空間は，これらの高次の特徴を全て含むので,「Bonanza」の特徴を全て含んでいることになる．また，他の強豪ソフトで用いられているような更に複雑な特徴，例えば「飛車成りを受けている駒の種類とその駒に利いている駒の種類」,「桂馬の前に相手が歩を打てる」などの特徴も，多項式カーネルによって表現可能である．よって，多項式カーネルによる特徴の自動生成はコンピュータ将棋において有効であると考えられる．

2.5 棋譜データの利用

本研究では，多くの強豪プログラムと同様に，プロ棋士の棋譜データセットを用いて学習サンプル S を構築する．すなわち，各棋譜上の各局面 $\boldsymbol{z}$ に対し，合法手により得られる全ての次局面からなる集合 $\mathrm{Next}(\boldsymbol{z})$ を求め，各次局面 $\boldsymbol{x} \in \mathrm{Next}(\boldsymbol{z})$ に対し，局面 $\boldsymbol{x}$ が棋譜データセットのどこかに存在するなら正例 $(\boldsymbol{x}, 1)$，存在しないなら負例 $(\boldsymbol{x}, -1)$ としてサンプル S に加える．こうして作成したサンプルを訓練データとして学習アルゴリズムに与え，評価関数を作成する．

2.6 SVM 法

学習アルゴリズムとして SVM を用いた素朴な方法について述べる．以降では，この手法を SVM 法と呼ぶ．**2.5** で作成したサンプルを SVM に与え，得られる関数を f_{SVM} とする．対局を行う際は，f_{SVM} を局面の評価関数として用いる．すなわち，局面 $\boldsymbol{x}$ に対し，$f_{\mathrm{SVM}}(\boldsymbol{x})$ がその局面の評価値となる．本研究では，与えられた局面 $\boldsymbol{z}$ の次の一手として，次局面 $\boldsymbol{x}^*$ が

$$f_{\mathrm{SVM}}(\boldsymbol{x}^*) = \max_{\boldsymbol{x} \in \mathrm{Next}(\boldsymbol{z})} f_{\mathrm{SVM}}(\boldsymbol{x})$$

を満たすような手を選択する．

3. ランキング SVM による評価関数の学習

本節では，評価関数の学習を，2 値ラベルのついた学習サンプルからランキング関数を学習する 2 部ランキング関数の学習問題と捉え，SVM を用いた学習法を提案する．以下では，2 部ランキング関数の学習を，単にランキング学習と呼ぶ．

3.1 ランキング学習

ここでは，ランキング学習の一般的な枠組みと，これを 2 クラス分類学習の問題に還元する手法について述べる．ランキング学習では，入力として，**2.1** で述べたような，2 値ラベルのついたサンプル

$$S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \subseteq X \times \{1, -1\}$$

が与えられる．学習アルゴリズムは，仮説としてランキング関数と呼ばれる実数値関数 $f : X \to \mathbb{R}$ を出力する．ここで，各事例 $\boldsymbol{x}_i$ は，X 上のある確率分布 P に従ってそれぞれ独立に選ばれると仮定する．アルゴリズムの目標は，分布 P を正例集合に制限した分布 P^+ に従って選ばれた事例 $\boldsymbol{x}$ と，負例集合に制限した分布 P^- に従って選ばれた事例 $\boldsymbol{x}'$ に対し，高い確率で $f(\boldsymbol{x}) > f(\boldsymbol{x}')$ が成り立つことである．この不等式が成立しない確率を，ランキング関数 f の汎化誤差と呼ぶ．いま，学習アルゴリズムが用いる仮説クラスを F とする．すなわち，アルゴリズムが出力するランキング関数 f は，F から選ばれるとする．

この問題は，次のようにして 2 クラス分類学習の問題に還元できる．まず，サンプル S から，事例空間 $X \times X$ 上のサンプル

$$\hat{S} = \{((\boldsymbol{x}_i, \boldsymbol{x}_j), 1) \mid y_i = 1, y_j = -1, 1 \leq i, j \leq m\}$$

を構成する．すなわち，$\hat{S}$ は，S の全ての正例 $\boldsymbol{x}_i$ と全ての負例 $\boldsymbol{x}_j$ の対 $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ を事例としてもつ．全ての事例には正のラベルが付与されていることに注意されたい．$\hat{S}$ の各事例は，分布 $P^+ \times P^-$ からそれぞれ独立に選ばれたものと考えることができる．ここで，

$$\hat{F} = \{\hat{f}(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}) - f(\boldsymbol{x}') \mid f \in F\}$$

とすると，$\hat{F}$ を仮説クラスとし，サンプル $\hat{S}$ から 2 値分類仮説 $\hat{f} \in \hat{F}$（$\hat{f}$ の符号が正のとき 1，負のとき 0 と分類）を求めることができれば，元の問題に対するランキング関数 $f \in F$ を得ることができる．明らかに，2 値分類関数 $\hat{f}$ とランキング関数 f の汎化誤差は等しい．すなわち，事例 $(\boldsymbol{x}, \boldsymbol{x}')$ が分布 $P^+ \times P^-$ に従って選ばれるとき，$\Pr(\hat{f}(\boldsymbol{x}, \boldsymbol{x}') < 0) = \Pr(f(\boldsymbol{x}) < f(\boldsymbol{x}'))$.

3.2 ランキング SVM

前節において，仮説クラス F を特徴空間上での

線形関数のクラスとすることにより，SVM を用いてランキング学習を行うことができる [2]．特徴写像 $\phi : X \to \mathbb{R}^N$ を固定し，

$$F = \{f(\boldsymbol{x}) = W \cdot \phi(\boldsymbol{x}) \mid W \in \mathbb{R}^N\}$$

とする．ランキング関数の定数項は，一般性を失わずに 0 とおくことができることに注意されたい．ϕ により定義されるカーネルを

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$$

とおく．更に，特徴写像 $\psi : X \times X \to \mathbb{R}^N$ を

$$\psi(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) - \phi(\boldsymbol{x}')$$

と定義すると，前節で述べたように，仮説クラス F を用いたランキング学習の問題は，$X \times X$ 上の線形関数のクラス

$$\hat{F} = \{\hat{f}(\boldsymbol{x}, \boldsymbol{x}') = W \cdot \psi(\boldsymbol{x}, \boldsymbol{x}') \mid W \in \mathbb{R}^N\}$$

を仮説クラスとする 2 値分類学習問題に還元される．したがって，$X \times X$ 上のカーネル

$$\begin{aligned}
&\hat{K}((\boldsymbol{x}_1, \boldsymbol{x}_1'), (\boldsymbol{x}_2, \boldsymbol{x}_2')) \\
=&\psi(\boldsymbol{x}_1, \boldsymbol{x}_1') \cdot \psi(\boldsymbol{x}_2, \boldsymbol{x}_2') \\
=&(\phi(\boldsymbol{x}_1) - \phi(\boldsymbol{x}_1')) \cdot (\phi(\boldsymbol{x}_2) - \phi(\boldsymbol{x}_2')) \\
=&K(\boldsymbol{x}_1, \boldsymbol{x}_2) - K(\boldsymbol{x}_1, \boldsymbol{x}_2') \\
&-K(\boldsymbol{x}_1', \boldsymbol{x}_2) + K(\boldsymbol{x}_1', \boldsymbol{x}_2')
\end{aligned}$$

を用いて，サンプル $\hat{S}$ を SVM に与えることにより，汎化誤差の小さい仮説 $\hat{f}$ が得られることが期待できる．通常，仮説 $\hat{f}$ は，$\hat{S}$ の各事例に対する重み $\{\alpha_{i,j} \mid y_i = 1, y_j = -1, 1 \le i, j \le m\}$ として得られる．仮説 $\hat{f}$ の重みベクトル W は，

$$W = \sum_{i,j} \alpha_{i,j} \psi(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

のように表せるので，元のランキング問題における仮説ランキング関数 f は，

$$\begin{aligned}
f(\boldsymbol{x}) &= W \cdot \phi(\boldsymbol{x}) \\
&= \sum_{i,j} \alpha_{i,j} \psi(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \phi(\boldsymbol{x}) \\
&= \sum_{i,j} \alpha_{i,j} (K(\boldsymbol{x}_i, \boldsymbol{x}) - K(\boldsymbol{x}_j, \boldsymbol{x}))
\end{aligned}$$

と表すことができる．

3.3 棋譜データの利用

ランキング SVM においては，一般に，還元の際に生成されるサンプル $\hat{S}$ の大きさが $\Omega(m^2)$ となるので，元のサンプルのサイズ m が大きい場合には，前節の還元手法は適用できない．実際，プロ棋士の 5 万局の棋譜データセットを用いて学習サンプル S を構築すると，m は 4 億を超える値となる．

しかし，将棋においては，あらゆる局面を順位付けするのではなく，次局面の候補集合に対してのみ順位づけができればよい．したがって，本論文では，サンプル $\hat{S}$ を次のように構成する．棋譜データ上の各局面 $\boldsymbol{z}$ に対し，$\boldsymbol{z}$ の次局面のみからなる S の部分サンプル

$$S_{\boldsymbol{z}} = S \cap (\mathrm{Next}(\boldsymbol{z}) \times \{1, -1\})$$

に対して前節の還元 $\hat{S}_{\boldsymbol{z}}$ を行い，

$$\hat{S} = \bigcup_{\boldsymbol{z}} \hat{S}_{\boldsymbol{z}}$$

とする．こうして作成したサンプルをランキング SVM に与え，得られる関数を f_{RSVM} とする．

対局を行う際は，f_{RSVM} を局面の評価関数として用いる．すなわち，局面 $\boldsymbol{x}$ に対し，$f_{\mathrm{RSVM}}(\boldsymbol{x})$ がその局面の評価値となる．すなわち，与えられた局面 $\boldsymbol{z}$ の次の一手として，次局面 $\boldsymbol{x}^*$ が

$$f_{\mathrm{RSVM}}(\boldsymbol{x}^*) = \max_{\boldsymbol{x} \in \mathrm{Next}(\boldsymbol{z})} f_{\mathrm{RSVM}}(\boldsymbol{x})$$

を満たすような手を選択する．

4. 実　　験

本章では，カーネル法の有効性を示す例として対局実験（各大会における実践）と，評価関数の出力した手のランキングがどれだけ棋譜と一致しているかを測る実験を行う．

4.1 実 戦 譜

棋譜約 100 局よりサンプルを作成し，多項式カーネルの次数 d は 10 とし，SVM-light [8] を用いて学習を行い，f_{SVM} を得た．この評価関数を用いてコンピュータ将棋ソフト「STR」の実装を行い，コンピュータ将棋の各大会において他ソフトと対局を行った．他のコンピュータ将棋ソフトに用いられているような定跡データベースは一切用いておらず，7 手詰みルーチンを組み込んである以外は，先読みのための探索は行っ

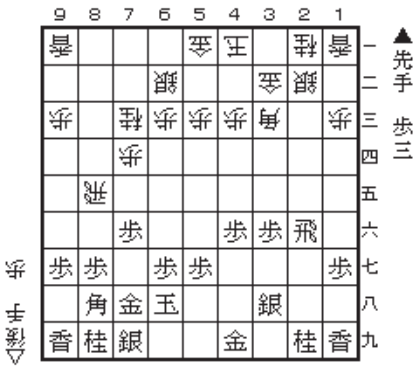

図 1　先手 ponanza，後手 STR．後手 5 一金まで
Fig. 1　ponanza (black) vs STR (white).

図 2　先手 Bonanza，後手 STR．後手 3 二銀まで
Fig. 2　Bonanza (black) vs STR (white).

ていない．図 1 は，第 20 回世界コンピュータ将棋選手権において，ponanza と対局した際に現れた局面，図 2 は，第 15 回コンピュータオリンピアード将棋部門において Bonanza と対局した際に現れた局面である．定跡データを組み込んでいないにもかかわらず，第 1 図では ponanza の横歩取りに対し，8 五飛車戦法で中原囲いと呼ばれる陣形を作っている．第 2 図では Bonanza の振り飛車穴熊に対し，居飛車戦法で美濃囲いと呼ばれる陣形を作っており，人間のような自然な駒組みを行えていることがわかる．先読みや定跡データベースの搭載を行わずこのような人間らしい駒組みを行うためには，駒同士の位置関係や相手の戦法など，多数の特徴の組み合わせを考慮しなければならず，カーネル法の有効性が現れていると言える．

4.2　棋譜との一致率

次に，SVM 法とランキング SVM 法との比較実験を行う．無作為に選んだ約 1000 局の棋譜をサンプルとして用いて学習を行った(注2)．SVM の実装には Pegasos [12] を用いた．Pegasos はランダムサンプリングを用いた SVM で，従来の SVM 手法よりも高速かつ省メモリであるため多くのサンプルを与えることができる．Pegasos のパラメータ λ（正規化パラメータ）を 0.01，0.05，0.1，ϵ（解の精度を調整するパラメータ）を 0.01，0.1 と変化させ，計 16 個の評価関数 f_{SVM}，f_{RSVM} を作成した．アルゴリズムの反復回数 T は，$\frac{1}{\lambda\epsilon}$ としてある．また，無作為に選んだテストデータ約 100 局分を用意し，棋譜上の投了局面以外の全ての局面に対して合法手後の局面を生成し，それぞれの評価関数を用いてランキング付けを行う．なお，探索はもちろん，定跡データベースや詰将棋ルーチンは組み込んでいない．ランキング r 位以内の手を指した局面の中に，棋譜の次局面と一致したものがあったとき，第 r ランキング以内で一致したとみなす．プロ棋士の棋譜との一致率を，(棋譜の一致数/棋譜での全局面数）とする．実験結果を表 1 に示す．ランキング 1 位の一致率にこそあまり差がないものの，多くのパラメータ設定において f_{RSVM} の方が，高い一致率を示していることがわかる．

表 1　プロ棋士の棋譜との一致率
Table 1　Agreement rate with professional shogi player.

ϵ	λ	r	SVM	RSVM
0.001	0.01	1	0.056	0.058
		3	0.119	**0.146**
		5	0.163	**0.2014**
	0.05	1	0.057	**0.063**
		3	0.118	0.137
		5	0.163	0.194
	0.1	1	0.056	0.049
		3	0.119	0.119
		5	0.163	0.172
0.01	0.01	1	0.056	0.051
		3	0.118	0.116
		5	0.162	0.172
	0.05	1	0.056	0.038
		3	0.121	0.110
		5	0.164	0.156
	0.1	1	0.058	0.056
		3	0.120	0.131
		5	0.164	0.174

更に，パラメータが $\epsilon = 0.001, \lambda = 0.01$ の f_{SVM}，f_{RSVM} を用い，各 r に対する棋譜との一致率を度数分布にしたものが図 3 である．r を増やしていくと，

(注2)：ランキング SVM 法を用いた場合，1000 局をサンプルとした学習でも約 1 周間ほどの時間を要する．Bonanza 等強豪ソフトで行っている数万局の学習には膨大な計算時間を要するため，本論文ではサンプル数を大幅に小さくして実験を行った．

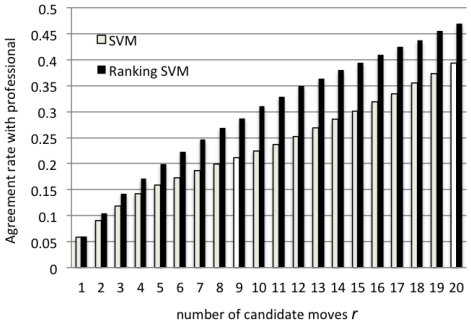

図 3　プロ棋士の棋譜との一致率 ($\epsilon = 0.001, \lambda = 0.01$)
Fig. 3　Agreement rate with professional shogi player.

f_{RSVM} が f_{SVM} に比べ著しく一致率が高くなっている．これらの結果から，SVM 法に比べ，ランキング SVM 法がコンピュータ将棋に対して有効であることが考えられる．

5. む す び

より正確な局面評価を行うためには，ある程度 minimax 探索を行うことが必要であろう．したがって，学習のフェーズにおいても，サンプルに含まれる局面ベクトルをそのまま事例として用いるのではなく，minimax 探索を行って得られる数手先の局面を事例として用いることが考えられる．しかし，ここでの minimax 探索は，教師（プロ棋士）の読みによるものでなければならないが，そのようなデータは存在しない．したがって，minimax 探索を組み込んだ場合の学習の目的関数として，**2.1** で記したような二次計画問題としての定式化ができそうもない．そこで，現在の評価関数を用いて行うという，標準的なヒューリスティックを用いることなどが考えられる．また，学習サンプルを増やすほど SVM が出力する仮説の重みベクトルの数が増大するため，実際の対局等において局面ベクトルとの内積を計算するために多くの時間を要する．よって，重みの小さい仮説は切り捨てるなどの工夫が必要である．

文　　献

[1] N. Balcan, N. Bansal, A. Beygelzimer, D. Coppersmith, J. Langford, and G.B. Sorkin, “Robust reductions from ranking to classification,” Proc. 20th Annual Conference on Learning Theory (COLT 2007), pp.604–619, 2007.

[2] U. Brefeld and T. Scheffer, “Auc maximizing support vector learning,” Proc. ICML Workshop on ROC Analysis in Machine Learning, 2005.

[3] W.W. Cohen, R.E. Schapire, and Y. Singer, “Learning to order things,” J. Artificial Intelliegence Research, vol.10, pp.243–279, 1999.

[4] C. Cortes and M. Mohri, “Auc optimization vs. error rate minimization,” Adances in Neural Information Processing Systems 16, 2004.

[5] Y. Freund, R. Iyer, R.E. Shapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” Journal of Machine Learning Research, vol.4, pp.933–969, 2003.

[6] J. Moribe, K. Hatano, E. Takimoto, and M. Takeda, “Smooth boostring for margin-based ranking,” Proc. 19th International Conference on Algorithmic Learning Theory, pp.227–239, 2008.

[7] T. Joachims, “Optimizing search engines using clickthrough data,” Proc. Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[8] T. Joachims, “Advances in kernel methods,” chapter Making large-scale support vector machine learning practical, pp.169–184, MIT Press, Cambridge, MA, USA, 1999.

[9] P.M. Long and R.A. Servedio, “Boosting the area under the roc curve,” Advances in Neural Information Processing Systems 20, 2008.

[10] C. Rudin, “Ranking with a p-norm push,” Proc. 19th Annual Conference on Learning Theory, pp.589–604, 2006.

[11] C. Rudin, C. Cortes, M. Mohri, and R.E. Shapire, “Margin-based ranking meets boosting in the middle,” Proc. 18th Annual Conference on Learning Theory, pp.63–78, 2005.

[12] S. Shalev-Shwartz, Y. Singer, and N. Srebro, “Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, 25th International Conference on Machine Learning (ICML), 2008.

[13] 保木邦仁，“局面評価の学習を目指した探索結果の最適制御，” 第 11 回ゲームプログラミングワークショップ，pp.78–83, 2006.

[14] 末廣大貴，畑埜晃平，坂内英夫，瀧本英二，竹田正幸，“カーネル法を用いたコンピュータ将棋の評価関数の学習，” 第 15 回ゲームプログラミングワークショップ，pp.23–27, 2010.

（平成 25 年 7 月 12 日受付，10 月 15 日再受付）

末廣　大貴

2011 年九州大学大学院システム情報科学府情報学専攻修了．現在，同博士後期課程在学中．ランキング学習などの機械学習分野，コンピュータ将棋の研究に従事．

畑埜　晃平　（正員）

2005 年東京工業大学大学院情報理工学研究科数理･計算科学専攻博士課程修了．現在，九州大学大学院システム情報科学研究員情報学部門助教．工学博士．ブースティングやオンライン学習等，機械学習に関する研究に従事．電子情報通信学会，人工知能学会，LA シンポジウム各会員．

坂内　英夫

2000 年東京大学大学院理学系研究科情報科学専攻修士課程修了，同年東京大学医科学研究所助手．現在，九州大学大学院システム情報科学研究院情報学部門准教授．情報理工学博士．文字列データの処理，特に分類・パターン発見アルゴリズムに関する研究に従事．情報処理学会，日本バイオインフォマティクス学会各会員．

瀧本　英二　（正員）

1991 年東北大学大学院工学研究科情報工学専攻博士課程修了．現在，九州大学大学院システム情報科学研究院情報学部門教授．工学博士．計算複雑性理論，計算学習理論，オンライン学習に関する研究に従事．電子情報通信学会，情報処理学会，LA シンポジウム各会員．

竹田　正幸　（正員）

1989 年同大大学院総合理工学研究科情報システム学専攻修了，同年同大工学部電気工学科助手．現在，同教授．工学博士．文字列検索アルゴリズム，データ圧縮に関する研究に従事．電子情報通信学会，人工知能学会，情報処理学会各会員．