A Unified View of Local Learning:

Theory and Algorithms for Enhancing Linear Models

Valentina Zantedeschi

Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien
UMR 5516, F-42023, SAINT-ETIENNE, France

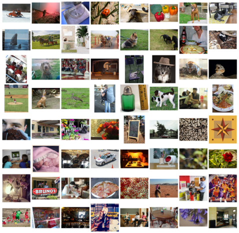
18/12/2018

Florence D'ALCHE-BUC	Professeure, Télécom ParisTech	Rapporteure
Marianne CLAUSEL	Professeure, Université de Lorraine	Rapporteure
Marc TOMMASI	Professeur, Université de Lille	Examinateur
Pascal GERMAIN	Chargé de Recherche, Université de Lille	Examinateur
Marc SEBBAN	Professeur, Université de Saint-Étienne	Directeur
Rémi EMONET	Maître de Conférences, Université de Saint-Étienne	Co-encadrant

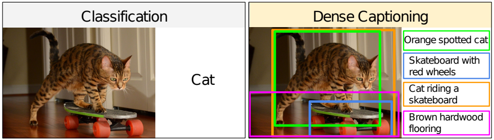
Machine Learning

Learning to perform a task from examples

Examples [Deng et al., 2009]:



Possible tasks [Johnson et al., 2016]:



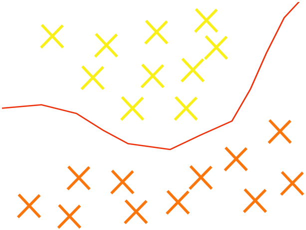
1. extrapolate new information
2. estimate the probability of certain events
3. make decisions

Machine Learning

Learning to perform a task from examples

In practice

- examples are embedded in feature spaces (representation)
- mathematical models are inferred through an algorithm

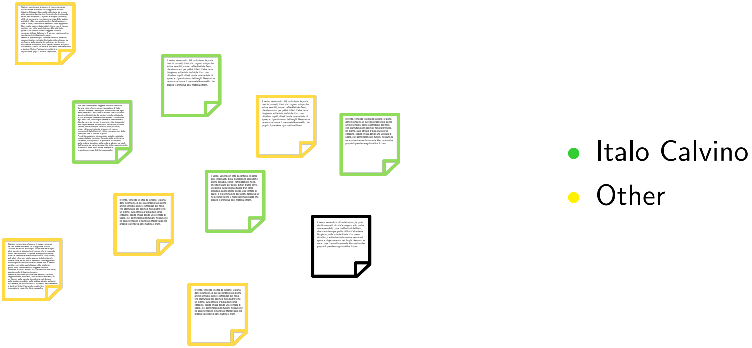


Supervised Learning

- annotated examples $S = \{z_i = (x_i \in \mathcal{X}, y_i \in \mathcal{Y})\}_{i=1}^{m}$
- learn to predict the target output y_i from the given input x_i

Example: Author Recognition

Corpora of documents written by a given author or not



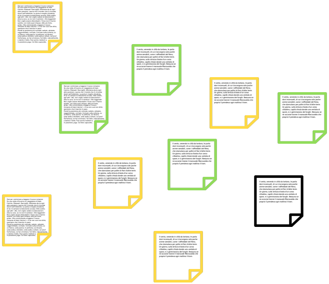
example of features: histograms of words from a dictionary

Supervised Learning

- annotated examples $S = \{z_i = (x_i \in \mathcal{X}, y_i \in \mathcal{Y})\}_{i=1}^m$
- learn to predict the target output y_i from the given input x_i

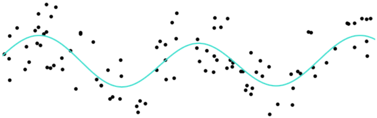
Binary Classification

$y_i \in \{-1, 1\}$



Regression

$y_i \in \mathbb{R}$



Learning Procedure

1. fix the **hypothesis class** $\mathcal{C}$

Definition

(**Hypothesis class**) A hypothesis class $\mathcal{C}$ is the set of candidate models from which the learning algorithm selects the most suitable model for the task.

ex. set of linear classifiers $f(x) = \text{sign}(\langle \theta, x \rangle + b)$

Learning Procedure

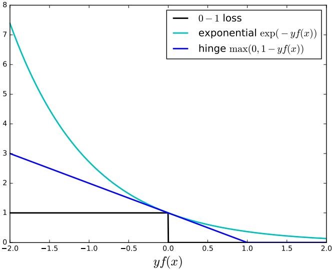
1. fix the **hypothesis class** $\mathcal{C}$
2. choose a **loss** function ℓ

Definition

(**Loss function**) A loss function ℓ assesses the agreement between predicted and target values.

ex. margin-based losses for $f \in \mathcal{C}$ and $z = (x, y)$:

$$\text{hinge loss} \quad \ell(f, z) = \max(0, 1 - yf(x))$$
$$\text{exponential loss} \quad \ell(f, z) = \exp(-yf(x))$$



Learning Procedure

1. fix the **hypothesis class** $\mathcal{C}$
2. choose a **loss** function ℓ
3. minimize the **empirical risk** on sample $S = \{z_i\}_{i=1}^m$

$$\min_{f \in \mathcal{C}} \hat{R}_S(f)$$

$$\begin{aligned} \hat{R}_S(f) &= \mathbb{E}_{z \sim S}\ \ell(f, z) \\ &= \frac{1}{m} \sum_{i=1}^{m} \ell(f, z_i) \end{aligned}$$

Regularization

$$\min_{f \in \mathcal{C}} \hat{R}_S(f) + \lambda \|f\|$$

Regularization

$$\min_{f \in \mathcal{C}} \hat{R}_S(f) + \lambda \|f\|$$

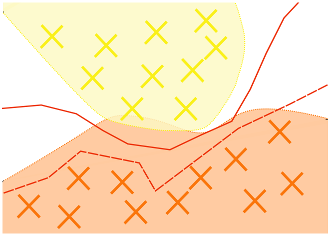
limited sample S drawn from data distribution $\mathcal{D}$

memorization (*over-fitting*): have good performance only on S

generalization: have good performance on any sample from $\mathcal{D}$

Occam's razor principle:

the simplest solution tends to be the best one



Regularization

$$\min_{f \in \mathcal{C}} \hat{R}_S(f) + \lambda \|f\|$$

limited sample S drawn from data distribution $\mathcal{D}$

memorization (*over-fitting*): have good performance only on S

generalization: have good performance on any sample from $\mathcal{D}$

Occam's razor principle:

the simplest solution tends to be the best one

Other reasons

- to inject side-information, prior knowledge on the problem
- to correct ill-posed problems
- to converge faster

Evaluation

estimating the true risk $R_{\mathcal{D}}$

Theoretical Guarantees

- generalization bounds on the gap between the true risk $R_{\mathcal{D}}$ and the empirical risk $\hat{R}_S$ [Valiant, 1984]:

$$\mathbb{P}\left(\left|R_{\mathcal{D}}(f) - \hat{R}_S(f)\right| \leq \varepsilon\right) \geq 1 - \delta.$$

Different Frameworks

- based on hypothesis class complexity
- considering the learning algorithm:
  1. Algorithmic Robustness [Xu and Mannor, 2012]
     $\rightarrow$ consistent predictions on points that belong to the same region of the space
  2. Uniform Stability [Bousquet and Elisseeff, 2002]
     $\rightarrow$ similar models learned on similar training sets

Contributions of the Thesis

Tackled problems:

1. local learning [Zantedeschi et al., 2016d,a,c, 2017a]
2. decentralized learning [Zantedeschi et al., 2018a]
3. learning from weakly-labeled data [Zantedeschi et al., 2016b]
4. learning from multi-view data [Zantedeschi et al., 2018b]
5. graph optimization [Zantedeschi et al., 2018a]
6. adversarial robustness [Zantedeschi et al., 2017b]

Applications:

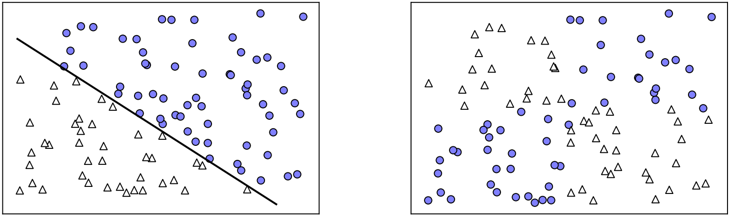
1. perceptual color distance [Zantedeschi et al., 2016d,a]
2. word similarity [Zantedeschi et al., 2016d,a]
3. image segmentation [Zantedeschi et al., 2016d,a]
4. human activity recognition [Zantedeschi et al., 2018a]
5. autism spectrum disorder detection [Zantedeschi et al., 2018b]

Outline

1. Introduction to Global/Local Learning
2. Local Learning by **Data Partitioning**
   2.1 Learning Convex Combinations of Local Metrics
   "Metric learning as convex combinations of local models with generalization guarantees."
   2.2 Decentralized Adaboosting of Personalized Models
   "Decentralized Frank-Wolfe Boosting for Collaborative Learning of Personalized Models."
3. Local Learning using **Landmark Similarities**
   3.1 Landmark Support Vectors Machines
   "L^3-SVMs: Landmark-based Linear Local Support Vectors Machines."
4. Conclusion and Perspectives

Limitations of Global Learning

Learning linear models $f(x) = \text{sign}(\langle \theta, x \rangle + b)$



- $+$ great scalability at training and test time w.r.t. m (# examples) and d (# features)
- $-$ cannot capture complex distributions

Local Learning

how to capture local characteristics of the space?

- + keep scalability at training and test time w.r.t. m and d
- + capture complex distributions

local consistency: consistent predictions for similar points

Local Learning

how to capture local characteristics of the space?

+ keep scalability at training and test time w.r.t. m and d
+ capture complex distributions

local consistency: consistent predictions for similar points

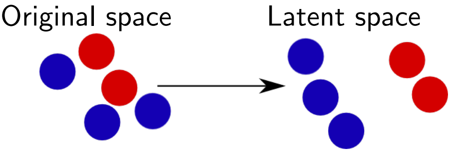
1. partition the data and learn a model per subset of data
   $\rightarrow$ learn multiple linear models
   - how to partition the data?
   - how to learn the single models?
2. compare the instances to a set of points spread over the space
   $\rightarrow$ learn a single linear model on a new representation
   - how to select the landmarks?
   - how to perform the comparisons?

Outline

C2LM: Learning Convex Combinations of Local Metrics

Metric Learning

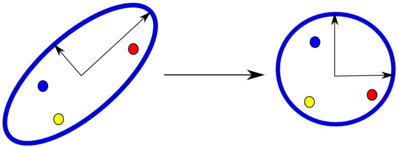
learn a **metric** (distance or similarity) adapted to the task



Example: Mahalanobis-like distance

$$d_A(x_1, x_2) = \sqrt{(x_1 - x_2)^T A (x_1 - x_2)}$$

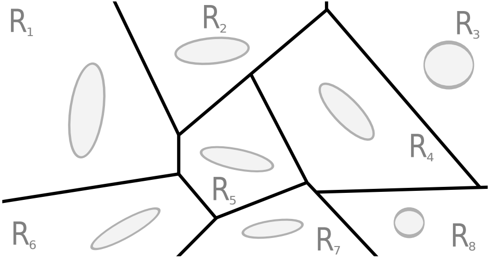
with PSD matrix $A \in \mathbb{R}^{d^2}$ of parameters



C2LM: Learning Convex Combinations of Local Metrics

Local Metric Learning

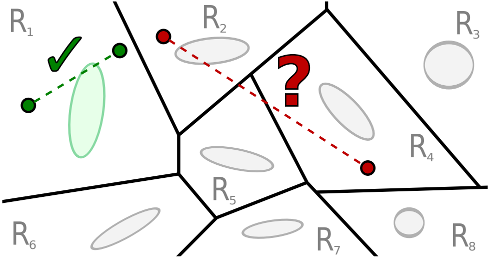
naive solution: learn a set of local metrics, one per region



C2LM: Learning Convex Combinations of Local Metrics

Local Metric Learning

naive solution: learn a set of local metrics, one per region



- loss of smoothness in prediction
- high risk of over-fitting the local set
- overall model is locally but not globally stationary
- how to compare instances from different regions?

C2LM: Learning Convex Combinations of Local Metrics

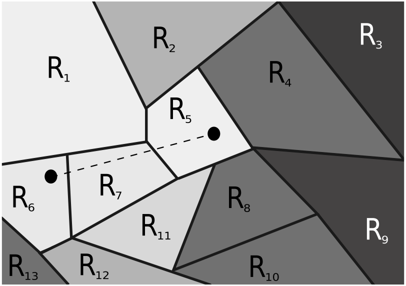
$\forall$ pair of regions (R_i, R_j) we define $t_{ij}(x_1, x_2)$ and learn $\alpha_{ij} \in \mathbb{R}^K$

$$t_{ij}(x_1, x_2) \; = \; \sum_{k=1}^{K} \alpha_{ijk} \; s_k(x_1, x_2)$$

i $\alpha_{ij} = \alpha_{ji}$ (symmetry)

ii $\forall k, \alpha_{ijk} \geq 0$ (positivity)

iii $\sum_{k=1}^{K} \alpha_{ijk} = 1$ (convexity)



α_{ijk}: influence of local metric s_k for pair of regions (R_i, R_j)

C2LM: Learning Convex Combinations of Local Metrics

Optimization Problem

$$\underset{\alpha \in \mathbb{R}^{K^3}}{\arg\min} \quad \frac{1}{m} \sum_{i=1,j=1}^{K,i} \sum_{(x_1,x_2)\in R_{ij}} \left| \sum_{k=1}^{K} \alpha_{ijk} s_k(x_1,x_2) - y(x_1,x_2) \right| + \lambda_1 D(\alpha) + \lambda_2 S(\alpha)$$

$$s.t. \quad \forall i,j : \sum_{k=1}^{K} \alpha_{ijk} = 1 \text{ and } \alpha_{ij} \geq 0$$

→ loss minimization: least absolute regression

→ cluster distance regularization

→ vector similarity regularization

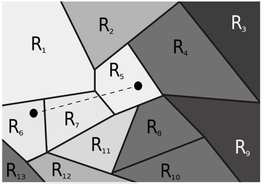
C2LM: Learning Convex Combinations of Local Metrics

Regularization Terms

considering the topological characteristics of the input space

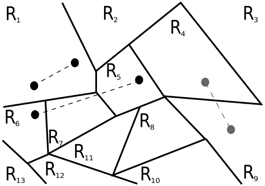
cluster distance regularization

$$D(\alpha) = \sum_{i=1,j=1}^{K,i} \sum_{k=1}^{K} (E_{ijk}\alpha_{ijk})^2$$



vector similarity regularization

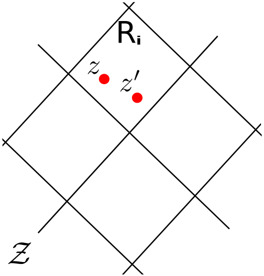
$$S(\alpha) = \sum_{i=1,j=1}^{K,i} \sum_{i'=1,j'=1}^{K,i'} W_{iji'j'} \left\|\alpha_{ij} - \alpha_{i'j'}\right\|_2^2$$



Generalization Guarantees

Algorithmic Robustness Framework [Xu and Mannor, 2012]

does f have similar predictions on $z \in S_{train}$ and on $z' \in S_{test}$?



Steps for deriving the bound:

- derive **convering number** of space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
- prove **Lipschitz continuity** of loss ℓ
- apply a **concentration inequality** to bound $R_{\mathcal{D}} - \hat{R}_S$

Generalization Guarantees

Algorithmic Robustness Bound

with probability at least $1 - \delta$, for the learned α

$$|R_{\mathcal{D}}(\alpha) - \hat{R}_S(\alpha)| \leq O\left(\gamma + \sqrt{\frac{K + \ln 1/\delta}{m}}\right)$$

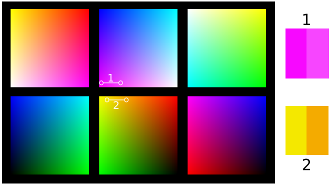
- true risk on the underlying distribution $\mathcal{D}$
- empirical risk on the training sample S
- generalization gap with
  γ = the maximal diameter of the clusters

$$\underset{\alpha \in \mathbb{R}^{K^3}}{\arg\min} \quad \frac{1}{m} \sum_{i=1,j=1}^{K,i} \sum_{(x_1,x_2) \in R_{ij}} \left| \sum_{k=1}^{K} \alpha_{ijk} s_k(x_1,x_2) - y(x_1,x_2) \right| + \lambda_1 D(\alpha) + \lambda_2 S(\alpha)$$

$$s.t. \quad \forall i,j : \sum_{k=1}^{K} \alpha_{ijk} = 1 \; and \; \alpha_{ij} \geq 0$$

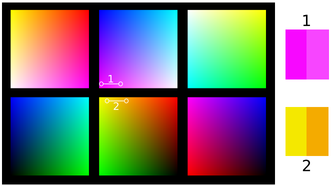
Experiments on Perceptual Color Distance

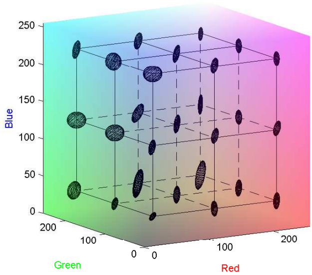
euclidean distance on RGB cube does not correspond to the distance perceived by humans



Experiments on Perceptual Color Distance

euclidean distance on RGB cube does not correspond to the distance perceived by humans





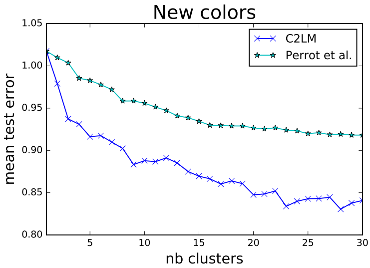
Experiments on Perceptual Color Distance

Dataset clustered using K-means

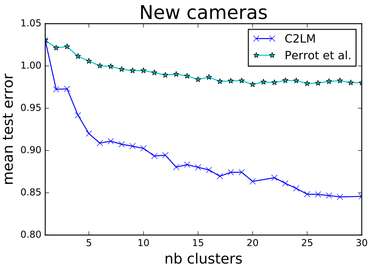
- 41800 pairs of color patches, taken under several viewing conditions with their reference perceptual distance ΔE_{00}
- 4 cameras

State of the art

- Local Metric Learning [Perrot et al., 2014]



6-fold cross-validation of the color patches



leave one camera out cross-validation

Outline

1. Introduction to Global/Local Learning

2. Local Learning by **Data Partitioning**
   2.1 Learning Convex Combinations of Local Metrics
   "Metric learning as convex combinations of local models with generalization guarantees."
   2.2 Decentralized Adaboosting of Personalized Models
   "Decentralized Frank-Wolfe Boosting for Collaborative Learning of Personalized Models."

3. Local Learning using **Landmark Similarities**
   3.1 Landmark Support Vectors Machines
   "L^3-SVMs: Landmark-based Linear Local Support Vectors Machines."

4. Conclusion and Perspectives

Dada: Decentralized Adaboost of Personalized Models

context

personal data = generated by a set of K users

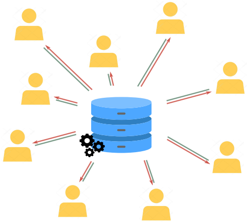
sample S is partitioned by user into $\{S_k\}_{k=1}^{K}$

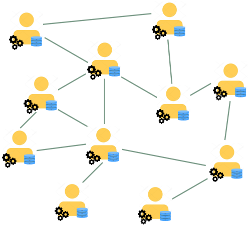
Dada: Decentralized Adaboost of Personalized Models

context

personal data = generated by a set of K users

sample S is partitioned by user into $\{S_k\}_{k=1}^{K}$





+ better reliability
+ harder to attack
+ easier to ensure privacy

– communication complexity is a bottleneck
  → focus on **sparsity**

Dada: Decentralized Adaboost of Personalized Models

Objectives

1. learn local (personalized) models
2. harness similarities between users
3. enforce smoothness in prediction

Dada: Decentralized Adaboost of Personalized Models

Objectives

1. learn local (personalized) models
2. harness similarities between users
3. enforce smoothness in prediction

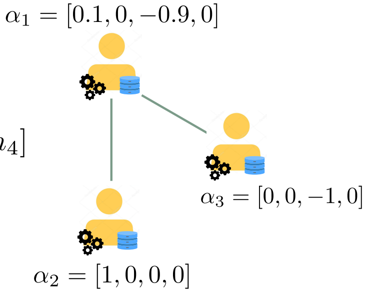
undirected and weighted collaboration graph $\mathcal{G} = (V, E, W)$

- V is the set of K users or nodes
- E is the set of M edges
- each agent k is connected to a subset $N_k \subseteq V$
- $W \in \mathbb{R}^{K^2}$ is the similarity matrix
  $\rightarrow W_{kl}$ describes the similarity between user k and user l

Dada: Decentralized Adaboost of Personalized Models

- given a fixed set of n base functions $H = \{h_j : \mathcal{X} \to \mathbb{R}\}_{j=1}^n$
- learn a set of local vectors $\{\alpha_k \in \mathbb{R}^n\}_{k=1}^K$
  α_{kj} is the weight of user k associated with the base function h_j
- to obtain binary classifiers by weighted majority vote
  $x \mapsto \text{sign}[\sum_{j=1}^n \alpha_{kj} h_j(x)]$

$H = [h_1, h_2, h_3, h_4]$



Dada: Decentralized Adaboost of Personalized Models

Optimization Problem

$$\min_{\alpha \in \mathbb{R}^{Kn}} \sum_{k=1}^{K} D_k c_k \log \left(\sum_{i=1}^{m_k} \exp\left(-(A_k \alpha_k)_i\right) \right) + \frac{\mu}{2} \sum_{k=1}^{K} \sum_{l=1}^{k-1} W_{kl} \left\| \alpha_k - \alpha_l \right\|_2^2$$

$$s.t. \quad \forall k : \|\alpha_k\|_1 \leq \beta$$

→ local loss minimization of node k

- D_k is its degree
- c_k is its confidence (proportional to m_k)
- $A_k \in \mathbb{R}^{m_k \times n}$ is its margin matrix of entries $a_i j = y_i h_j(x_i)$

→ vector similarity regularization

- smoothness in prediction
- communication with direct neighbors

→ sparsity constraint

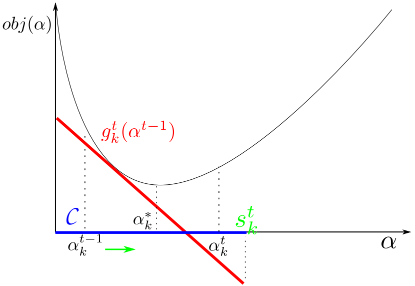
Dada: Decentralized Adaboost of Personalized Models

Frank-Wolfe Optimization [Frank and Wolfe, 1956]

Block-coordinate descent: optimize over one α_k at each iteration

ensure sparse updates

- **only one coordinate** α_{kj} updated at a time
- only $O(|N_k| \log n)$ communications per update



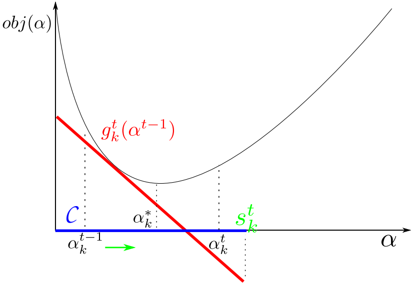
Dada: Decentralized Adaboost of Personalized Models

Frank-Wolfe Optimization [Frank and Wolfe, 1956]

Block-coordinate descent: optimize over one α_k at each iteration

ensure sparse updates

- **only one coordinate** α_{kj} updated at a time
- only $O(|N_k| \log n)$ communications per update



solve a linearization of the problem over $\mathcal{C} = \|\alpha_k\|_1 \leq \beta$:

$$s_k^{(t)} = \underset{\|s\|_1 \leq \beta}{\arg\min} \ \langle s, g_k^{(t)} \rangle$$

$$g_k^{(t)} = -D_k c_k \eta_k^T A_k + \mu (D_k \alpha_k^{(t-1)} - \sum_l W_{kl} \alpha_l^{(t-1)}) \ ; \quad \eta_k = \frac{\exp(-A_k \alpha_k^{(t-1)})}{\sum_{i=1}^{m_k} \exp(-A_k \alpha_k^{(t-1)})_i}$$

Theoretical Analysis

for K users, T iterations, n base functions and M edges

Convergence Rate

Dada converges in expectation with a rate $O\left(\frac{K}{T}\right)$

Communication Complexity

Dada has a communication complexity of $O\left(T \log n \frac{M}{K}\right)$

To recapitulate

+ improve discriminative power of local models
+ avoid over-fitting
+ achieve smoothness in prediction

	C2LM	Dada
Setting	regression	classification
Partition by	**features**	**user**
Learn combinations of	local models	base functions
Smoothing regularization term	**similarity graph**	
Other regularizations	topology of input space	sparsity

– learn multiple models
– rely on the goodness of the hard partition
– need to estimate the similarity matrix W
  $\rightarrow$ either by using prior-knowledge or by optimizing it

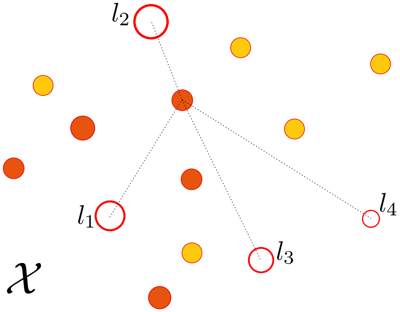
Outline

Local Learning using Landmark Similarities

optimize a single model capable of extracting the local characteristics and evolving smoothly over the distribution

Definition

(**Landmarks**) The set of landmarks $\mathcal{L}$ is a set of points $\{l_p \in \mathcal{X}\}_{p=1}^{L}$ used to create a new representation $\mathcal{H}$.



Similarity principle:
$\forall x \in S$ described using $\mathcal{L}$ and μ

$$\mu_{\mathcal{L}}(.) = [\mu(., l_1), ..., \mu(., l_L)]$$

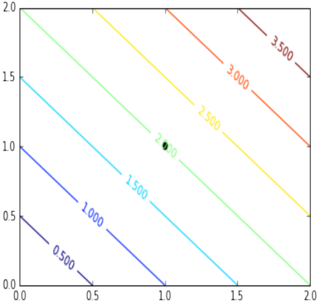
explicit mapping from $\mathcal{X}$ to $\mathcal{H}$

Local Learning using Landmark Similarities

examples of similarity functions

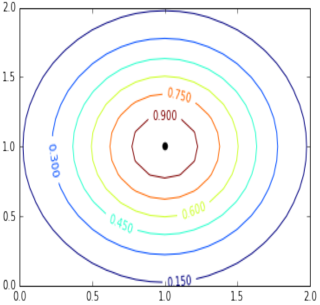
For a given $x \in \mathcal{X}$ and $\forall\ x_1 \in \mathcal{X}$:

Linear kernel



$$\mu(x, x_1) = \langle x, x_1 \rangle$$

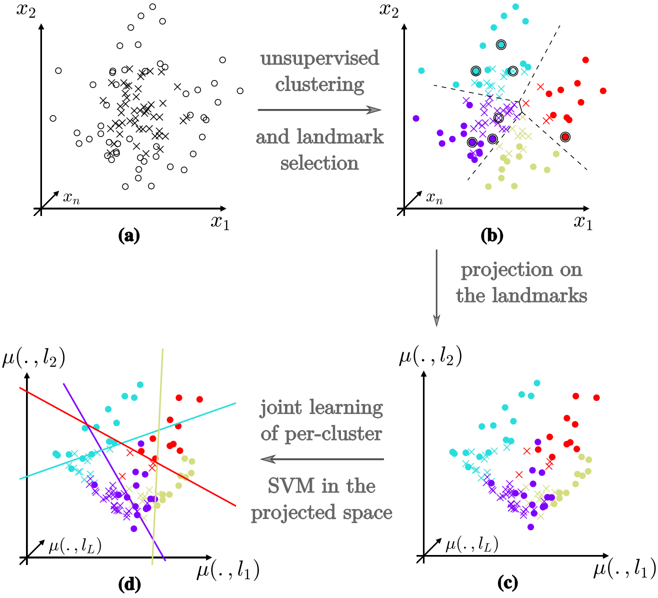
Radial Basis Function RBF



Given $\gamma \in \mathbb{R}^+$,

$$\mu(x, x_1) = \exp\left(-\frac{\|x - x_1\|_2^2}{\gamma}\right)$$

L^3-SVMs: Landmark-based Support Vector Machines



L^3-SVMs: Landmark-based Support Vector Machines

Optimization Problem

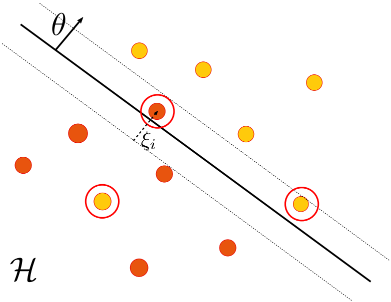
learn a linear Support Vector Machines on the latent space $\mathcal{H}$

$$\arg\min_{\theta,b,\xi} \frac{1}{2}\|\theta\|_2^2 + \frac{\lambda}{m}\sum_{i=1}^{m}\xi_i$$

$$s.t. \quad y_i\left(\theta_{k_i.}\mu_{\mathcal{L}}(x_i)^T + b\right) \geq 1 - \xi_i \ \forall i = 1..m$$

$$\xi_i \geq 0 \ \forall i = 1..m$$

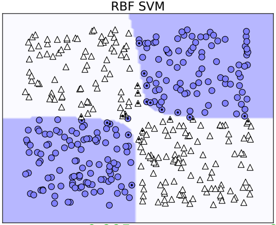
1. projection:
   $\mu_{\mathcal{L}}(.) = [\mu(., l_1), ..., \mu(., l_L)] \in \mathbb{R}^L$
2. clustering: $z_i = (x_i, y_i, k_i)$
3. training: $\theta \in \mathbb{R}^{KL}, b \in \mathbb{R}$



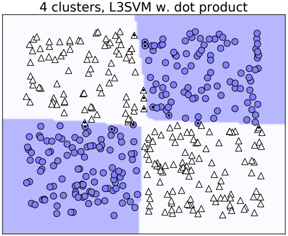
Experiments on Synthetic Data

capturing non-linearities

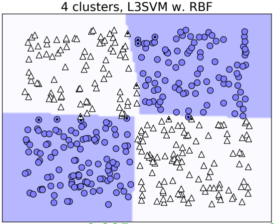
10 landmarks uniformly drawn from S



train accuracy = 0.995, test accuracy = 0.9725
nb support vectors = 26



train accuracy = 0.9925, test accuracy = 0.975
nb support vectors = 14



train accuracy = 0.995, test accuracy = 0.9725
nb support vectors = 13

Generalization Guarantees

Uniform Stability framework [Xu and Mannor, 2012]

does f_S learned from S is similar to $f_{S'}$ learned from S'?

$$S = \{z_1, \ldots, z_i, \ldots, z_m\} \qquad S' = \{z_1, \ldots, {z_i}', \ldots, z_m\}$$

S and S' differ for one instance.

Steps for deriving the bound:

- derive **stability constant** of the problem w.r.t. ℓ
- prove σ-**admissibility** of loss ℓ
- apply a **concentration inequality** to bound $R_{\mathcal{D}} - \hat{R}_S$

Generalization Guarantees

Uniform Stability bound

with probability at least $1 - \delta$ and learned model $f = (\theta, b)$

$$R_{\mathcal{D}}(f) \leq \hat{R}_S(f) + O\left(\lambda M \sqrt{\frac{L}{m} \ln \frac{1}{\delta}}\right) \tag{1}$$

- true risk on the underlying distribution $\mathcal{D}$
- empirical on the training sample S
- generalization gap with $M = \max_{x \in S, l_p \in \mathcal{L}} \mu(x, l_p)$

$$\underset{\theta, b, \xi}{\arg\min} \frac{1}{2} \|\theta\|_2^2 + \frac{\lambda}{m} \sum_{i=1}^{m} \xi_i$$

$$s.t. \;\; y_i \left(\theta_{k_{i,\cdot}} \mu_{\mathcal{L}}(x_i)^T + b\right) \geq 1 - \xi_i \; ; \; \xi_i \geq 0 \;\; \forall i = 1..m$$

Outline

1. Introduction to Global/Local Learning

2. Local Learning by **Data Partitioning**
   2.1 Learning Convex Combinations of Local Metrics
   "Metric learning as convex combinations of local models with generalization guarantees."
   2.2 Decentralized Adaboosting of Personalized Models
   "Decentralized Frank-Wolfe Boosting for Collaborative Learning of Personalized Models."

3. Local Learning using **Landmark Similarities**
   3.1 Landmark Support Vectors Machines
   "L^3-SVMs: Landmark-based Linear Local Support Vectors Machines."

4. Conclusion and Perspectives

Conclusion

what I presented

Unified view of Local Learning

1. partition the data and learn a model per subset of data
   $\rightarrow$ learn multiple linear models
   - how to partition the data?
   - how to learn the single models?
2. compare the instances to a set of points spread over the space
   $\rightarrow$ learn single linear model on a new representation
   - how to select the landmarks?
   - how to perform the comparisons?

	Data Partitioning	Landmark Similarities
Smoothing regularization term	required	not required
Stationarity	local	local and global
Learn multiple models	required	not required
Define latent space	not required	required
Adapted to decentralized learning	yes	no

Conclusion

what I did not present

1. application of **C2LM** to word similarity estimation
2. graph optimization for **Dada**
3. extension of **L^3-SVMs** to multi-view data
4. works on learning from weakly-labeled data
5. works on adversarial robustness of Deep Neural Networks

Perspectives

smoothing regularization

Optimization of similarity graph for Dada

1. allow for heterogeneous weights
2. enforce connectivity

Following [Kalofolias, 2016],

$$\min_{\alpha, W} \sum_{k=1}^{K} D_k c_k \mathcal{L}_k(\alpha_k; S_k) + \frac{\mu}{2} \sum_{k<l} W_{kl} \|\alpha_k - \alpha_l\|^2 - \nu 1^T \log(D + \delta) + \lambda \|W\|_{\mathcal{F}}^2$$

Perspective: optimize hyperbolic random graphs

Perspectives

landmark selection

Principal questions

1. how many landmarks are sufficient for the task?
2. how should they be selected?

Following [Yu et al., 2009],

$$L \propto \text{intrinsic dimensionality of the manifold of } \mathcal{D}$$

Following [Balcan et al., 2008],

$$L \propto \text{intrinsic complexity of } \mathcal{D}$$

Perspectives

landmark selection

The set of landmarks $\mathcal{L}$ should be

- minimal for scalability
- representative of the task for accuracy

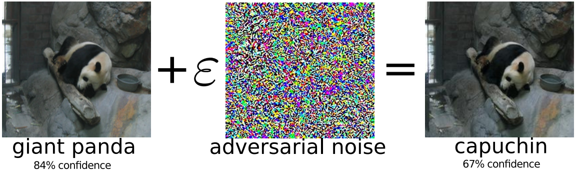
Derivation of generalization bounds dependent on task complexity and class complexity (estimated through $\mathcal{L}$)

$$\mathbb{P}\left(\left|R_{\mathcal{D}} - \hat{R}_S\right| \geq O(\text{class complexity}, \text{task complexity}, m)\right) \leq 1 - \delta.$$

Perspectives

adversarial robustness

$$\min_{\|\Delta x\| \leq r} f(x + \Delta x) \neq f(x).$$



$\|\Delta x\| \leq r$ is a bad criterion:

- all perturbations are equally accounted for
- leads to accuracy loss

Perspectives

adversarial robustness

1. investigate robustness of approaches based on latent space:
   - generative models
   - RBF nets
2. investigate advantages of disentangled features:
   - allow for considering a feature at a time
   - easier to study error propagation
   - may be easier to defend

Thank you for your attention!

International Conferences

- Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. "Fast and Provably Effective Multi-view Classification with Landmark-based SVM." (ECML PKDD), 2018 [Zantedeschi et al., 2018b].
- Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. "Beta-risk: a new surrogate risk for learning from weakly labeled data." (NeurIPS), 2016 [Zantedeschi et al., 2016b].
- Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. "Metric learning as convex combinations of local models with generalization guarantees." (CVPR), 2016 [Zantedeschi et al., 2016d].

National Conferences

- Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. "Decentralized Frank-Wolfe Boosting for Collaborative Learning of Personalized Models." (CAp), 2018 [Zantedeschi et al., 2018a].
- Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. "L^3-SVMs: Landmarks-based Linear Local Support Vectors Machines." (CAp), 2017 [Zantedeschi et al., 2017a].
- Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. "Apprentissage de Combinaisons Convexes de Métriques Locales avec Garanties de Généralisation." (CAp), 2016 [Zantedeschi et al., 2016a].

International Workshops

- Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. "Communication-Efficient Decentralized Boosting while Discovering the Collaboration Graph." (MLPCD 2), 2018.
- Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. "Efficient defenses against adversarial attacks." (AISEC), 2017 [Zantedeschi et al., 2017b].

Open-Source Software

- "Adversarial Robustness Toolbox", Python [Nicolae et al., 2018]
  `https://github.com/IBM/adversarial-robustness-toolbox`
- and others...

Johnson-Lindenstrauss Projections

Lemma

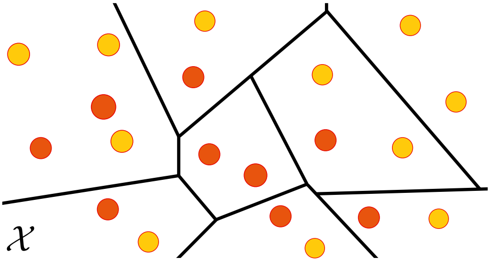
Let a set of points $S = \{x_i \in \mathbb{R}^d\}_{i=1}^m$, *a constant* $\varepsilon \in]0,1[$ *and a number* $L > 8\frac{\log(m)}{\varepsilon^2}$, $\exists$ *a linear projection* $f : \mathbb{R}^d \rightarrow \mathbb{R}^L$ *such that:*

$$(1-\epsilon)\left\|x_i - x_{i'}\right\| \leq \left\|f(x_i) - f(x_{i'})\right\| \leq (1+\epsilon)\left\|x_i - x_{i'}\right\|.$$

	JL	**L^3-SVMs**
supervision	none	none
projection	random	through similarity
	linear	any
distance preservation	yes	not necessarily
task linearization	no	yes
dimensionality reduction	$L = O(\frac{\log(m)}{\varepsilon^2})$	$L = ?$

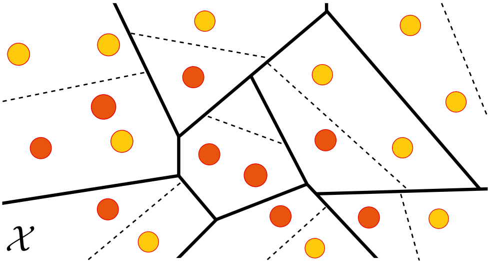
Approach 1: Divide and Conquer

1. partition the data into K clusters $\{R_k\}_{k=1}^{K}$



Approach 1: Divide and Conquer

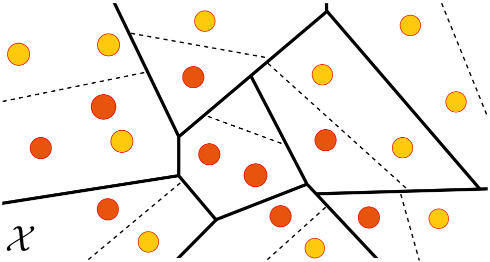
1. partition the data into K clusters $\{R_k\}_{k=1}^{K}$
2. learn a linear model per subgroup $\{s_k(.)\}_{k=1}^{K}$



Approach 1: Divide and Conquer

1. partition the data into K clusters $\{R_k\}_{k=1}^K$
2. learn a linear model per subgroup $\{s_k(.)\}_{k=1}^K$

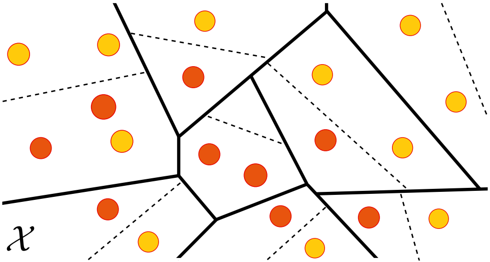
Possible criteria: **spatial**, class, **meta-data**, etc.



Approach 1: Divide and Conquer

Drawbacks:

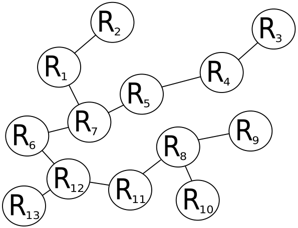
- loss of smoothness in prediction
- high risk of over-fitting the local set
- overall model is stationary on each subset individually but not globally



C2LM: Learning Convex Combinations of Local Metrics

Regularization Terms

considering the topological characteristics of the input space



Minimum Spanning Tree

d_{ij} = number of edges of shortest path between R_i and R_j

$$E_{ijk} = d_{ik} + d_{jk}$$

$$W_{iji'j'} = \exp\left[-min(d_{ii'} + d_{jj'}, d_{ij'} + d_{i'j})\right]$$

ex.

$$E_{567} = 2, E_{569} = 10$$

$$W_{56,77} = e^{-2}, W_{56,89} = e^{-9}$$

Generalization Guarantees

Algorithmic Robustness Bound

For any $\delta > 0$ with probability at least $1 - \delta$, we have:

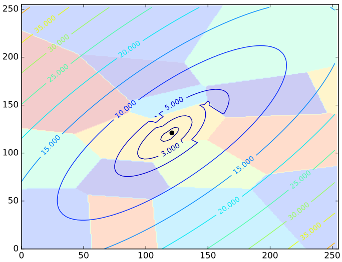
$$|R_{\mathcal{D}}(\alpha) - \hat{R}_S(\alpha)| \leq \theta\sqrt{2}\gamma_1 + \gamma_2 + B\sqrt{\frac{2H \ln 2 + 2 \ln 1/\delta}{m}} .$$

covering number $H = \mathcal{N}(\gamma_1/2, U, \|.\|_2)\mathcal{N}(\gamma_2/2, Y, |.|)$

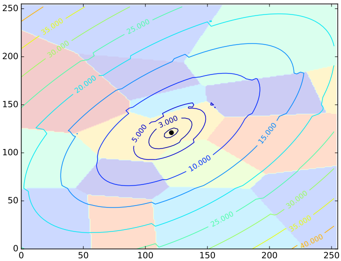
Experiments on Perceptual Color Distance

section from the RGB cube

distance levels from a given center (the dot)
clusters are marked by colors



Set of local models + one global

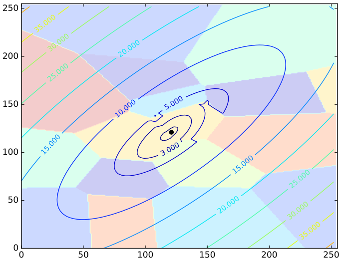


C2LM

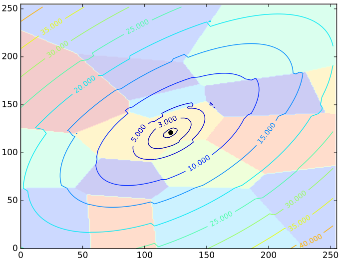
Experiments on Perceptual Color Distance

section from the RGB cube

+ better estimation of the distance



Set of local models + one global

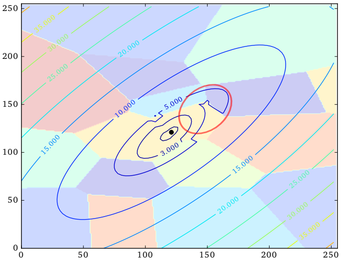


C2LM

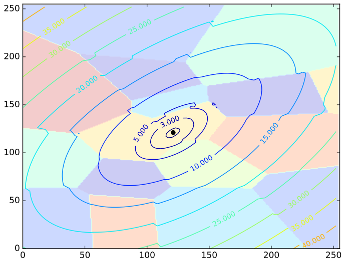
Experiments on Perceptual Color Distance

section from the RGB cube

+ better estimation of the distance
+ better smoothness in prediction

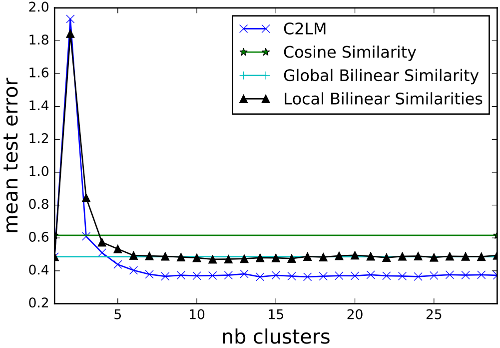


Set of local models + one global



C2LM

Experiments on Perceptual Color Distance



Dada: Decentralized Adaboost of Personalized Models

Frank-Wolfe Optimization

iterative algorithm over T iterations

Algorithm 1 iterative algorithms over T iterations

1: initialize $\{\alpha_k\}_{k=1}^{K}$ to 0
2: **for** $t = 1$ to T **do**
3: draw k uniformly from $\{1, \ldots, K\}$
4: update α_k following

$$\alpha_k^{(t)} = (1 - \gamma^{(t)})\alpha_k^{(t-1)} + \gamma^{(t)}\, s_k^{(t)}$$

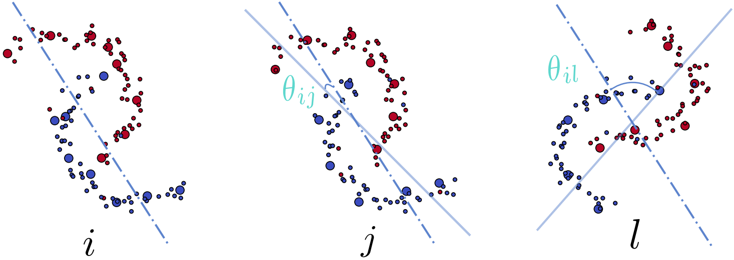
$$\text{where } s_k^{(t)} = \beta \,\text{sign}(-(g_k^{(t)})_j) e^{j_k^{(t)}} \text{ and } \gamma^{(t)} = \frac{2K}{t + 2K}$$

5: agent k sends $\alpha_k^{(t)}$ to its neighborhood N_k.
6: **end for**

Experiments on Synthetic Data

Dataset

points drawn from the two interleaving Moons dataset and rotated following a local axis:

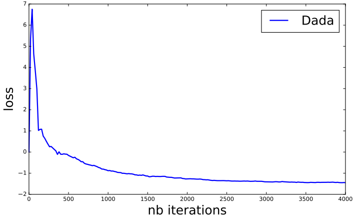


- $K = 100$ or $K = 20$ agents with a randomly drawn rotation axis each;
- $W_{ij} = \exp(10\cos(\theta_{ij}) - 1)$
- $d = 20$ total dimensions

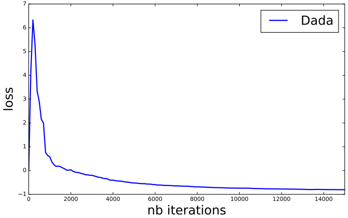
Experiments on Synthetic Data

Baselines

- Personalized linear [Vanhaesebrouck et al., 2017]
- Adaboost based: global l_1, global-local mixture, purely local $\rightarrow \boldsymbol{n = 200}$ decision stumps uniformly spread over the dimensions

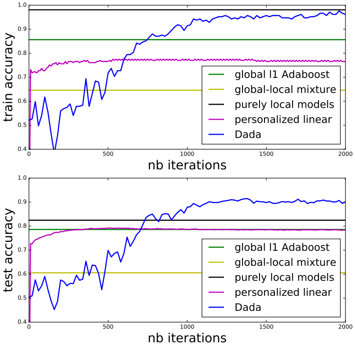


$K = 20$

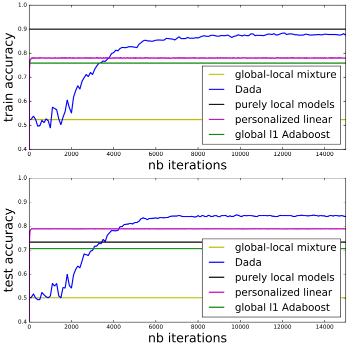


$K = 100$

Experiments on Synthetic Data



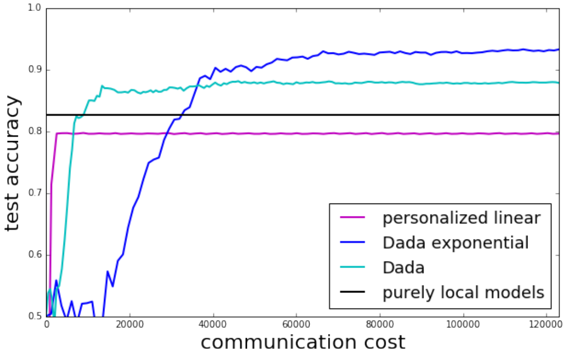
$K = 20$



$K = 100$

Experiments on Synthetic Data

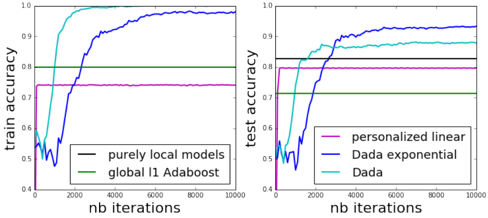
communication

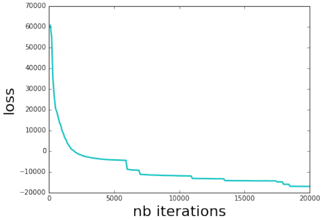


$K = 100$

Experiments on Synthetic Data

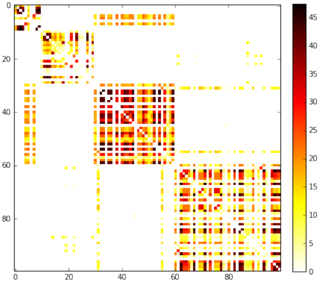
graph optimization

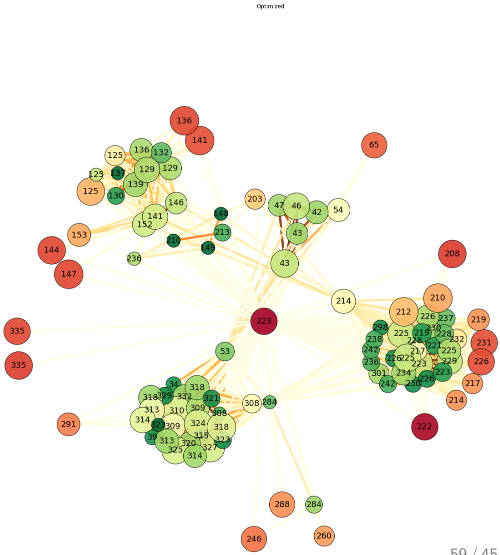




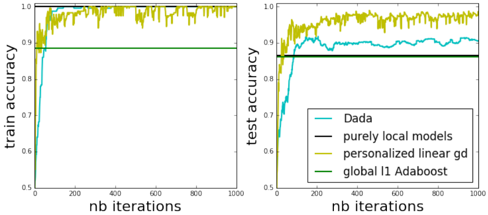
Experiments on Synthetic Data

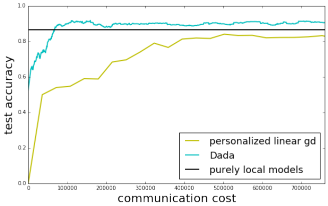
graph optimization





Experiments on Activity Recognition

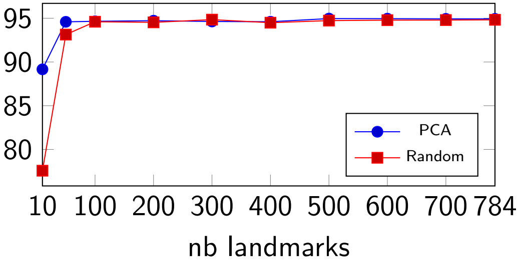




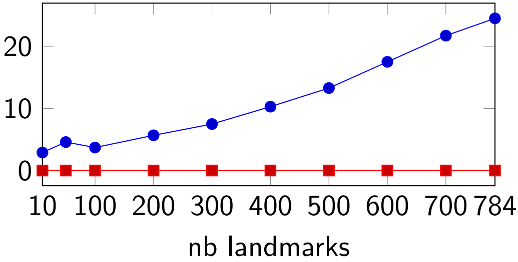
Experiments on MNIST

landmark selection

Test Accuracy (%)

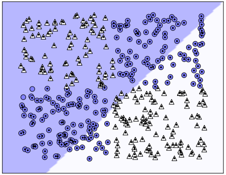


Selection Time (s)



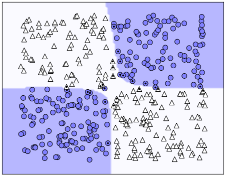
XOR Distribution

Linear SVM



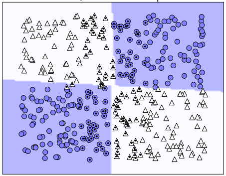
train accuracy = 0.645, test accuracy = 0.585
nb support vectors = 397

RBF SVM



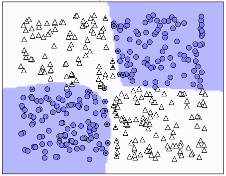
train accuracy = 0.995, test accuracy = 0.9725
nb support vectors = 26

2 clusters, L3SVM w. dot product



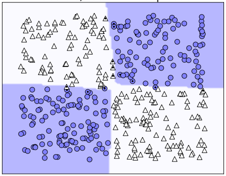
train accuracy = 0.9925, test accuracy = 0.97375
nb support vectors = 141

2 clusters, L3SVM w. RBF



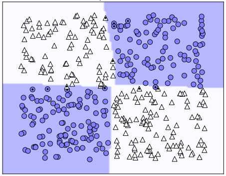
train accuracy = 0.99, test accuracy = 0.965
nb support vectors = 26

4 clusters, L3SVM w. dot product



train accuracy = 0.9925, test accuracy = 0.975
nb support vectors = 14

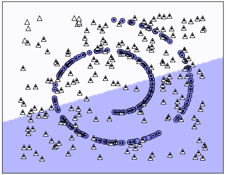
4 clusters, L3SVM w. RBF



train accuracy = 0.995, test accuracy = 0.9725
nb support vectors = 13

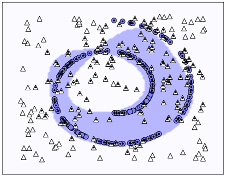
Swissroll Distribution

Linear SVM



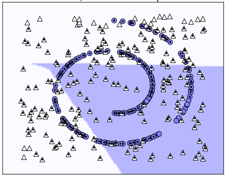
train accuracy = 0.575, test accuracy = 0.52375
nb support vectors = 384

RBF SVM



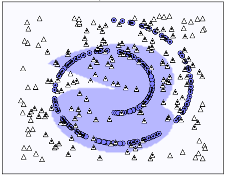
train accuracy = 0.7425, test accuracy = 0.72125
nb support vectors = 296

2 clusters, L3SVM w. dot product



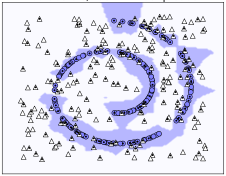
train accuracy = 0.5875, test accuracy = 0.52375
nb support vectors = 350

2 clusters, L3SVM w. RBF



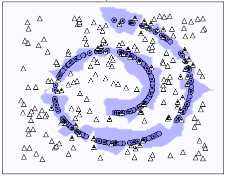
train accuracy = 0.69, test accuracy = 0.6575
nb support vectors = 300

100 clusters, L3SVM w. dot product



train accuracy = 0.8725, test accuracy = 0.82625
nb support vectors = 217

100 clusters, L3SVM w. RBF



train accuracy = 0.905, test accuracy = 0.8525
nb support vectors = 171

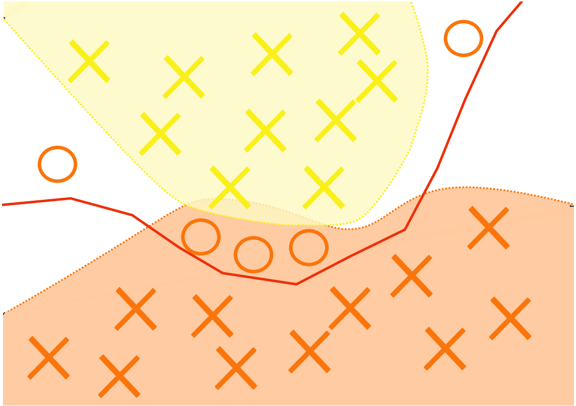
Experiments on Real Datasets

	#training	#testing	#features	#classes	#models
SVMGUIDE1	3089	4000	4	2	100
IJCNN1	49990	91701	22	2	100
USPS	7291	2007	256	10	80
MNIST	60000	10000	784	10	90
PASCAL VOC 2007	5011	4952	4096	20	20

	SVMGUIDE1		**IJCNN1**		**USPS**		**MNIST**		**PASCAL VOC**	
RBF-SVM	96.53	*1*×	97.08	*1*×	94.07	*1*×	96.62	*1*×	96.9	*1*×
Poly-SVM	96.35	*2.1*×	92.65	*5.2*×	N/A	*N/A*	N/A	*N/A*	N/A	*N/A*
Linear-SVM	95.38	*9.8*×	89.68	*140.5*×	91.72	*30.6*×	91.8	*112.5*×	96.7	*12.1*×
CSVM	95.05	*0.3*×	96.35	*45.2*×	N/A	*N/A*	N/A	*N/A*	N/A	*N/A*
LLSVM	94.08	*1.7*×	92.93	*16.8*×	75.69	*0.4*×	88.65	*1.9*×	N/A	*N/A*
ML3	96.68	*0.3*×	97.73	*5.9*×	93.22	*1.1*×	97.04	*2.1*×	96.5	*17.7*×
L^3-SVMs	95.73	*1.8*×	95.74	*7.4*×	92.12	*1.3*×	95.05	*9.8*×	96.7	*19.2*×

Table: Testing Accuracies (%) and Training Speedups w.r.t. RBF-SVM.

Adversarial Examples



References I

Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved guarantees for learning via similarity functions. *Computer Science Department*, page 126, 2008.

Olivier Bousquet and André Elisseeff. Stability and generalization. volume 2, pages 499–526. JMLR. org, 2002.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. volume 3, pages 95–110, 1956.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *AISTATS*, 2016.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Ian M Molloy, and Ben Edwards. Adversarial robustness toolbox v0. 2.2. *arXiv preprint arXiv:1807.01069*, 2018.

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modeling perceptual color differences by local metric learning. In *European conference on computer vision*, pages 96–111. Springer, 2014.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized Collaborative Learning of Personalized Models over Networks. In *AISTATS*, 2017.

Huan Xu and Shie Mannor. Robustness and generalization. volume 86, pages 391–423. Springer, 2012.

Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2223–2231. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3875-nonlinear-learning-using-local-coordinate-coding.pdf`.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Apprentissage de combinaisons convexes de métriques locales avec garanties de généralisation. In *CAp2016*, 2016a.

References II

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Beta-risk: a new surrogate risk for learning from weakly labeled data. In *Advances in Neural Information Processing Systems*, pages 4365–4373, 2016b.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Lipschitz continuity of mahalanobis distances and bilinear forms. 2016c.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Metric learning as convex combinations of local models with generalization guarantees. In *CVPR*, 2016d.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. L ^{3}-svms: Landmark-based linear local support vector machines. In *CAp*, 2017a.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49. ACM, 2017b.

Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. Decentralized Frank-Wolfe boosting for collaborative learning of personalized models. In *CAp*, 2018a.

Valentina Zantedeschi, Rémi Emonet, and Marc Sebban. Fast and provably effective multi-view classification with landmark-based svm. In *ECML PKDD*, 2018b.