`multiboost.org`: an implementation of cost-sensitive multi-class/multi-label AdaBoost.MH

Róbert Busa-Fekete[1], Norman Casagrande[2], and Balázs Kégl[1]

[1]University of Paris-Sud / CNRS
[2] last.fm

ICML'10 open source workshop
June 25, 2010

Position slide

reviewer 1: “using the program is marginally faster and much less instructive than writing your own version of these simple algorithms”

- Simple?
  - yes: binary AdaBoost with decision stumps is simple, but an average biologist will never sit down to implement it
  - no: implementing full cost-sensitive multi-class/multi-label AdaBoost.MH[1] with decision trees/products/Haar filters is not simple even for an average computer scientist
  - WEKA contains a very suboptimal multi-class AdaBoost implementation, other than that there is no widely used implementation that we know of

[1] Schapire, R. E. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, 1999.

A “simple” base learner for nominal features

IndicatorBase($\mathbf{X}, \mathbf{Y}, \mathbf{W}$)

for $j \leftarrow 1$ **to** d ▷ *all (nominal) features*
  $(\alpha_j, \mathbf{v}_j, \mathbf{u}_j) \leftarrow$ BestIndicator$(\mathbf{x}^{(j)}, \mathbf{Y}, \mathbf{W}, I^{(j)})$ ▷ $\mathbf{x}^{(j)} \triangleq (x_1^{(j)}, \ldots, x_n^{(j)})$
$j^* \leftarrow \arg\min_j E\big(\alpha_j \mathbf{v}_j \varphi_{j,\mathbf{u}_j}, \mathbf{W}\big)$
return $(\alpha_{j^*}, \mathbf{v}_{j^*}, \varphi_{j^*,\mathbf{u}_{j^*}}(\cdot))$

BestIndicator($\mathbf{x}, \mathbf{Y}, \mathbf{W}, I$)

for $\iota \in I$
  for $\ell \leftarrow 1$ **to** K
    $\gamma^+_{\iota,\ell} \leftarrow \gamma^-_{\iota,\ell} \leftarrow 0$
  $u_\iota \leftarrow$ Random(± 1)
for $i \leftarrow 1$ **to** n **for** $\ell \leftarrow 1$ **to** K
  if $w_{i,\ell} y_{i,\ell} > 0$ **then**
    $\gamma^+_{x_i,\ell} \leftarrow \gamma^+_{x_i,\ell} + w_{i,\ell} y_{i,\ell}$
  else
    $\gamma^-_{x_i,\ell} \leftarrow \gamma^-_{x_i,\ell} - w_{i,\ell} y_{i,\ell}$
$\alpha \leftarrow 0$, $\mathbf{v} \leftarrow \mathbf{0}$
while TRUE
  $\alpha_{\text{prev}} \leftarrow \alpha$, $\mathbf{v}_{\text{prev}} \leftarrow \mathbf{v}$ ▷ *save current optimal* α *and* $\mathbf{v}$
  for $\ell \leftarrow 1$ **to** K
    $v_\ell \leftarrow \operatorname{sign}\Big(\sum_{\iota \in I} (\gamma^+_{\iota,\ell} - \gamma^-_{\iota,\ell}) u_\iota\Big)$ or $v_\ell \leftarrow \frac{1}{2} \ln \frac{\sum_{\iota \in I}\big(\gamma^+_{\iota,\ell}\mathbb{I}\{u_\iota > 0\} + \gamma^-_{\iota,\ell}\mathbb{I}\{u_\iota < 0\}\big)}{\sum_{\iota \in I}\big(\gamma^-_{\iota,\ell}\mathbb{I}\{u_\iota > 0\} + \gamma^+_{\iota,\ell}\mathbb{I}\{u_\iota < 0\}\big)}$
  $\alpha \leftarrow \frac{1}{2} \ln \frac{\sum_{\ell=1}^K \sum_{\iota \in I}\big(\gamma^+_{\iota,\ell}\mathbb{I}\{u_\iota > 0\} + \gamma^-_{\iota,\ell}\mathbb{I}\{u_\iota < 0\}\big)}{\sum_{\ell=1}^K \sum_{\iota \in I}\big(\gamma^-_{\iota,\ell}\mathbb{I}\{u_\iota > 0\} + \gamma^+_{\iota,\ell}\mathbb{I}\{u_\iota < 0\}\big)}$ or $\alpha \leftarrow 1$
  if $E(\alpha \mathbf{v} \varphi_{\mathbf{u}}, \mathbf{W}) \geq E(\alpha_{\text{prev}} \mathbf{v}_{\text{prev}} \varphi_{\mathbf{u}}, \mathbf{W})$ **then**
    return $(\alpha_{\text{prev}}, \mathbf{v}_{\text{prev}}, \mathbf{u})$
  $\alpha_{\text{prev}} \leftarrow \alpha$, $\mathbf{u}_{\text{prev}} \leftarrow \mathbf{u}$ ▷ *save current optimal* α *and* $\mathbf{u}$
  for $\iota \in I$
    $u_\iota \leftarrow \operatorname{sign}\Big(\sum_{\ell=1}^K (\gamma^+_{\iota,\ell} - \gamma^-_{\iota,\ell}) v_\ell\Big)$ or $u_\iota \leftarrow \frac{1}{2} \ln \frac{\sum_{\ell=1}^K\big(\gamma^+_{\iota,\ell}\mathbb{I}\{v_\ell > 0\} + \gamma^-_{\iota,\ell}\mathbb{I}\{v_\ell < 0\}\big)}{\sum_{\ell=1}^K\big(\gamma^-_{\iota,\ell}\mathbb{I}\{v_\ell > 0\} + \gamma^+_{\iota,\ell}\mathbb{I}\{v_\ell < 0\}\big)}$
  $\alpha \leftarrow \frac{1}{2} \ln \frac{\sum_{\iota \in I} \sum_{\ell=1}^K \big(\gamma^+_{\iota,\ell}\mathbb{I}\{v_\ell > 0\} + \gamma^-_{\iota,\ell}\mathbb{I}\{v_\ell < 0\}\big)}{\sum_{\iota \in I} \sum_{\ell=1}^K \big(\gamma^-_{\iota,\ell}\mathbb{I}\{v_\ell > 0\} + \gamma^+_{\iota,\ell}\mathbb{I}\{v_\ell < 0\}\big)}$ or $\alpha \leftarrow 1$
  if $E(\alpha \mathbf{v} \varphi_{\mathbf{u}}, \mathbf{W}) \geq E(\alpha_{\text{prev}} \mathbf{v} \varphi_{\mathbf{u}_{\text{prev}}}, \mathbf{W})$ **then**
    return $(\alpha_{\text{prev}}, \mathbf{v}, \mathbf{u}_{\text{prev}})$

Cost sensitive multi-label/multi-class

- Input vector $\mathbf{x} \in \mathcal{X}^d$: $x^{(j)}$ is either real-valued or nominal
- Label vector $\mathbf{y} \in \{-1, 1\}^K$
- Cost (initial weight) vector $\mathbf{w}^{(1)} \in \left(\mathbb{R}^+ \cup \{0\}\right)^K$
  - for example, classical multi-class

$$w_{i,\ell}^{(1)} = \begin{cases} \frac{1}{2n} & \text{if } \ell \text{ is the correct class of } \mathbf{x}_i \text{ (if } y_{i,\ell} = 1\text{),} \\ \frac{1}{2n(K-1)} & \text{otherwise (if } y_{i,\ell} = -1\text{).} \end{cases}$$

Cost sensitive multi-label/multi-class

- AdaBoost.MH learns a vector-valued discriminant function

$$\mathbf{f}(\mathbf{x}) : \mathcal{X}^d \to \mathbb{R}^K$$

by minimizing the weighted Hamming loss

$$R\left(\mathbf{f}^{(T)}, \mathbf{w}^{(1)}\right) = \sum_{i=1}^{n} \sum_{\ell=1}^{K} w_{i,\ell}^{(1)} \mathbb{I}\left\{ f^{(\ell)}(\mathbf{x}_i) y_{i,\ell} < 0 \right\}$$

- Strong learners
  - AdaBoost.MH[2]
  - FeatureBoost[3]
- Weak learners
  - Desision stump for real-valued features
  - Selector and subset indicator[4] for nominal features
  - Haar filter[5] for image input
  - Decision tree and decision product[4] for combining simple base-classifiers
  - Easy to add new base learners without affecting the main boosting engine

---

[2] Schapire, R. E. and Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, 1999.
[3] Bradley, J.K. and Schapire, R.E. FilterBoost: Regression and classification on large datasets. In *NIPS*, volume 20, 2008.
[4] Kégl, B. and Busa-Fekete, R. Boosting products of base classifiers. In *ICML*, volume 26, pp. 497–504, 2009.
[5] Viola, P. and Jones, M. Robust real-time face detection. *Int. J. of Comp. Vis.*, 57:137–154, 2004.

- Other features
  - Bandit boosting[6]: adaptive feature selection to accelerate training
  - Efficient multi-platform C++ implementation
  - Training and test data are input in ARFF format
  - Support for sparse data and/or label matrix
  - The classifiers are saved in XML format
  - Test/training error and other iteration-wise statistics are saved in a tab separated data file

[6] Busa-Fekete, R. and Kégl, B. Boosting products of base classifiers. In *ICML*, volume 27, 2010.

Bechmark and challenge results

- MIREX'05
  www.music-ir.org/evaluation/mirex-results
  winner in genre classification track and runner up in the artist Identification track[7]

- MNIST
  yann.lecun.com/exdb/mnist
  Boosting decision product of stumps[8] is the best reported no-domain-knowledge algorithm on MNIST after Hinton and Salakhutdinov's deep belief nets

- Yahoo - Learning to Rank Challenge (ICML'10 workshop)
  learningtorankchallenge.yahoo.com
  6th place in track 1 and 11th place in track 2. The difference between our calibrated AdaBoost.MH approach and the winners was less then 0.003 in both tracks.

---

[7] J. Bergstra et al. , "Aggregate features and AdaBoost for music classification," *Machine Learning Journal*, vol. 65, no. 2/3, pp. 473–484, 2006.
[8] Kégl, B. and Busa-Fekete, R. Boosting products of base classifiers. In *ICML*, volume 26, pp. 497–504, 2009.