

チームSTR

STR アピール文書 2012

評価関数について

従来・・・人間の手作業による調整



- 大変な労力
- 作成者の棋力, 感覚に依存

ボナンザメソッド(保木, 2006)

機械学習による、特徴ベクトルの重みの自動調整

STRの目指すところ

- ✕ 特徴ベクトルの要素は作成者が用意しなければならない
 - やはり作成者の感覚、棋力に依存
- ✕ 実際、現在の強豪プログラムの多くは、将棋の高度な知識が必要とされるような特徴ベクトルを多く用い、学習を行っている

STRの目指すところ

- ✕ 局面の単純な情報のみから、必要な特徴を自動的に作成し、かつパラメータの学習を行う

	特徴ベクトル	パラメータ
従来	人間が用意	手調整
ボナメソ以来	人間が用意	自動調整
STR	自動抽出	自動調整

STRの学習手法

サポートベクターマシンとカーネル法

- ✕ サポートベクターマシン
カーネル法と相性の良い優秀な学習器
- ✕ カーネル法
特徴ベクトルの要素同士の組み合わせから成る
新しい特徴を構成する

SVM (サポートベクトルマシン)

座標上に学習サンプルが与えられる



マージン(超平面と一番近い例との距離)
が最大となるような超平面を作成

未知の例が与えられる



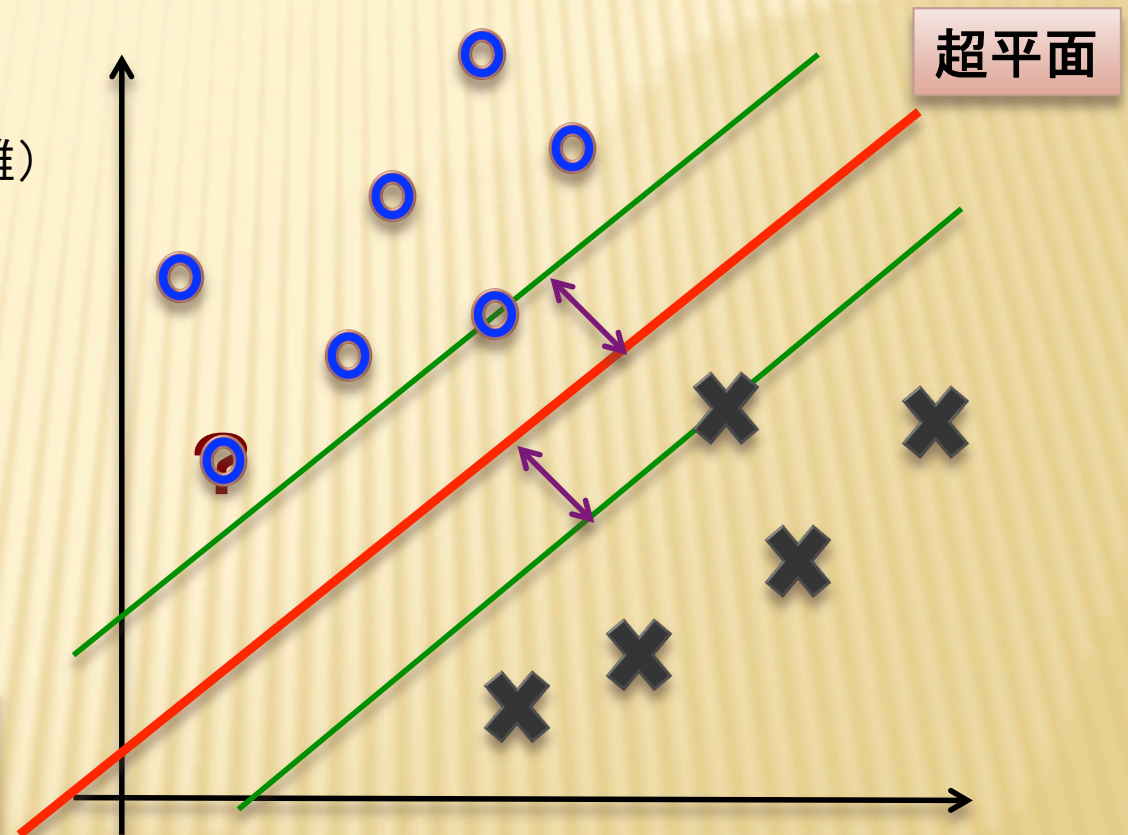
超平面に応じて分類

maximise

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l z_i z_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{y}_j \rangle$$

subject to

$$\sum_i \alpha_i z_i = 0, \alpha_i \geq 0 \quad (i = 1, \dots, l)$$

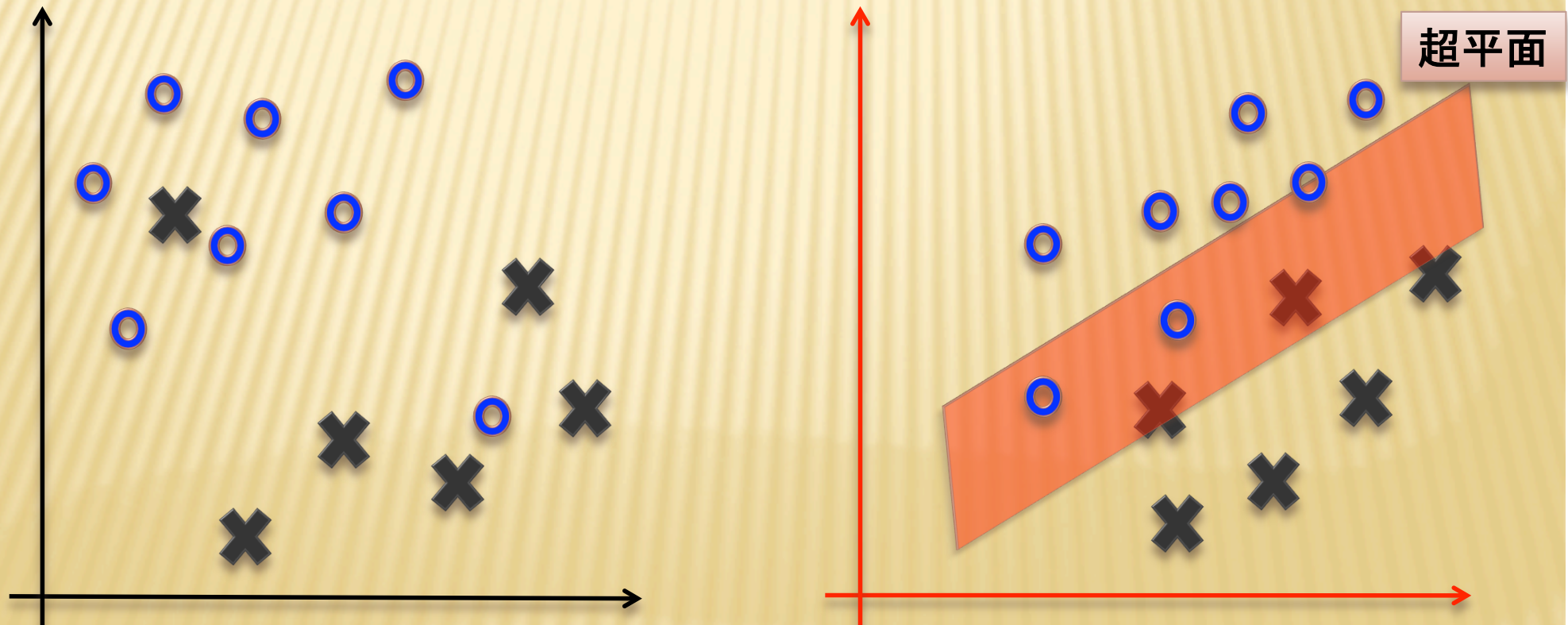


線形分離不可能なデータには性能が不十分

カーネル法

- ✖ データを高次元の特徴空間上へ写像し、SVMを線形分離不可能な問題にも対応させる

$$\mathbf{x} = (x_1, \dots, x_n) \longrightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_n(\mathbf{x}))$$



カーネル法

✕ カーネル関数

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

先ほどの式の $\langle \mathbf{x}_i, \mathbf{y}_j \rangle$ を $K(\mathbf{x}_i, \mathbf{y}_j)$ に置き換えるだけで、特徴空間上でのマージン最大化超平面を求めることが出来る

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l z_i z_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{y}_j)$$

カーネル法に期待するもの

新たな特徴ベクトルの抽出

【1図】

	9	8	7	6	5	4	3	2	1	
										一
										二
										三
										四
										五
			歩	歩						六
	歩	歩	銀	金						七
		玉	金							八
香	桂									九

先手なし

玉 桂 香

例: 駒の位置関係

(金[6七])×(銀[7七])×(金[7八])
(3つの特徴の内積)

探索の代替

【2図】

	9	8	7	6	5	4	3	2	1	
香	桂	銀	金	玉	金	銀	桂	香		一
	飛						角			二
歩	歩	歩	歩	歩	歩		歩	歩	歩	三
						歩				四
										五
		歩	歩							六
歩	歩			歩	歩	歩	歩	歩		七
	角						飛			八
香	桂	銀	金	玉	金	銀	桂	香		九

先手なし

玉 桂 香

一瞬の駒得の6六角を指さないのは、
探索を行うことで悪手だとわかるが、
(6六角)×(6六に駒が利いている)
という特徴の内積でも表現できる

学習手法

- × プロ棋士の棋譜を用意
- × 棋譜の全ての局面に対し
 - + プロが指した手の後の局面（棋譜の次局面）を正例
 - + その他の合法手後の局面を負例とする
- × 全ての局面を単純な特徴ベクトルで表したものを学習サンプルとして、SVMとカーネル法を用いた学習を行う

対局方法

- × 学習で得たSVMの超平面を用意
- × 局面の全ての合法手後の局面の単純な特徴ベクトルを書き出す
- × 超平面と局面のベクトル表現との内積をその局面の評価値とする
- × 最も評価値の高い局面を選択する

STR2012の特徴

× 将棋の盤面をグラフ化

- + グラフ・・・ノードの集合とエッジの集合
- + 盤面の各マスをも81個のノード
- + 利きをエッジ
- + 駒の種類をノードのラベル(ハッシュ値を用いる)

× グラフカーネル

- + 近傍ノードの類似度(将棋で言えば, 利きが利いている先が近傍)により, 正例負例を学習
- + 多項式カーネルより高速?

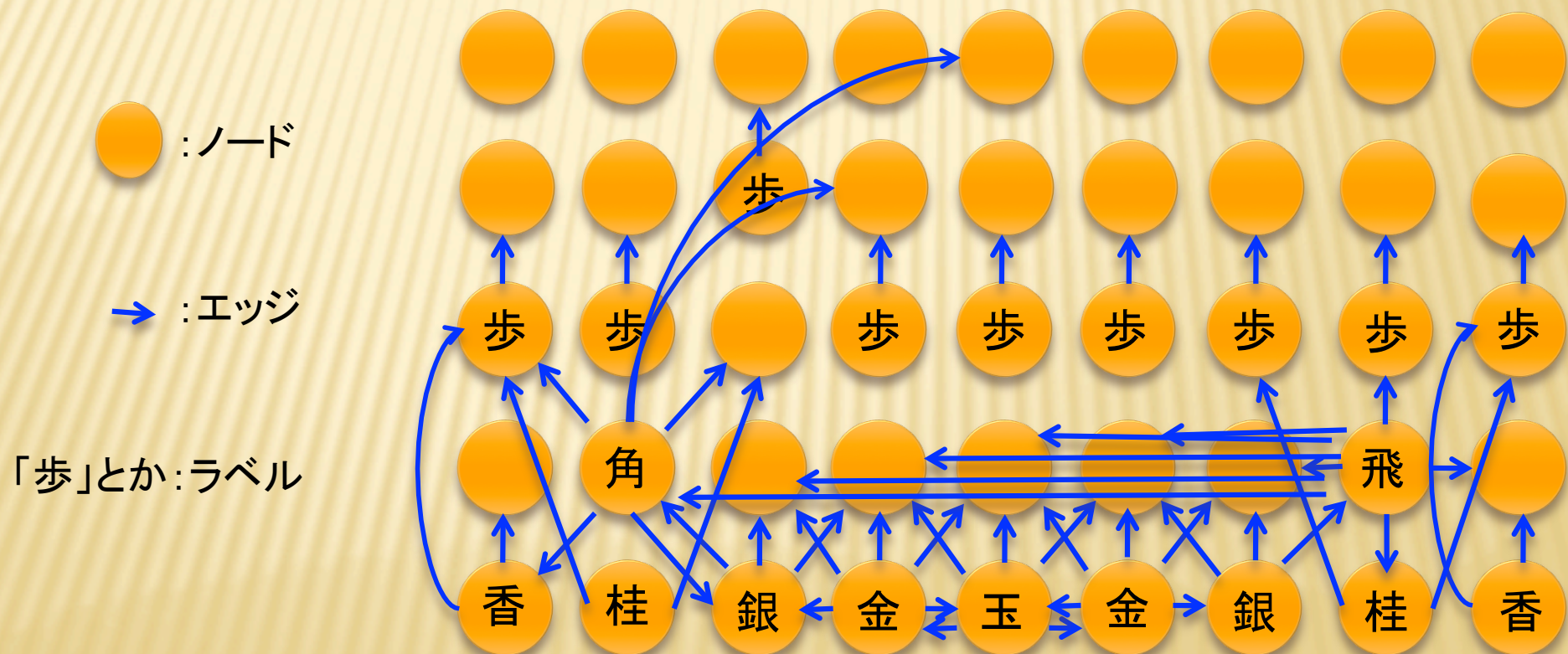
グラフ将棋のイメージ

近傍: 例えば, あるノード v に対して, ノード u からエッジが来ている時,

u は v の近傍とする. (将棋で言えば, 各マスの利かされ情報)

→ 直感的にわかりやすい構造

→ 位置関係の学習に期待



が・・・駄目っ・・・！

学習の失敗

- ✕ ギリギリまで学習を試行錯誤したものの、良い学習結果は得られず
 - + ほぼ、どの局面も同じ評価値となってしまう
 - + 持ち駒の扱いが難しい？（空きマスに利いている？）
 - + サンプルデータが重い（10局で1.3GBほど）

STR2012の妥協と独自性

- ✕ 評価関数を成すサポートベクターは昨年と同じSVMと多項式カーネルで学習したもの
- ✕ SVMの動作が非常に重いので、探索もなし
- ✕ 局面データ構造は一応グラフ構造
- ✕ その他合法手作成などは全てオリジナル
- ✕ GUI は AA 将棋 (New !)
 - + 将棋駒画像およびマス目の画像は <http://mucho.girly.jp/bona/> から、AA 駒の素材は <http://matsucon.net/> からいただきました

✕ 今年も世界最弱の予感