HUMBOLDT-UNIVERSITÄT ZU BERLIN

Extreme Classification

Tighter Bounds, Distributed Training, and new Algorithms

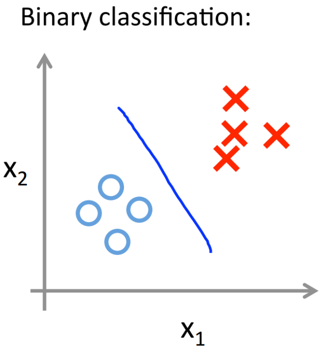
Marius Kloft (HU Berlin)

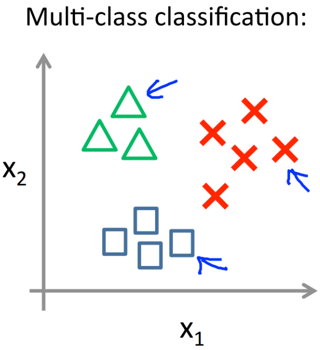
Krakow, Feb 16, 2017

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.



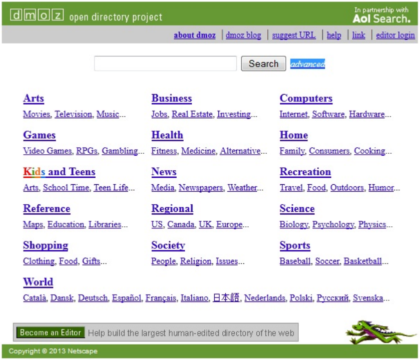


What is Extreme Classification?

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.

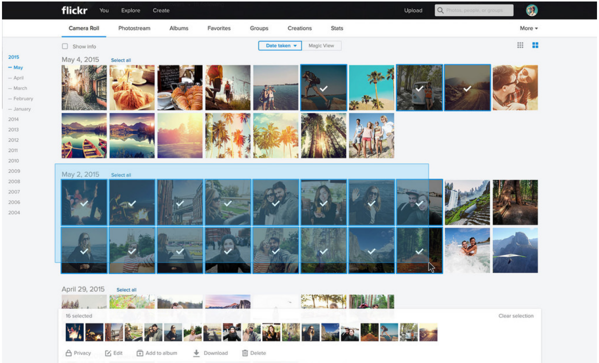


Example 2

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

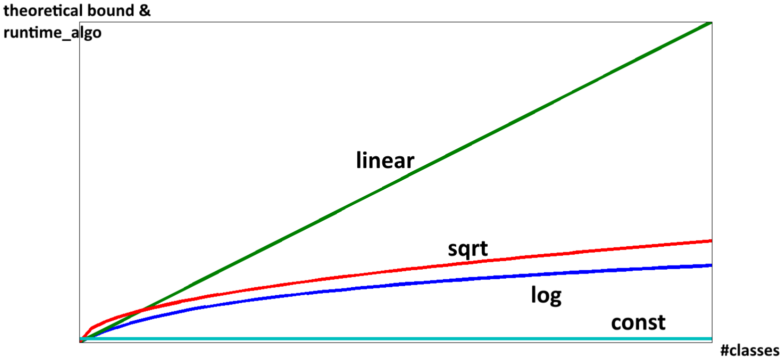
We add new articles to an online encyclopedia and intend to predict the categories of the articles.

Need

Need for theory and algorithms for **extreme classification**.

How do **algorithms** and **bounds** scale in **#classes**?

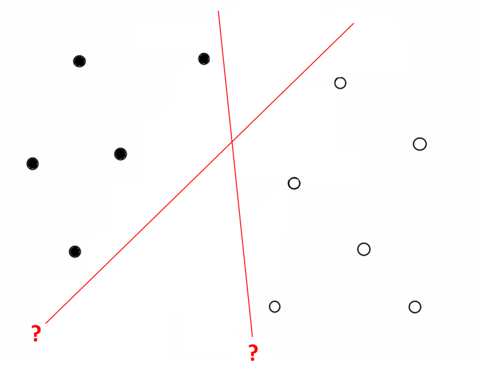
How do **algorithms** and **bounds** scale in **#classes**?



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

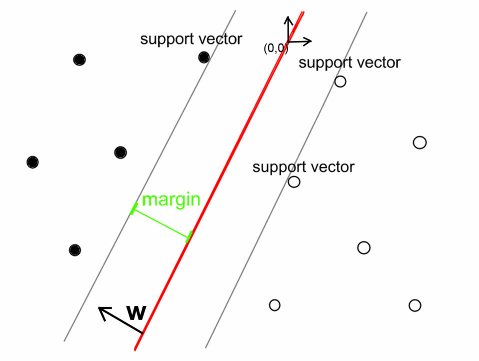
Core idea:

- 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**



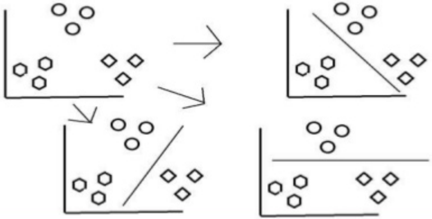
Popular Generalization to Multiple Classes: One-vs.-Rest SVM

Put $\mathcal{C}$:= #classes.

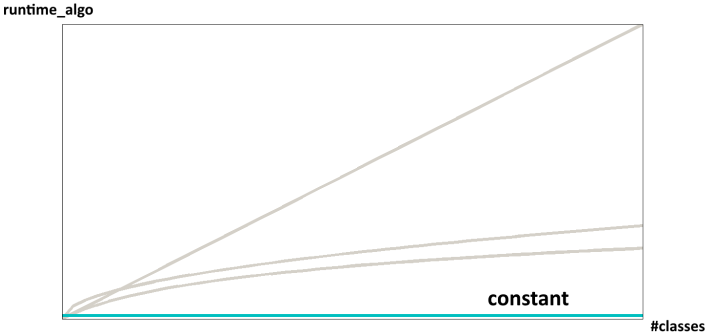
One-vs.-rest SVM

```
For c = 1..C
    class1 := c, class2 := union(allOtherClasses)
    w_c := solutionOfSVM(class1,class2)
end
Given a test point x, predict c_predicted := arg max_c w_c^T x
```

For $c = 1..\mathcal{C}$
  class1 := c, class2 := union(allOtherClasses)
  w_c := solutionOfSVM(class1,class2)
end
Given a test point x, predict $c_{\text{predicted}} := \arg\max_c w_c^\top x$



Runtime of One-vs.-Rest



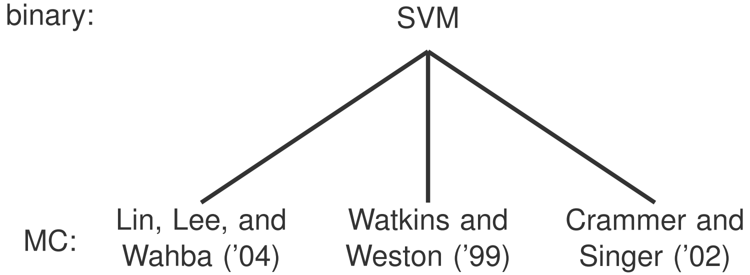
... assuming sufficient computational resources (#classes many computers)

Problem With One-vs.-Rest

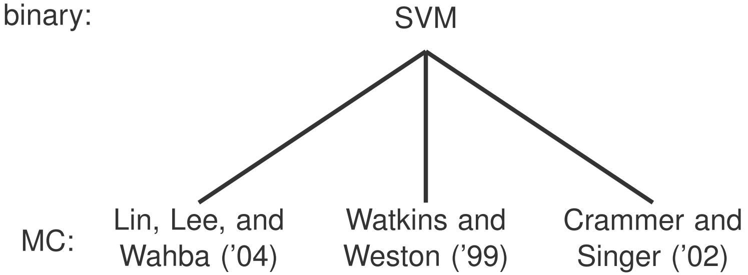
:) training **can be parallelized** in the number of classes (extreme classification!)

:(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

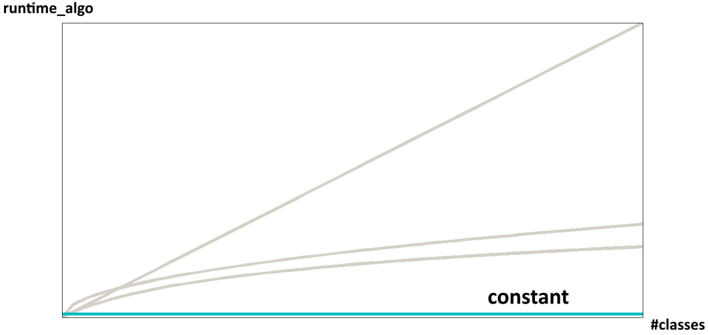


There are "True" Multi-class SVMs, So-called **All-in-one** Multi-class SVMs



Problem: State of the art solvers require a training time complexity of $\mathcal{O}(dn \cdot \mathbf{C})$, where d =dim, n=#examples, and $\mathbf{C}$:=#classes.

Aim: Develop algorithms where $\mathcal{O}(\mathbf{C})$ machines in **parallel** and in $\mathcal{O}(dn)$ runtime train all-in-one MC-SVMs.



⇒ 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:

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

All-in-one SVMs

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

$$\min_{w=(w_1,\ldots,\mathcal{C})} \frac{1}{2}\sum_{c} \|w_c\|^2 + C * L(w, \mathtt{data})$$

All Three MC-SVMs have:

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

All Three MC-SVMs have:

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

But they differ in the loss:

note: $l(x) := max(0, 1 - x)$

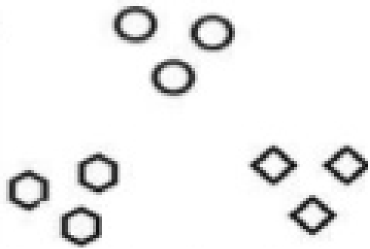
CS: $$\ldots \sum_{i=1}^{n} \left[\max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right]$$

All Three MC-SVMs have:

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

But they differ in the loss: note: $l(x) := max(0, 1-x)$

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



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:

note: $l(x) := max(0, 1 - x)$

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

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

Sources: Lee, Lin, and Wahba (2004), Weston and Watkins (1999), Crammer and Singer (2002)

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:

note: $l(x) := max(0, 1 - x)$

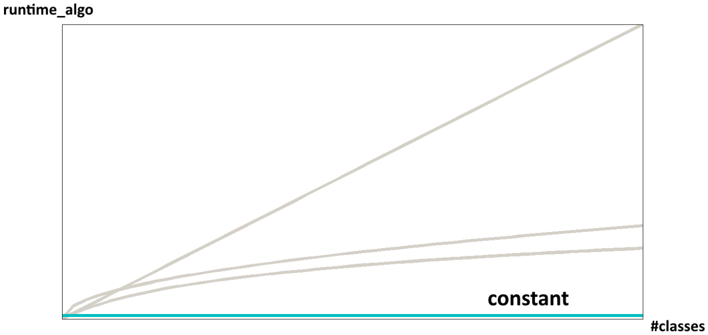
CS: $$\ldots \sum_{i=1}^{n} \left[\max_{c \neq y_i} l((w_{y_i} - w_c)^T x_i) \right]$$

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

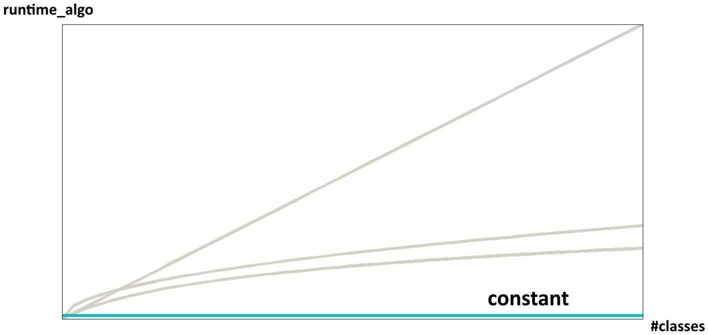
LLW: $$\ldots \sum_{i=1}^{n} \left[\sum_{c \neq y_i} l(-w_c^T x_i) \right], \text{ s.t. } \sum_c w_c = 0$$

Sources: Lee, Lin, and Wahba (2004), Weston and Watkins (1999), Crammer and Singer (2002)

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

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{c=1}^{\mathcal{C}} ||X\alpha_c - \underbrace{\frac{1}{\mathcal{C}}\sum_{\tilde{c}} X\alpha_{\tilde{c}}}\,||^2 + \sum_{c,i:y_i=c} \alpha_i$$

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

$0 \leq \alpha_{i,c} \leq C$

This is the LLW **Dual** Problem

$$\max_{\alpha} \max_{\bar{w}} -\frac{1}{2}\sum_{c=1}^{\mathcal{C}} ||X\alpha_c - \underbrace{\frac{1}{\mathcal{C}}\sum_{\tilde{c}} X\alpha_{\tilde{c}}}_{=\bar{w}}||^2 + \sum_{c,i:y_i=c} \alpha_i$$

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

$0 \leq \alpha_{i,c} \leq C$

This is the LLW **Dual** Problem

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

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

$0 \leq \alpha_{i,c} \leq C$

LLW: Proposed Algorithm

Algorithm Simple wrapper algorithm

```
function SimpleSolve-LLW(C, X, Y)
    while not converged do
        for c = 1..𝒞 do in parallel
            α_c ← argmax_{α̃_c} D_c(α̃_c, w̄)
        end for
        w̄ ← argmax_w D(α, w)
    end while
end function
```

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

LLW: Proposed Algorithm

Algorithm Simple wrapper algorithm

```
function SimpleSolve-LLW(C, X, Y)
    while not converged do
        for c = 1..𝒞 do in parallel
            α_c ← argmax_{α̃_c} D_c(α̃_c, w̄)
        end for
        w̄ ← argmax_w D(α, w)
    end while
end function
```

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

LLW: Proposed Algorithm

Algorithm Simple wrapper algorithm

```
function SimpleSolve-LLW(C, X, Y)
    while not converged do
        for c = 1..𝒞 do in parallel
            α_c ← argmax_{α̃_c} D_c(α̃_c, w̄)
        end for
        w̄ ← argmax_w D(α, w)
    end while
end function
```

Alber, Zimmert, Dogan, and Kloft (2016):
NIPS ~~submitted~~ rejected ;)
PLoS submitted, arXiv:1611.08480

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

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

$$\text{s.t.} \qquad \forall i: \ \alpha_{i,y_i} = - \sum_{c:c \neq y_i} \alpha_{i,c},$$

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

WW: This is How the **Dual** Problem Looks Like

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

$$\text{s.t.} \qquad \forall i: \; \alpha_{i,y_i} = - \sum_{c:c \neq y_i} \alpha_{i,c},$$

$$\forall c \neq y_i: \; 0 \leq \alpha_{i,c} \leq C$$

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

for $i = 1, ..., n$

  for $c = 1, \ldots, \mathcal{C}$

    $\alpha_{i,c} = \max_{\alpha_{i,c}} D(\alpha)$

  end

end

This is Now the Story...

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

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

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

This is Now the Story...

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

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

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

Can we exploit this?

Analogy: Soccer League Schedule

We are given a football league (e.g., Bundesliga) with $\mathcal{C}$ 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 $\mathcal{C} = 18$ teams.

$\Rightarrow$ $\mathcal{C} - 1 = 17$ 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 $\mathcal{C}$ 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 $\mathcal{C} = 18$ teams.

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

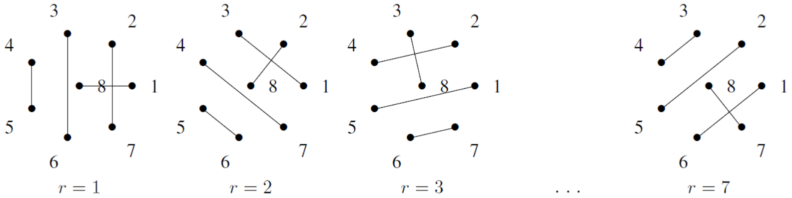
How can we come up with a schedule?

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: $\mathcal{C} = 8$ many teams, 7 matchdays

WW: Proposed Algorithm

Algorithm Simplistic DBCA
wrapper algorithm

function SIMPLESOLVE-WW(C, X, Y)
    while not converged **do**
        for $r = 1 ... \mathcal{C} - 1$ **do** `# iterate over "matchdays"`
            for $c = 1..\mathcal{C}/2$ **do in parallel** `# iterate over "matches"`
                $(c_i, c_j) \leftarrow$ the two classes ("opposing teams")
                $\alpha_{I_{c_i}, c_j}, \alpha_{I_{c_j}, c_i} \leftarrow \arg\max_{\alpha_1, \alpha_2} D_c(\alpha_1, \alpha_2)$
            end for
        end for
    end while
end function

Alber, Zimmert, Dogan, and Kloft (2016): arXiv:1611.08480

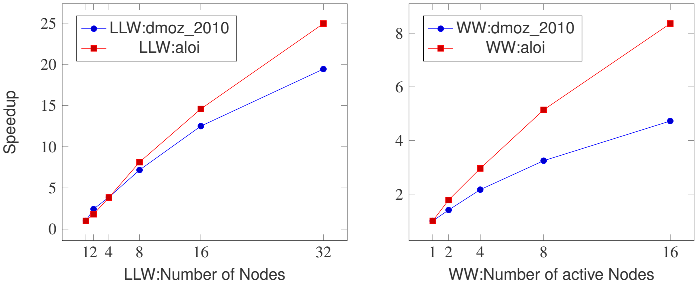
Accuracies

Dataset	# Training	# Test	# Classes	# Features
ALOI	98,200	10,800	1000	128
LSHTCsmall	4,463	1,858	1,139	51,033
DMOZ2010	128,710	34,880	12,294	381,581

Dataset	OVR	CS	WW	LLW
ALOI	0.1824	0.0974	0.0930	0.6560
LSHTCsmall	0.549	0.5919	0.5505	0.9263
DMOZ2010	0.5721	-	0.5432	0.9586

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

Results: Speedup

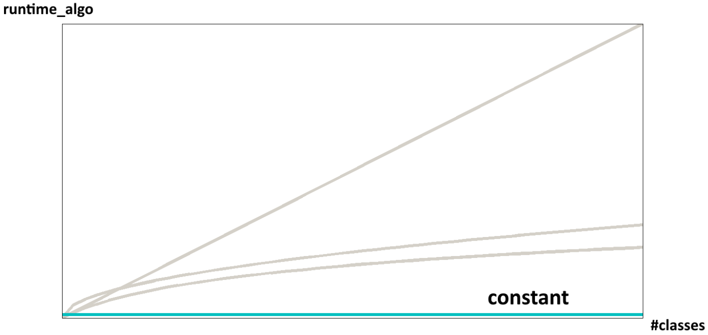


Open questions

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

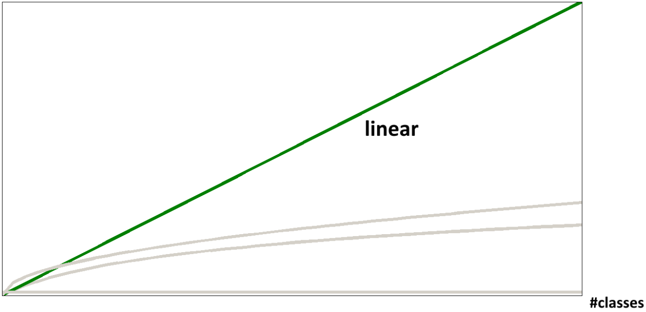
Theory and Algorithms in Extreme Classification

- 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)

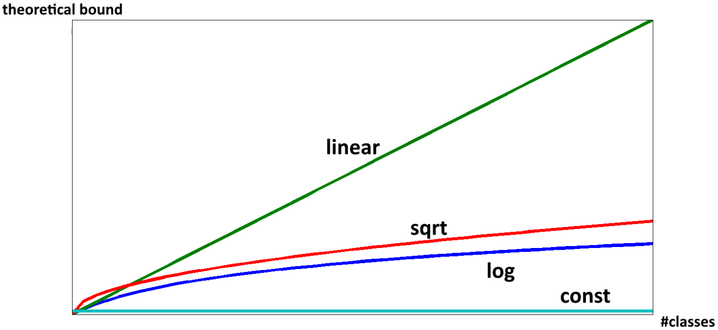


Theory of Extreme Classification

Questions

- Can we get bounds with **mild** dependence on #classes?

  $\Rightarrow$ Novel algorithms?



Multi-class Classification

Given:

- Training data $\underbrace{z_1 = (x_1, y_1), \ldots, z_n = (x_n, y_n)}_{\in \mathcal{X} \times \mathcal{Y}} \overset{\text{i.i.d.}}{\sim} P$
  - $\mathcal{Y} := \{1, 2, \ldots, \mathcal{C}\}$
  - $\mathcal{C}$ = number of classes

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

$$\hat{y} := \boxed{\arg\max}_{y \in \mathcal{Y}} h_y(x)$$

Multi-class SVMs:

- $h_y(x) = \langle \mathbf{w}_y, \phi(x) \rangle$
- 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 ± 1, with equal probability).
- The **Rademacher complexity** (RC) is defined as

$$\mathfrak{R}(H) := \mathbb{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \boxed{\sigma_i} h(z_i) \Big]$$

- Let $g_1, \ldots, g_n \sim N(0,1)$.
- The **Gaussian complexity** (GC) is defined as

$$\mathfrak{G}(H) = \mathbb{E}_{\boldsymbol{g}} \Big[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \boxed{g_i} h(z_i) \Big]$$

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 ± 1, with equal probability).
- The **Rademacher complexity** (RC) is defined as

$$\mathfrak{R}(H) := \mathbb{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \boxed{\sigma_i} h(z_i) \Big]$$

- Let $g_1, \ldots, g_n \sim N(0, 1)$.
- The **Gaussian complexity** (GC) is defined as

$$\mathfrak{G}(H) = \mathbb{E}_{\boldsymbol{g}} \Big[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \boxed{g_i} h(z_i) \Big]$$

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

Theorem ((Ledoux and Talagrand, 1991))

$$\mathfrak{R}(H) \leq \sqrt{\frac{\pi}{2}} \mathfrak{G}(H) \leq 3 \sqrt{\frac{\pi}{2}} \sqrt{\log n} \mathfrak{R}(H).$$

Existing Data-Dependent Analysis

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

Existing bounds build on the structural result:

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

Existing Data-Dependent Analysis

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

Existing bounds build on the structural result:

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

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 $\mathfrak{R}(\{\rho_h : h \in H\})$ induced from the **margin operator** ρ_h and class H.

Existing bounds build on the structural result:

$$\mathfrak{R}(\max\{h_1, \ldots, h_{\mathcal{C}}\} : h_j \in H_c, c = 1, \ldots, \mathcal{C}) \le \boxed{\sum_{c=1}^{\mathcal{C}} \mathfrak{R}(H_c)} \quad (1)$$

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 $\mathfrak{R}(\{\rho_h : h \in H\})$ induced from the **margin operator** ρ_h and class H.

Existing bounds build on the structural result:

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

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?

The correlation among class-wise components is ignored.

A New Structural Lemma on Gaussian Complexities

We consider Gaussian complexity.

- We show:

$$\mathfrak{G}\Big(\{\max\{h_1,\ldots,h_{\mathcal{C}}\} : h = (h_1,\ldots,h_{\mathcal{C}}) \in H\}\Big) \leq \boxed{\frac{1}{n}\mathbb{E}_{\boldsymbol{g}} \sup_{h=(h_1,\ldots,h_{\mathcal{C}})\in H} \sum_{i=1}^{n}\sum_{c=1}^{\mathcal{C}} g_{ic} h_c(x_i)}. \quad (2)$$

A New Structural Lemma on Gaussian Complexities

We consider Gaussian complexity.

- We show:

$$\mathfrak{G}\big(\{\max\{h_1,\ldots,h_{\mathcal{C}}\} : h=(h_1,\ldots,h_{\mathcal{C}})\in H\}\big) \leq \boxed{\frac{1}{n}\mathbb{E}_{\boldsymbol{g}} \sup_{h=(h_1,\ldots,h_{\mathcal{C}})\in H} \sum_{i=1}^{n}\sum_{c=1}^{\mathcal{C}} g_{ic}h_c(x_i)}. \quad (2)$$

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

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

$$\mathbb{E}[(\mathfrak{X}_\theta-\mathfrak{X}_{\bar\theta})^2] \leq \mathbb{E}[(\mathfrak{Y}_\theta-\mathfrak{Y}_{\bar\theta})^2] \Longrightarrow \mathbb{E}[\sup_{\theta\in\Theta}\mathfrak{X}_\theta] \leq \mathbb{E}[\sup_{\theta\in\Theta}\mathfrak{Y}_\theta].$$

Eq. (2) preserves the coupling among class-wise components!

Example on Comparison of the Structural Lemma

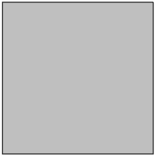
- Consider

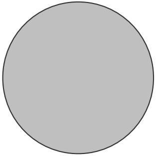
$$H := \{(x_1, x_2) \to (h_1, h_2)(x_1, x_2) = (w_1 x_1, w_2 x_2) : \|(w_1, w_2)\|_2 \le 1\}$$

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

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

$$\sup_{(h_1,h_2)\in H} \sum_{i=1}^n [g_{i1} h_1(x_i) + g_{i2} h_2(x_i)]$$





Preserving the coupling means supremum in a smaller space!

Estimating Multi-class Gaussian Complexity

- Consider a **vector-valued** function class defined by

$$H := \{h^{\mathbf{w}} = (\langle \mathbf{w}_1, \phi(x)\rangle, \ldots, \langle \mathbf{w}_c, \phi(x)\rangle) : f(\mathbf{w}) \leq \Lambda\},$$

where f is **β-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

$$\frac{1}{n}\mathbb{E}_{\mathbf{g}} \sup_{h^{\mathbf{w}} \in H} \sum_{i=1}^{n} \sum_{c=1}^{\mathcal{C}} g_{ic} h_c^{\mathbf{w}}(x_i) \leq \frac{1}{n}\sqrt{\frac{2\pi\Lambda}{\beta}\mathbb{E}_{\mathbf{g}} \sum_{i=1}^{n} \left\| \Big(g_{ic}\phi(x_i)\Big)_{c=1}^{\mathcal{C}} \right\|_*^2}, \quad (3)$$

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 $h_1, \ldots, h_{\mathcal{C}}$ are correlated through the term $\left\| \left(g_{ic}\phi(x_i) \right)_{c=1}^{\mathcal{C}} \right\|_*^2$
- Consider class $H_{p,\Lambda} := \{h^{\mathbf{w}} : \|\mathbf{w}\|_{2,p} \leq \Lambda\}$, $(\frac{1}{p} + \frac{1}{p^*} = 1)$; then:

$$\frac{1}{n}\mathbb{E}_{\boldsymbol{g}} \sup_{h^{\mathbf{w}} \in H_{p,\Lambda}} \sum_{i=1}^{n} \sum_{c=1}^{\mathcal{C}} g_{ic} h_c^{\mathbf{w}}(x_i) \leq \frac{\Lambda}{n} \sqrt{\sum_{i=1}^{n} k(x_i, x_i)} \times$$

$$\begin{cases} \sqrt{e}(4\log\mathcal{C})^{1+\frac{1}{2\log\mathcal{C}}}, & \text{if } p^* \geq 2\log\mathcal{C}, \\ (2p^*)^{1+\frac{1}{p^*}} \boxed{\mathcal{C}^{\frac{1}{p^*}}}, & \text{otherwise.} \end{cases}$$

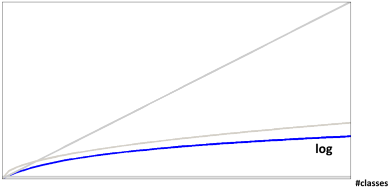
The dependence is **sublinear** for $1 \leq p \leq 2$, and even **logarithmic** when p approaches to 1!

ℓ_p-norm MC-SVM

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

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

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?

$\Rightarrow$ Need more powerful structural result on Gaussian complexity for functions induced by **maximum operator**.

- Might be worth to look into ℓ_∞**-norm covering numbers**.

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

ℓ_p-norm Multi-class SVM

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

(ℓ_p-norm) Multi-class SVM, $1 \le p \le 2$

$$\min_{\mathbf{w}} \frac{1}{2}\Big[\sum_{c=1}^{\mathcal{C}} \|\mathbf{w}_c\|_2^p\Big]^{\frac{2}{p}} + C\sum_{i=1}^{n}(1-t_i)_+, \qquad \text{(P)}$$

$$\text{s.t. } t_i = \langle \mathbf{w}_{y_i}, \phi(x_i)\rangle - \max_{y:y\neq y_i}\langle \mathbf{w}_y, \phi(x_i)\rangle,$$

Empirical Results

Empirical Results:

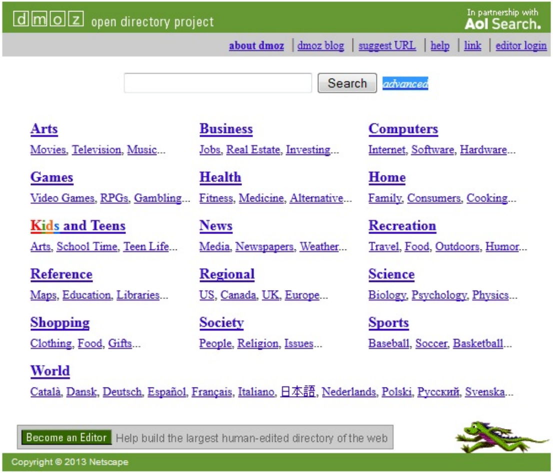
Method / Dataset	Sector	News 20	Rcv1	Birds 50	Caltech 256
ℓ_p-norm MC-SVM	94.2±0.3	86.2±0.1	85.7±0.7	27.9±0.2	56.0±1.2
Crammer & Singer	93.9±0.3	85.1±0.3	85.2±0.3	26.3±0.3	55.0±1.1

Proposed ℓ_p-norm MC-SVM consistently better on benchmark datasets.

Wait... I performed this Experiment:

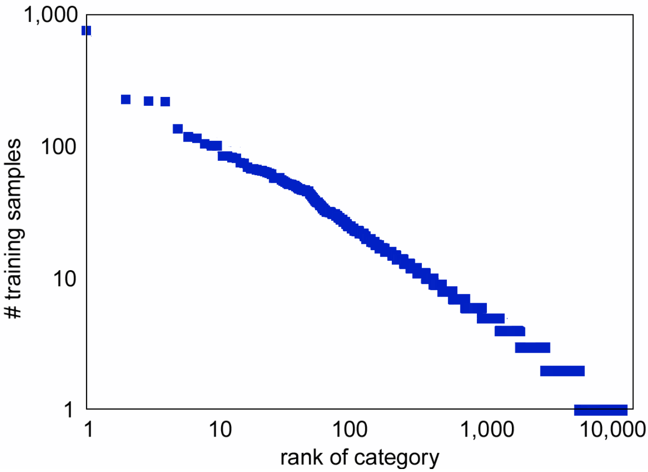
Wait... I performed this Experiment:

- 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

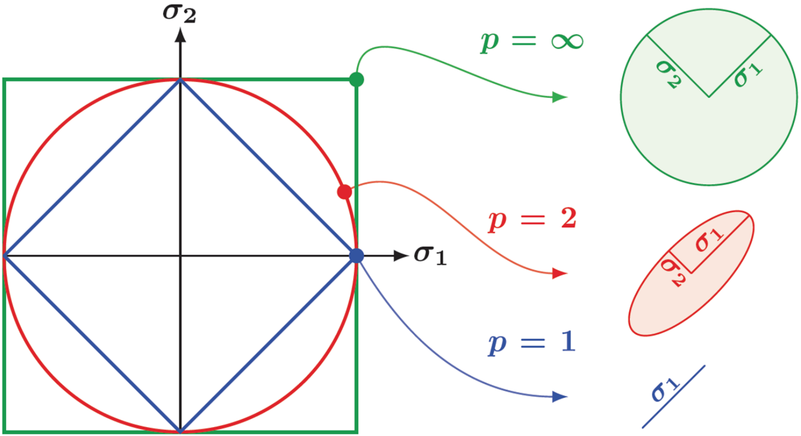
Schatten-SVM

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

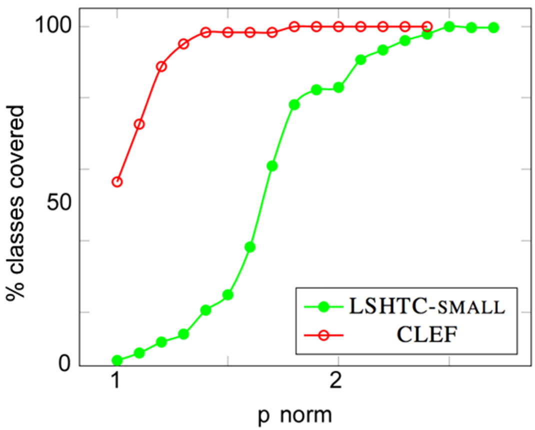
Schatten-p norm

$$\|W\|_{S_p} := \sqrt[p]{\sum_i \sigma_i^p(\sqrt{W^\top W})}$$

Geometry of Schatten Norm



Schatten-norm Parameter p Controls **coverage**



Results

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

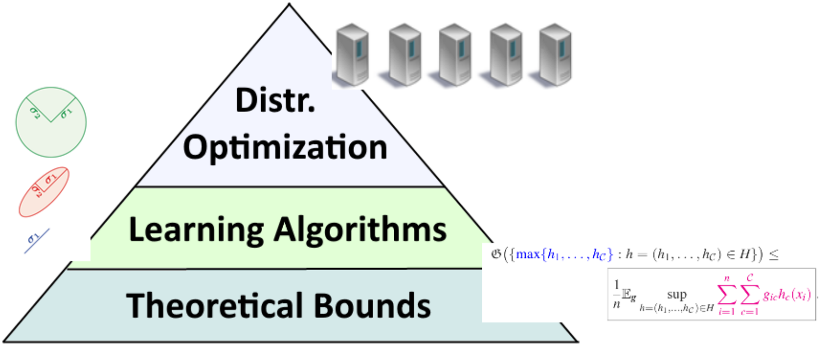
Future Directions

Algorithms: New models & efficient solvers

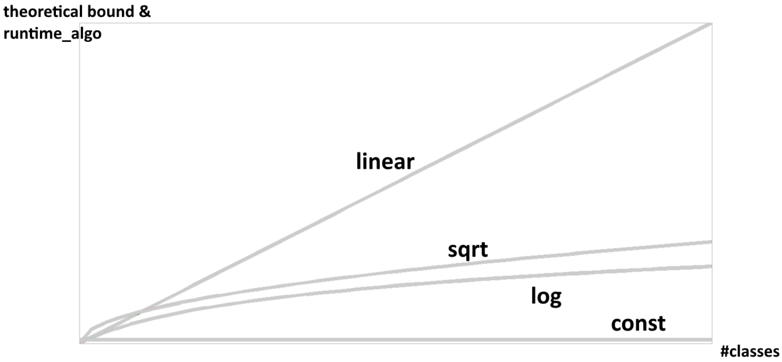
- **Novel models** motivated by theory
  - top-k MC-SVM (Lapin et al., 2015)
- Analyze $p > 2$ regime
- Extensions to **multi-label** learning

Conclusion

Extreme Classification



Conclusion


theoretical bound &
runtime_algo
linear
sqrt
log
const
#classes

Refs I

M. Alber, J. Zimmert, U. Dogan, and M. Kloft. Distributed Optimization of Multi-Class SVMs. **in submission**, 2016.

C. Cortes, M. Mohri, and A. Rostamizadeh. Multi-class classification with maximum margin multiple kernel. In **ICML-13**, pages 46–54, 2013.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. **Journal of Machine Learning Research**, 2:265–292, 2002.

Y. Guermeur. Combining discriminant models with new multi-class svms. **Pattern Analysis & Applications**, 5(2): 168–179, 2002.

S. I. Hill and A. Doucet. A framework for kernel-based multi-category classification. **Journal of Artificial Intelligence Research**, 30(1):525–564, 2007.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. **Annals of Statistics**, pages 1–50, 2002.

V. Kuznetsov, M. Mohri, and U. Syed. Multi-class deep boosting. In **Advances in Neural Information Processing Systems**, pages 2501–2509, 2014.

M. Lapin, M. Hein, and B. Schiele. Top-k multiclass SVM. **CoRR**, abs/1511.06683, 2015. URL `http://arxiv.org/abs/1511.06683`.

M. Ledoux and M. Talagrand. **Probability in Banach Spaces: isoperimetry and processes**, volume 23. Springer, Berlin, 1991.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. **Journal of the American Statistical Association**, 99(465):67–82, 2004.

Y. Lei, Ü. Dogan, A. Binder, and M. Kloft. Multi-class svms: From tighter data-dependent generalization bounds to novel algorithms. In **Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada**, pages 2035–2043, 2015. URL `http://papers.nips.cc/paper/6012-multi-class-svms-from-tighter-data-dependent-generalization-bounds-to-novel-algorith`

M. Mohri, A. Rostamizadeh, and A. Talwalkar. **Foundations of machine learning**. MIT press, 2012.

Refs II

D. Slepian. The one-sided barrier problem for gaussian noise. **Bell System Technical Journal**, 41(2):463–501, 1962.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In M. Verleysen, editor, **Proceedings of the Seventh European Symposium On Artificial Neural Networks (ESANN)**, pages 219–224. Evere, Belgium: d-side publications, 1999.

T. Zhang. Class-size independent generalization analsysis of some discriminative multi-category classification. In **Advances in Neural Information Processing Systems**, pages 1625–1632, 2004a.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. **The Journal of Machine Learning Research**, 5:1225–1251, 2004b.