Highly Scalable Discriminative Spam Filtering

Michael Brückner

Why?

- 3.2 billion email accounts.
  - 70% spam & phishing emails; >10% undetected.
- 2.4 billion social networking accounts.
- 2.6 billion IM accounts.
- 1 billion WhatsApp messages per day.

Outline

- Problem Statement
  - Discriminative Classifiers
- Data Provisioning & Preprocessing
  - Text Feature Extraction
  - Hashing Trick
- Large-scale Learning Algorithms
  - Parallelized Stochastic Gradient Descent
  - (Alternatives?)

Problem Statement

- Given:
  - Sample of n messages with class labels $y_i = +1$ (spam) or $y_i = -1$ (non-spam).
  - Feature mapping: message → feature vector $x_i \in \mathrm{R}^m$.
- Objective:
  - Decision function f so that

$$y = \mathrm{sign}\, f(x)$$

holds for future messages.

Discriminative Classifier

Approach: Find function f which minimizes

- Empirical classification error (loss):

$$R[f] = \Sigma_i \, l(y_i, f(x_i))$$

<u>and</u>

- Model complexity (regularizer):

$$\Omega[f] = \|f\|_H$$

Linear Discriminative Classifier

Solve for linear decision function $f(x) = <x, w>$:

$$\min_f \ \Sigma_i \ l(y_i, f(x_i)) + \lambda\Omega[f]$$

Model	*Loss function l*	*Regularizer Ω*
Perceptron	$\max(0, -y\, f(x))$	-
SVM	$\max(0, 1 - y\, f(x))$	$1/2 \|\|w\|\|_2^2$
Logistic Reg.	$\log(1 + \exp(-y\, f(x)))$	$1/2 \|\|w\|\|_2^2$
Linear Reg.	$(y - f(x))^2$	$1/2 \|\|w\|\|_2^2$
Lasso	$(y - f(x))^2$	$\|\|w\|\|_1$

Data Provisioning

- Task: Continuously collection of labeled data.
- Spam messages:
  - From blacklisted IPs, honeypots, user reports, etc.
- Non-spam messages:
  - Public sources (newsletters, moderated groups & bulletin boards, Enron email corpus etc.)
  - Two-way communication.
- Distributed storage.

Text Feature Extraction

- Word-based feature extraction:
  - Parsing & tokenization (e.g. using Apache OpenNLP, Lucene).
  - No stemming.
- Character-based feature extraction:
  - Character n-grams (shingles).

Text Feature Extraction

- Feature mapping:
  - Binary Bag of Words, Orthogonal Sparse Bigrams, Sparse Binary Polynomial Hashing.
  - No term frequencies (TF) or TF-IDF.
  - L2-Normalization per instance:

$$x' = \frac{1}{\|x\|_2} x$$

  - No explicit feature selection.

Hashing Trick

- Multiple binary features are combined to one.
- Example:

```
You have exceeded the storage limit ...
 6   3       5        1      9      8
```

- Binary Bag of Word representation:

$$x = [1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0]^{\mathrm{T}}$$

Hashing Trick

`You have exceeded the storage limit ...`

You	have	exceeded	the	storage	limit	...
6	3	5	1	9	8	

- Feature mapping for Binary Bag of Word:

$$x = \varphi(\text{message}) = \begin{bmatrix} \text{contains } \texttt{the}? \\ \text{contains } \texttt{hello}? \\ \text{contains } \texttt{have}? \\ \text{contains } \texttt{world}? \\ \text{contains } \texttt{exceeded}? \\ \text{contains } \texttt{You}? \\ \ldots \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ \ldots \end{bmatrix}$$

Hashing Trick

You have exceeded the storage limit ...

6 3 5 1 9 8

- Feature mapping with hashing:

$$x = \varphi(\text{message}) = \begin{bmatrix} \text{contains } \texttt{the} & \text{or } \texttt{You}? \\ \text{contains } \texttt{hello} & \text{or } \texttt{limit}? \\ \text{contains } \texttt{have} & \text{or } \texttt{storage}? \\ \text{contains } \texttt{world} & \text{or } \texttt{young}? \\ \text{contains } \texttt{exceeded} & \text{or } \texttt{me}? \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Large-scale Learning Algorithms

- Distributed & iterative methods.
- Given are n message-label pairs (x_i, y_i).
- Objective function for $f_w(x) = \langle x, w \rangle$:

$$c(w) = \sum_{i=1}^{n} l(y_i, \langle x_i, w \rangle) + \square \frac{1}{2} \|w\|_2^2$$

- Solve for w of decision function f_w:

$$\min_{w} c(w)$$

Gradient Descent

- Gradient: $$c'(w) = \sum_{i=1}^{n} l'(y_i, \langle x_i, w \rangle) x_i + \square w$$
- Example SVM: $$l'(y, z) = \begin{cases} 0 & if \; yz \geq 1 \\ -y & if \; yz < 1 \end{cases}$$
- Initialize w to all-zeros vector.
- Repeat until w converged:

$$w \leftarrow w - \square \, c'(w)$$

Stochastic Gradient Descent (SGD)

- Stochastic Gradients: $c_i{}'(w) = l{}'(y_i, \langle x_i, w \rangle)\, x_i + \frac{\lambda}{n} w$

  where $c{}'(w) = \sum_{i=1}^{n} c_i{}'(w)$

- Initialize w to all-zeros vector.
- Repeat until w converged:
  - Randomly draw $i \in \{1, \ldots, n\}$ and update

$$w \leftarrow w - \eta\, c_i{}'(w)$$

Parallelized Stochastic Gradient Descent

- Randomly distribute n message-label pairs.
  - $T > n / N$ pairs to each of the N nodes.
- SGD on each machine j to compute $w^{(j)}$.
- Average the N computed weight vectors:

$$w = \frac{1}{N} \sum_{j=1}^{N} w^{(j)}$$

Alternatives:
Bayes Point Machine

- Given: N nodes, every node has access to all data.
- Large-scale Bayes Point Machine:
  - For each node: solve Perceptron with random order of training data.
  - Average (normalized) Perceptron solutions:

$$w = \frac{1}{N} \sum_{j=1}^{N} \frac{w^{(j)}}{\left\| w^{(j)} \right\|_2}$$

Alternatives: Consensus Propagation

- Given: N weakly connected nodes, potentially non-randomly distributed data.
- Idea: Decompose minimization problem into coupled sub-problems $j = 1, \ldots, N$.

$$min \sum_{i \in S^{(j)}} l\left(y_i, \langle x_i, w^{(j)} \rangle\right) + \square \frac{1}{2N} \|w^{(j)}\|_2^2$$

$$s.t. \quad w^{(j)} = w^{(k)} \quad \forall k \in neighbors\ of\ node\ j$$

Summary

- As many data as possible for training with as many attributes as possible.
- Implicit feature reduction by hashing trick.
- Large-scale discriminative classifier based on (parallelized) stochastic gradient descent.

References

- Joshua Attenberg et al. *Collaborative Email-Spam Filtering with the Hashing-Trick.* CEAS, 2009.
- Aditya Krishna Menon. *Large-Scale Support Vector Machines: Algorithms and Theory*. 2009.
- Martin Zinkevich et al. *Parallelized Stochastic Gradient Descent.* NIPS, 2010.
- Ralf Herbrich and Thore Graepel. *Large Scale Bayes Point Machines*. NIPS, 2000.
- Pedro A. Forero et al. *Consensus-Based Distributed Support Vector Machines.* JMLR, 2010.
- Neal Parikh and Stephen Boyd. *Graph Projection Block Splitting for Distributed Optimization.* 2012.

Alternatives: Block Splitting

- Given: N nodes, data is heavily distributed.
  - E.g. attributes of messages are distributed.
- Idea:
  - Block-wise decomposition of the minimization problem.
  - Applying *alternating direction method of multipliers* (ADMM).