Extreme Classification

Tighter Bounds, Distributed Training, and new Algorithms

Marius Kloft (HU Berlin)

Krakow, Feb 16, 2017

Joint work with Urun Dogan (Microsoft Research), Yunwen Lei (CU Hong Kong), Maximilian Albers (Berlin Big Data Center), Julian Zimmert (HU Berlin), Moustapha Cisse (Facebook AI Lab), Alexander Binder (Singapore), and Rohit Babbar (MPI Tübingen)
**What is Multi-class Classification?**

*Multiclass classification* is, given a data point $x$, decide on the class with which the data point is annotated.
What is Extreme Classification?

**Extreme classification** is multi-class classification using an extremely large amount of classes.
Example 1

We are continuously monitoring the internet for new webpages, which we would like to categorize.
Example 2

We have data from an online biomedical bibliographic database that we want to index for quick access to clinicians.
Example 3

We are collecting data from an online feed of photographs that we would like to classify into image categories.
Example 4

We add new articles to an online encyclopedia and intend to predict the categories of the articles.
Need for theory and algorithms for extreme classification.
How do algorithms and bounds scale in #classes?
How do algorithms and bounds scale in \#classes?
1 Introduction

2 Distributed Algorithms

3 Theory

4 Learning Algorithms

5 Conclusion
Support Vector Machine (SVM) is a Popular Method for Binary Classification (Cortes and Vapnik, ’95)

Core idea:
- Which hyperplane to take?
Support Vector Machine (SVM) is a Popular Method for Binary Classification

- Which hyperplane to take?
- The one that separates the data with the largest margin
Popular Generalization to Multiple Classes: One-vs.-Rest SVM

Put \( C := \# \text{classes} \).

One-vs.-rest SVM

1. For \( c = 1..C \)
2. class1 := \( c \), class2 := \( \text{union(allOtherClasses)} \)
3. \( w_c := \text{solutionOfSVM(class1, class2)} \)
4. end
5. Given a test point \( x \), predict \( c_{\text{predicted}} := \arg \max_c w_c^T x \)
Runtime of One-vs.-Rest

... assuming sufficient computational resources (#classes many computers)
Problem With One-vs.-Rest

:) training **can be parallelized** in the number of classes (extreme classification!)

:( Is just a hack. One-vs.-Rest SVM is not built for multiple classes (coupling of classes not exploited)!
There are “True” Multi-class SVMs, So-called **All-in-one** Multi-class SVMs

- **binary:**
  - Lin, Lee, and Wahba ('04)
  - Watkins and Weston ('99)
  - Crammer and Singer ('02)

- **MC:**
  - Lin, Lee, and Wahba ('04)
  - Watkins and Weston ('99)
  - Crammer and Singer ('02)
There are “True” Multi-class SVMs, So-called **All-in-one** Multi-class SVMs

Problem: State of the art solvers require a training time complexity of $O(dn \cdot C)$, where $d = \text{dim}$, $n = \#\text{examples}$, and $C = \#\text{classes}$.
Aim: Develop algorithms where $O(C)$ machines in parallel and in $O(dn)$ runtime train all-in-one MC-SVMs.

$\Rightarrow$ same time complexity as one-vs.-rest, yet more sophisticated algorithm
All-in-one SVMs

All of them have in common that they minimize a trade-off of a regularizer and a loss term:

$$\min_{w=(w_1,\ldots,C)} \frac{1}{2} \sum_c \|w_c\|^2 + C \cdot L(w, \text{data})$$
All-in-one SVMs

All of them have in common that they minimize a trade-off of a regularizer and a loss term:

$$\min_{w=(w_1,\ldots,c)} \frac{1}{2} \sum_{c} \|w_c\|^2 + C \ast L(w, \text{data})$$
All Three MC-SVMs have:

\[ \min_{\mathbf{w}=\left(\mathbf{w}_1, \ldots, \mathbf{w}_C\right)} \frac{1}{2} \sum_c \| \mathbf{w}_c \|^2 + C^* \ldots \]
All Three MC-SVMs have:

$$\min_{w=(w_1,\ldots,w_C)} \frac{1}{2} \sum_c \|w_c\|^2 + C*$$

But they differ in the loss:

CS:

$$\ldots \sum_{i=1}^{n} \left[ \max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right]$$

note: $$l(x) := \max(0, 1 - x)$$
All Three MC-SVMs have:

\[
\min_{w=(w_1,...,w_C)} \frac{1}{2} \sum_c \|w_c\|^2 + C* \quad ...
\]

But they differ in the loss:

CS:

\[
\ldots \sum_{i=1}^{n} \left[ \max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right]
\]

note: \( l(x) := \max(0, 1 - x) \)
All Three MC-SVMs have:

\[ \min_{w=(w_1,\ldots,w_C)} \frac{1}{2} \sum_c \|w_c\|^2 + C^* \ldots \]

But they differ in the loss:

**CS:**
\[ \sum_{i=1}^{n} \left[ \max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right] \]

**WW:**
\[ \sum_{i=1}^{n} \left[ \sum_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right] \]

Note: \( l(x) := \max(0, 1 - x) \)

All Three MC-SVMs have:

\[ \min_{w=(w_1,\ldots,w_C)} \frac{1}{2} \sum_c \|w_c\|^2 + C^* \quad \ldots \]

But they differ in the loss:

**CS:**  \[ \ldots \sum_{i=1}^{n} \left[ \max_{c \neq y_i} l\left((w_{y_i} - w_c)^T x_i\right) \right] \]

**WW:**  \[ \ldots \sum_{i=1}^{n} \left[ \sum_{c \neq y_i} l\left((w_{y_i} - w_c)^T x_i\right) \right] \]

**LLW:**  \[ \ldots \sum_{i=1}^{n} \left[ \sum_{c \neq y_i} l(-w_c^T x_i) \right], \text{ s.t. } \sum_c w_c = 0 \]

\[ l(x) := \max(0, 1 - x) \]

Can we solve these all-in-one MC-SVMs in parallel?
Can we solve these all-in-one MC-SVMs in parallel?

Let’s look at Lee, Lin, and Wahba (LLW) first.
This is the LLW **Dual** Problem

\[
\begin{align*}
&\max_{\alpha} \quad -\frac{1}{2} \sum_{c=1}^{C} \|X\alpha_c - \frac{1}{C} \sum_{\tilde{c}} X\alpha_{\tilde{c}}\|^2 + \sum_{c,i:y_i=c} \alpha_i \\
\text{s.t.} \quad &\alpha_{i,y_i} = 0 \\
&0 \leq \alpha_{i,c} \leq C
\end{align*}
\]
This is the LLW **Dual** Problem

\[
\max_{\alpha} \max_{\vec{w}} - \frac{1}{2} \sum_{c=1}^{C} \|X\alpha_c - \frac{1}{\bar{c}} \sum_{\tilde{c}} X\alpha_{\tilde{c}} \|^2 + \sum_{c,i:y_i=c} \alpha_i \\
\text{s.t. } \alpha_{i,y_i} = 0 \\
0 \leq \alpha_{i,c} \leq C
\]
This is the LLW **Dual** Problem

\[
\max_{\alpha, \bar{w}} \sum_c \left[ -\frac{1}{2} ||X\alpha_c - \bar{w}||^2 + \sum_{i:y_i=c} \alpha_i \right]
\]

s.t. \( \alpha_{i,y_i} = 0 \)

\( 0 \leq \alpha_{i,c} \leq C \)
LLW: Proposed Algorithm

LLW: Proposed Algorithm

**Algorithm**  Simple wrapper algorithm

1:  function \textsc{SimpleSolve-LLW}(C, X, Y)
2:   while not converged do
3:     for \( c = 1 \ldots C \) do in parallel
4:       \( \alpha_c \leftarrow \arg \max_{\tilde{\alpha}_c} D_c(\tilde{\alpha}_c, \bar{w}) \)
5:     end for
6:     \( \bar{w} \leftarrow \arg \max_w D(\alpha, w) \)
7:   end while
8:  end function

Alber, Zimmert, Dogan, and Kloft (2016):  
NIPS submitted rejected ;)


**LLW: Proposed Algorithm**

**Algorithm**  Simple wrapper algorithm

1. **function** `SIMPLESOLVE-LLW(C, X, Y)`
2. **while** not converged **do**
3. **for** $c = 1..C$ **do** in parallel
4. \[ \alpha_c \leftarrow \arg \max_{\tilde{\alpha}_c} D_c(\tilde{\alpha}_c, \bar{w}) \]
5. **end for**
6. \[ \bar{w} \leftarrow \arg \max_w D(\alpha, w) \]
7. **end while**
8. **end function**

Alber, Zimmert, Dogan, and Kloft (2016):
NIPS submitted rejected ;)
Ok, fine so far with the LLW SVM. Now, let’s look at the *Weston and Watkins (WW)* SVM.
**WW: This is How the **Dual** Problem Looks Like**

$$\max_{\alpha \in \mathbb{R}^{n \times C}} \sum_{c=1}^{C} \left[ -\frac{1}{2} \| -X\alpha_c \|^2 + \sum_{i:y_i \neq c} \alpha_{i,c} \right]$$

s.t. \( \forall i: \alpha_{i,y_i} = -\sum_{c:c \neq y_i} \alpha_{i,c} \),

\( \forall c \neq y_i: 0 \leq \alpha_{i,c} \leq C \)
**WW:** This is How the Dual Problem Looks Like

\[
\max_{\alpha \in \mathbb{R}^{n \times C}} \sum_{c=1}^{C} \left[ -\frac{1}{2} ||X\alpha_c||^2 + \sum_{i: y_i \neq c} \alpha_{i,c} \right] =: D(\alpha)
\]

s.t. \( \forall i : \alpha_{i,y_i} = - \sum_{c:c \neq y_i} \alpha_{i,c} \),

\( \forall c \neq y_i : 0 \leq \alpha_{i,c} \leq C \)

A common strategy to optimize such a dual problem, is to optimize one coordinate after another ("dual coordinate ascent"):

1. for \( i = 1, \ldots, n \)
2. for \( c = 1, \ldots, C \)
3. \( \alpha_{i,c} = \max_{\alpha_{i,c}} D(\alpha) \)
4. end
5. end
This is Now the Story...

We optimize $\alpha_{i,c}$ into gradient direction:

$$\frac{\partial}{\partial \alpha_{i,c}} : 1 - (w_y - w_c)^T x_i$$

Derivative depends only on two weight vectors (not all $C$ many!).
This is Now the Story...

We optimize $\alpha_{i,c}$ into gradient direction:

$$\frac{\partial}{\partial \alpha_{i,c}} : 1 - (w_{yi} - w_c)^T x_i$$

Derivative depends only on **two** weight vectors (not all $C$ many!).

Can we exploit this?
Analogy: Soccer League Schedule

We are given a football league (e.g., Bundesliga) with $C$ many teams. Before the season, we have to decide on a schedule such that each team plays any other team exactly once. Furthermore, all teams shall play on every matchday so that in total we need only $C - 1$ matchdays.

Example

Bundesliga has $C = 18$ teams.

$\Rightarrow C - 1 = 17$ matchdays (or twice that many if counting home and away matches)
Analogy: Soccer League Schedule

We are given a football league (e.g., Bundesliga) with $C$ many teams. Before the season, we have to decide on a schedule such that each team plays any other team exactly once. Furthermore, all teams shall play on every matchday so that in total we need only $C - 1$ matchdays.

Example

Bundesliga has $C = 18$ teams.

$\Rightarrow C - 1 = 17$ matchdays (or twice that many if counting home and away matches)

How can we come up with a schedule?
This is a Classical Computer Science Problem...

This is the 1-factorization of a graph problem.
This is a Classical Computer Science Problem...

This is the $1$-factorization of a graph problem. The solution is known:

Here: $C = 8$ many teams, 7 matchdays
**WW: Proposed Algorithm**

**Algorithm**  Simplistic DBCA wrapper algorithm

```plaintext
1: function SIMPLESOLVE-WW(C, X, Y)
2:     while not converged do
3:         for r = 1...C − 1 do  # iterate over “matchdays”
4:             for c = 1..C/2 do in parallel  # iterate over “matches”
5:                 (c_i, c_j) ← the two classes (“opposing teams”)
6:                 α_{I_{c_i,c_j}} ← arg max_{α_1,α_2} D_c(α_1, α_2)
7:             end for
8:         end for
9:     end while
10: end function
```

Accuracies

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Training</th>
<th># Test</th>
<th># Classes</th>
<th># Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>98,200</td>
<td>10,800</td>
<td>1000</td>
<td>128</td>
</tr>
<tr>
<td>LSHTCsmall</td>
<td>4,463</td>
<td>1,858</td>
<td>1,139</td>
<td>51,033</td>
</tr>
<tr>
<td>DMOZ2010</td>
<td>128,710</td>
<td>34,880</td>
<td>12,294</td>
<td>381,581</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OVR</th>
<th>CS</th>
<th>WW</th>
<th>LLW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALOI</td>
<td>0.1824</td>
<td>0.0974</td>
<td>0.0930</td>
<td>0.6560</td>
</tr>
<tr>
<td>LSHTCsmall</td>
<td>0.549</td>
<td>0.5919</td>
<td>0.5505</td>
<td>0.9263</td>
</tr>
<tr>
<td>DMOZ2010</td>
<td>0.5721</td>
<td>-</td>
<td>0.5432</td>
<td>0.9586</td>
</tr>
</tbody>
</table>

**Table:** Datasets used in our paper, their properties and best test error over a grid of $C$ values.
Results: Speedup

![Graph 1: LLW: Number of Nodes vs. Speedup]

- LLW:dmoz_2010
- LLW:aloi

![Graph 2: WW: Number of active Nodes vs. Speedup]

- WW:dmoz_2010
- WW:aloi
Open questions

- higher efficiencies via GPUs?
- Why does LLW accuracy break?
- parallelization for CS?
Introduction

Distributed Algorithms

Theory

Learning Algorithms

Conclusion
Theory and Algorithms in Extreme Classification

▷ Just saw: **Algorithms** that better handle large number of classes
Theory and Algorithms in Extreme Classification

- **Theory** not prepared for extreme classification
  - Data-dependent bounds scale at least linearly with the number of classes
    (Koltchinskii and Panchenko, 2002; Mohri et al., 2012; Kuznetsov et al., 2014)
Theory of Extreme Classification

Questions

▶ Can we get bounds with mild dependence on #classes?
⇒ Novel algorithms?
Multi-class Classification

Given:

- Training data \(\mathbf{z}_1 = (x_1, y_1), \ldots, \mathbf{z}_n = (x_n, y_n)\) \(\overset{i.i.d.}{\sim} P\) \(\in \mathcal{X} \times \mathcal{Y}\)
- \(\mathcal{Y} := \{1, 2, \ldots, C\}\)
- \(C = \) number of classes

![Diagram of various objects]
Formal Problem Setting

**Aim:**

- Define a hypothesis class $H$ of functions $h = (h_1, \ldots, h_c)$
- Find an $h \in H$ that “predicts well” via

\[
\hat{y} := \arg \max_{y \in Y} h_y(x)
\]

**Multi-class SVMs:**

- $h_y(x) = \langle w_y, \phi(x) \rangle$
- Introduce notion of the **(multi-class) margin**

\[
\rho_h(x, y) := h_y(x) - \max_{y' : y' \neq y} h_{y'}(x)
\]

- the larger the margin, the better

**Want:** large expected margin $\mathbb{E} \rho_h(X, Y)$. 
Types of Generalization bounds for Multi-class Classification

**Data-independent bounds**
- based on covering numbers
  
  (Guermeur, 2002; Zhang, 2004a,b; Hill and Doucet, 2007)
  - conservative
    - unable to adapt to data

**Data-dependent bounds**
- based on Rademacher complexity
  
  (Koltchinskii and Panchenko, 2002; Mohri et al., 2012; Cortes et al., 2013; Kuznetsov et al., 2014)
  + tighter
    - able to capture the real data
    - computable from the data
Def.: Rademacher and Gaussian Complexity

- Let \( \sigma_1, \ldots, \sigma_n \) be independent Rademacher variables (taking only values \( \pm 1 \), with equal probability).

- The **Rademacher complexity** (RC) is defined as
  \[
  \mathcal{R}(H) := \mathbb{E}_\sigma \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(z_i) \right]
  \]

- Let \( g_1, \ldots, g_n \sim N(0, 1) \).

- The **Gaussian complexity** (GC) is defined as
  \[
  \mathcal{G}(H) = \mathbb{E}_g \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} g_i h(z_i) \right]
  \]

Interpretation: RC and GC reflect the ability of the hypothesis class to correlate with random noise.
Def.: Rademacher and Gaussian Complexity

- Let $\sigma_1, \ldots, \sigma_n$ be independent Rademacher variables (taking only values $\pm 1$, with equal probability).
- The **Rademacher complexity** (RC) is defined as
  \[
  \mathcal{R}(H) := \mathbb{E}_\sigma \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(z_i) \right]
  \]
- Let $g_1, \ldots, g_n \sim \mathcal{N}(0, 1)$.
- The **Gaussian complexity** (GC) is defined as
  \[
  \mathcal{G}(H) = \mathbb{E}_g \left[ \sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} g_i h(z_i) \right]
  \]

Interpretation: RC and GC reflect the ability of the hypothesis class to correlate with random noise.

**Theorem (Ledoux and Talagrand, 1991)**

\[
\mathcal{R}(H) \leq \sqrt{\frac{\pi}{2}} \mathcal{G}(H) \leq 3 \sqrt{\frac{\pi}{2}} \sqrt{\log n} \mathcal{R}(H).
\]
Existing Data-Dependent Analysis

The key step is estimating $\mathcal{R}(\{\rho_h : h \in H\})$ induced from the margin operator $\rho_h$ and class $H$.

Existing bounds build on the structural result:

$$\mathcal{R}(\max\{h_1, \ldots, h_C\} : h_j \in H_c, c = 1, \ldots, C) \leq \sum_{c=1}^{C} \mathcal{R}(H_c) \quad (1)$$
Existing Data-Dependent Analysis

The key step is estimating $\mathcal{R}(\{\rho_h : h \in H\})$ induced from the **margin operator** $\rho_h$ and class $H$.

Existing bounds build on the structural result:

$$\mathcal{R}(\max\{h_1, \ldots, h_C\} : h_j \in H_c, c = 1, \ldots, C) \leq \sum_{c=1}^{C} \mathcal{R}(H_c)$$  \hspace{1cm} (1)

Best known dependence on the number of classes:

- **quadratic dependence**  Koltchinskii and Panchenko (2002); Mohri et al. (2012); Cortes et al. (2013)

- **linear dependence**  Kuznetsov et al. (2014)
Existing Data-Dependent Analysis

The key step is estimating $\mathcal{R}(\{\rho_h : h \in H\})$ induced from the **margin operator** $\rho_h$ and class $H$.

Existing bounds build on the structural result:

$$\mathcal{R}(\max\{h_1, \ldots, h_C\} : h_j \in H_c, c = 1, \ldots, C) \leq \sum_{c=1}^{C} \mathcal{R}(H_c) \quad (1)$$

Best known dependence on the number of classes:

- **quadratic** dependence  
  Koltchinskii and Panchenko (2002); Mohri et al. (2012); Cortes et al. (2013)

- **linear** dependence  
  Kuznetsov et al. (2014)

Can we do better?
Existing Data-Dependent Analysis

The key step is estimating $\mathcal{R}(\{\rho_h : h \in H\})$ induced from the margin operator $\rho_h$ and class $H$.

Existing bounds build on the structural result:

$$\mathcal{R}(\max\{h_1, \ldots, h_C\} : h_j \in H_c, c = 1, \ldots, C) \leq \sum_{c=1}^{C} \mathcal{R}(H_c)$$  (1)

Best known dependence on the number of classes:

- **quadratic** dependence  Koltchinskii and Panchenko (2002); Mohri et al. (2012); Cortes et al. (2013)
- **linear** dependence  Kuznetsov et al. (2014)

Can we do better?

The correlation among class-wise components is ignored.
A New Structural Lemma on Gaussian Complexities

We consider Gaussian complexity.

We show:

\[ \mathcal{G}(\{\max\{h_1, \ldots, h_C\} : h = (h_1, \ldots, h_C) \in H\}) \leq \]

\[ \frac{1}{n} \mathbb{E}_g \sup_{h=(h_1, \ldots, h_C) \in H} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h_c(x_i) . \quad (2) \]
A New Structural Lemma on Gaussian Complexities

We consider Gaussian complexity.

We show:

\[ \mathcal{G}\left(\{\max\{h_1, \ldots, h_C\} : h = (h_1, \ldots, h_C) \in H\}\right) \leq \]
\[ \frac{1}{n} \mathbb{E}_g \sup_{h=(h_1, \ldots, h_C) \in H} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h_c(x_i). \]  \hspace{1cm} (2)

Core idea: **Comparison inequality** on GPs: (Slepian, 1962)

\[ \mathcal{X}_h := \sum_{i=1}^{n} g_i \max\{h_1(x_i), \ldots, h_C(x_i)\}, \mathcal{Y}_h := \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h_c(x_i), \forall h \in H. \]

\[ \mathbb{E}[(\mathcal{X}_\theta - \mathcal{X}_{\bar{\theta}})^2] \leq \mathbb{E}[(\mathcal{Y}_\theta - \mathcal{Y}_{\bar{\theta}})^2] \implies \mathbb{E}\left[\sup_{\theta \in \Theta} \mathcal{X}_\theta\right] \leq \mathbb{E}\left[\sup_{\theta \in \Theta} \mathcal{Y}_\theta\right]. \]

Eq. (2) preserves the coupling among class-wise components!
Example on Comparison of the Structural Lemma

Consider

\[ H := \{ (x_1, x_2) \rightarrow (h_1, h_2)(x_1, x_2) = (w_1 x_1, w_2 x_2) : \| (w_1, w_2) \|_2 \leq 1 \} \]

For the function class \( \{ \max \{ h_1, h_2 \} : h = (h_1, h_2) \in H \} \),

\[
\sup_{(h_1, h_2) \in H} \sum_{i=1}^{n} \sigma_i h_1(x_i) + \sup_{(h_1, h_2) \in H} \sum_{i=1}^{n} \sigma_i h_2(x_i)
\]

Preserving the coupling means supremum in a smaller space!
Estimating Multi-class Gaussian Complexity

Consider a vector-valued function class defined by

\[ H := \{ h^w = (\langle w_1, \phi(x) \rangle, \ldots, \langle w_c, \phi(x) \rangle) : f(w) \leq \Lambda \}, \]

where \( f \) is \( \beta \)-strongly convex w.r.t. \( \| \cdot \| \)

\[ f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \frac{\beta}{2} \alpha (1 - \alpha) \| x - y \|^2. \]

Theorem

\[
\frac{1}{n} \mathbb{E}_g \sup_{h^w \in H} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h^w_c(x_i) \leq \frac{1}{n} \sqrt{\frac{2\pi\Lambda}{\beta} \mathbb{E}_g \sum_{i=1}^{n} \| (g_{ic} \phi(x_i))_{c=1}^{C} \|_*^2}, \tag{3}
\]

where \( \| \cdot \|_* \) is the dual norm of \( \| \cdot \| \).
Features of the complexity bound

- Applies to a **general** function class defined through a strongly-convex regularizer \( f \)
- Class-wise components \( h_1, \ldots, h_C \) are correlated through the term
  \[
  \left\| \left( g_{ic} \phi(x_i) \right)_{c=1}^C \right\|_{*}^2
  \]
- Consider class \( H_{p,\Lambda} := \{ h^w : \|w\|_{2,p} \leq \Lambda \} \), \( \left( \frac{1}{p} + \frac{1}{p^*} = 1 \right) \); then:

\[
\frac{1}{n} \mathbb{E}_g \sup_{h^w \in H_{p,\Lambda}} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h^{w_c}(x_i) \leq \frac{\Lambda}{n} \sqrt{\sum_{i=1}^{n} k(x_i, x_i) \times \left\{ \begin{array}{ll}
\sqrt{e(4 \log C)}^{1 + \frac{1}{2 \log C}}, & \text{if } p^* \geq 2 \log C, \\
(2p^*)^{1 + \frac{1}{p^*}} \left( \frac{1}{C p^*} \right), & \text{otherwise.}
\end{array} \right.}
\]

The dependence is **sublinear** for \( 1 \leq p \leq 2 \), and even **logarithmic** when \( p \) approaches to 1!
\( \ell_p \)-norm MC-SVM

Consider class \( H_{p, \Lambda} := \{ h^w : \| w \|_{2,p} \leq \Lambda \} \), \((\frac{1}{p} + \frac{1}{p^*} = 1)\); then:

\[
\frac{1}{n} \mathbb{E}_g \sup_{h^w \in H_{p, \Lambda}} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h^w_c(x_i) \leq \frac{\Lambda}{n} \sqrt{\sum_{i=1}^{n} k(x_i, x_i) \times \begin{cases} \sqrt{e(4 \log C)}^{1+\frac{1}{2 \log C}}, & \text{if } p^* \geq 2 \log C, \\ (2p^*)^{1+\frac{1}{p^*}} C^{\frac{1}{p^*}}, & \text{otherwise}. \end{cases}}
\]

The dependence is **sublinear** for \( 1 \leq p \leq 2 \), and even **logarithmic** when \( p \) approaches to 1!
Future Directions

**Theory**: A data-dependent bound *independent* of the class size?
Future Directions

**Theory**: A data-dependent bound independent of the class size?

⇒ Need more powerful structural result on Gaussian complexity for functions induced by maximum operator.

► Might be worth to look into $\ell_\infty$-norm covering numbers.

**Reference**: Lei, Dogan, Binder, and Kloft (NIPS 2015); Journal submission forthcoming
Introduction

Distributed Algorithms

Theory

Learning Algorithms

Conclusion
Motivated by the **mild dependence** on $C$ as $p \to 1$, we consider

$$(\ell_p\text{-norm}) \text{ Multi-class SVM, } 1 \leq p \leq 2$$

$$\min_w \frac{1}{2} \left[ \sum_{c=1}^{C} \left\| w_c \right\|_p^2 \right]^{\frac{2}{p}} + C \sum_{i=1}^{n} (1 - t_i)_+, \quad (P)$$

s.t. $t_i = \langle w_{y_i}, \phi(x_i) \rangle - \max_{y:y \neq y_i} \langle w_y, \phi(x_i) \rangle$, 
## Empirical Results

<table>
<thead>
<tr>
<th>Method / Dataset</th>
<th>Sector</th>
<th>News 20</th>
<th>Rcv1</th>
<th>Birds 50</th>
<th>Caltech 256</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell_p )-norm MC-SVM</td>
<td>94.2±0.3</td>
<td>86.2±0.1</td>
<td>85.7±0.7</td>
<td>27.9±0.2</td>
<td>56.0±1.2</td>
</tr>
<tr>
<td>Crammer &amp; Singer</td>
<td>93.9±0.3</td>
<td>85.1±0.3</td>
<td>85.2±0.3</td>
<td>26.3±0.3</td>
<td>55.0±1.1</td>
</tr>
</tbody>
</table>

Proposed \( \ell_p \)-norm MC-SVM consistently better on benchmark datasets.
Wait... I performed this Experiment:
Wait... I performed this Experiment:

▶ So I took the DMOZ2010 dataset
(Aim: categorize new webpages)
Wait... I performed this Experiment:

- OVR-SVM, Train=128,710, Test=34,880; Result:

27% of classes never used in prediction
New Learning Algorithm

Schatten-SVM

\[
\min_{W=(w_1, \ldots, w_C)} \frac{1}{2} \sum_c \| W \|_{S_p}^2 + C \sum_{i=1}^n \left[ \max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right]
\]

Schatten norm

\[
\| W \|_{S_p} := \sqrt{\sum \sigma_i^p (\sqrt{W^T W})}
\]
Geometry of Schatten Norm

\[ \sigma_2 \]

\[ \sigma_1 \]

\[ p = \infty \]

\[ p = 2 \]

\[ p = 1 \]
Schatten-norm Parameter $p$ Controls coverage
## Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Schatten-SVM</th>
<th>OvR</th>
<th>CS-SVM</th>
<th>HR-SVM</th>
<th>HR-LR</th>
<th>TD-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLEF</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-F1</td>
<td>58.42 (52.20)</td>
<td>53.11</td>
<td>57.17</td>
<td>53.92</td>
<td>55.83</td>
<td>32.32</td>
</tr>
<tr>
<td>Micro-F1</td>
<td>80.21 (78.82)</td>
<td>78.92</td>
<td>79.94</td>
<td>80.02</td>
<td>80.12</td>
<td>70.11</td>
</tr>
<tr>
<td>Coverage</td>
<td>90.48 (85.71)</td>
<td>87.30</td>
<td>88.93</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LSHTC-SMALL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-F1</td>
<td>30.10 (30.12)</td>
<td>26.89</td>
<td>28.22</td>
<td>28.94</td>
<td>28.12</td>
<td>20.01</td>
</tr>
<tr>
<td>Micro-F1</td>
<td>46.12 (45.85)</td>
<td>43.34</td>
<td>45.77</td>
<td>45.31</td>
<td>44.94</td>
<td>38.48</td>
</tr>
<tr>
<td>Coverage</td>
<td>60.66 (61.54)</td>
<td>54.52</td>
<td>55.87</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WIKI-2011</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-F1</td>
<td>30.29</td>
<td>25.13</td>
<td>27.35</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Micro-F1</td>
<td>44.86</td>
<td>39.07</td>
<td>43.47</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Coverage</td>
<td>74.58</td>
<td>61.51</td>
<td>67.90</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>DMOZ-2010</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-F1</td>
<td>32.04</td>
<td>31.27</td>
<td>32.64</td>
<td>33.12</td>
<td>32.42</td>
<td>22.30</td>
</tr>
<tr>
<td>Micro-F1</td>
<td>44.12</td>
<td>45.12</td>
<td>45.36</td>
<td>46.02</td>
<td>45.84</td>
<td>38.64</td>
</tr>
<tr>
<td>Coverage</td>
<td>68.57</td>
<td>63.82</td>
<td>64.50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Future Directions

**Algorithms**: New models & efficient solvers

- **Novel models** motivated by theory
  - top-k MC-SVM (Lapin et al., 2015)
- Analyze $p > 2$ regime
- Extensions to **multi-label** learning
Conclusion

Extreme Classification

\[ \mathbb{E}\left( \max \{ h_1, \ldots, h_C \} : h = (h_1, \ldots, h_C) \in H \right) \leq \frac{1}{n} \mathbb{E}_g \sup_{h=(h_1, \ldots, h_C) \in H} \sum_{i=1}^{n} \sum_{c=1}^{C} g_{ic} h_c(x_i) \]
Conclusion


