

# Stacking utilizado nos experimentos

Raphael Rodrigues Campos

28 abril, 2016

## Stacking

Stacking também conhecido como “Stacked Generalization” é um método para combinar múltiplos classificadores usando algoritmos de aprendizagem heterogêneos  $L_1, \dots, L_N$  sobre um único conjunto de dados  $D$ , que consiste de exemplos  $e_i = (x_i, y_i)$ , onde  $x_i$  é o vetor de atributos e  $y_i$  sua classificação.

## Stacking Framework

O stacking framework utilizado é baseado no descrito em [1] David H. Wolpert, “Stacked Generalization”, Neural Networks, 5, 241–259, 1992. Foi utilizado um stacking de dois níveis (o framework não se limita a apenas dois níveis, é possível fazer o stacking de quantos níveis julgar necessário), que pode ser dividido em duas fases. Na primeira fase, um conjunto de classificadores do nível base  $C_1, C_2, \dots, C_N$  é gerado, onde  $C_i = L_i(D)$ . Na segunda fase um classificador do meta-nível aprende a combinar as saídas dos classificadores do nível base.

Para gerar o conjunto de treino para o aprendizado do classificador do meta-nível, pode-se aplicar o procedimento **leave-one-out** ou **cross validation**. Por questões óbvias de custo computacional, é utilizado nesse relatório cross validation, mais especificamente **5-fold cross validation**. Cada classificador do nível base aprende usando  $D - F_k$  deixando o  $k$ -ésimo *fold* para teste:  $\forall i = 1, \dots, N : \forall k = 1, \dots, 5 : C_i^k = L_i(D - F_k)$ . Agora, os classificadores recém aprendidos são usados para gerar as predições para  $\forall x_j \in F_k : \hat{y}_j^k = C_i^k(x_j)$ . O conjunto de treino do meta-nível consiste de exemplos da seguinte forma  $((\hat{y}_i^1, \dots, \hat{y}_i^N), y_i)$ , onde os atributos são as predições dos  $N$  classificadores do nível base e a classe é a classe correta sabida de antemão.

**Exemplo** Esse procedimento pode parecer complicado, mas na verdade é simples. Como um exemplo, vamos gerar alguns dados sintéticos com a função “saída = soma dos três componentes de entrada”. Nosso conjunto de treino  $D$  consiste de 5 pares de entrada e saída  $\{((0, 0, 0), 0), ((1, 0, 0), 1), ((1, 2, 0), 3), ((1, 1, 1), 3), ((1, -2, 4), 3)\}$ , todas as entradas sem ruídos. Vamos rotular esses 5 pares de entrada e saída como  $F_1$  até  $F_5$  (Então por exemplo  $D - F_2$  consiste dos quatro pares  $\{((0, 0, 0), 0), ((1, 2, 0), 3), ((1, 1, 1), 3), ((1, -2, 4), 3)\}$ ). Nesse exemplo, temos dois classificadores do nível base  $C_1$  e  $C_2$ , e um único classificador do meta-nível  $\Gamma$ . O conjunto de treino do meta-nível  $D'$  é dado pelo cinco pares de entrada e saída  $\{((C_1^k(F_k), C_2^k(F_k)), \text{componente de saída de } F_k) : \forall k \in \{1, \dots, 5\} \text{ e } C_i^k = L_i(D - F_k)\}$  (Esse espaço do meta-nível possui duas dimensões de entrada e uma de saída). Ou seja, a instância do conjunto de treino do meta-nível correspondente a  $k = 1$  tem o componente de saída 0 e entrada  $(C_1^1((0, 0, 0)), C_2^1((0, 0, 0)))$ . Agora nos é dado um exemplo de teste no formato do nível base  $(x_1, x_2, x_3)$ . Nós predizemos seu valor com  $\Gamma((C_1((x_1, x_2, x_3)), (C_2((x_1, x_2, x_3))))$ , onde  $C_1$  e  $C_2$  foram treinados com todo  $D$ , e  $\Gamma$  com  $D'$ . Em outras palavras, nós predizemos o valor da entrada de teste  $q = (x_1, x_2, x_3)$  treinando  $\Gamma$  em  $D'$  e assim predizendo a entrada formada pelas predições do valor do exemplo de teste  $q$ , de ambos classificadores do nível base  $C_1$  e  $C_2$ , que por suas vezes foram treinados com todo  $D$ .

## Stacking com distribuições de probabilidade

Usar probabilidade para gerar o conjunto de treino do meta-nível é mais vantajoso já que disponibiliza mais informação acerca das predições feitas pelos classificadores do nível base. Essas informações adicionais permitem que não seja usado somente a predição, mas também a confiança de cada classificador do nível base.

Nessa abordagem, cada classificador do nível base prediz uma Distribuição de Probabilidade (DP) sobre todas as classes possíveis. Então, a predição do classificador do nível base  $C$  aplicado a um exemplo  $x$  é a DP:  $p^C(x) = (p^C(c_1|x), \dots, p^C(c_m|x))$ , onde  $\{c_1, \dots, c_m\}$  é o conjunto de possíveis valores para as classes e  $p^C(c_i|x)$  descreve a probabilidade do exemplo  $x$  ser da classe  $c_i$  estimado pelo classificador  $C$ . A classe  $c_j$  com maior probabilidade será classe predita por  $C$ . Dessa forma, os atributos do meta-nível serão as probabilidade preditas para cada classe possível por cada classificador do nível base. O número total de atributos no conjunto de treino do meta-nível seria  $Nm$ ,  $m$  atributos para cada classificador do nível base.

Os experimentos rodados até então utilizaram o stacking framework com DPs.

### Stacking com DP, Entropia e probabilidade máxima

No artigo [2] Is combining classifiers better than selecting the best one, os autores propõem uma extensão para esse framework com DP expandindo o número de meta-atributos. Esse novos meta-atributos seriam:

- A distribuição de probabilidade mutiplicadao pela probabilidade máxima:  $p_{C_j} = p^{C_j}(c_i|x) \times M_{C_j}(x) = p^{C_j}(c_i|x) \times \max_{i=1}^m(p^{C_j}(c_i|x))$ ,  $\forall i \in \{1, \dots, m\}$  e  $\forall j \in \{1, \dots, N\}$ .
- As entropias das distripuições de probabilidade:  $E_{C_j}(x) = - \sum_{i=1}^m p^{C_j}(c_i|x) \cdot \log_2(p^{C_j}(c_i|x))$ .

O número total de atributos do meta-nível é  $N(2m + 1)$ .

A ideia é obter ainda mais informações em relação a predição feita pelos classificadores do nível base. Como Ting and Witten (1999) disseram: o uso de distribuição de probabilidades tem a vantagem de capturar não apenas as predições dos classificadores do nível base, mas também, suas certezas. Os atributos adicionais tentam capturar a certeza de forma mais explícita.

Entropia é uma medida de incerteza. Quanto maior a entropia da distribuição menor é a certeza sobre a predição. A probabilidade máxima de uma DP  $M_{C_j}$  também contém informação sobre certeza da predição: quanto maior  $M_{C_j}$  for mais certo daquela resposta o classificador do nível base está, e vice versa.

Esse é uma ideia para aplicarmos futuramente. Nesse momento continuarei utilizando somente a DP.