Algorithms for a family of non-convex issues in machine learning

Gilles GASSO

LITIS *EA 4108*

Séminaire Gipsa-Lab

May 31, 2012

À taille humaine
à l'échelle du monde

1. Introduction
   - General learning problem
   - Discussion of convexity and non-convexity of learning problem
   - Multi-stage convex relaxation
2. Case study
   - Learning under probability constraint
     - Problem formulation
     - Algorithms
     - Empirical evaluation
   - Multitask learning
     - Joint sparsity penalization
     - MKL-MTL Algorithms

General framework

Learning problem

- Dataset $S = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$ i.i.d. sampled
- Goal: learn a functional relation $f : \mathcal{X} \to \mathcal{Y}$
- f belongs to space of functions $\mathcal{H}$
- Many learning problems come in the form

$$(P) \quad \min_{f \in \mathcal{C}} J(f, S) \quad \text{with} \quad J(f, S) = L(f, S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

- L: data fidelity cost, Ω: penalization term and $\lambda \geq 0$

Examples

$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

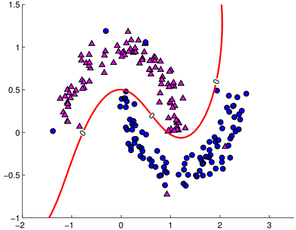
SVM for binary classification

- f: a non-linear function
- Hinge loss based data fidelity cost

$$L(f,S) = \sum_{i=1}^{n} \max(0, 1 - y_i f(\mathbf{x}_i))$$

- Smoothness penalization

$$\Omega(f) = \|f\|_{\mathcal{H}}^2$$



Examples

$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

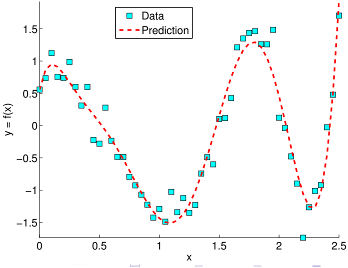
Regression

- $f(x) = \langle w, \phi(\mathbf{x})\rangle + b$
- Least squares loss

$$L(f,S) = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$

- Smoothness penalization

$$\Omega(f) = \|\mathbf{w}\|^2$$



Features of the learning problem

$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

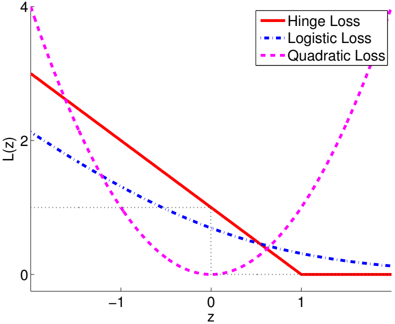
Convexity of Problem (P)

1. J is convex, and
2. Set $\mathcal{C}$ is convex

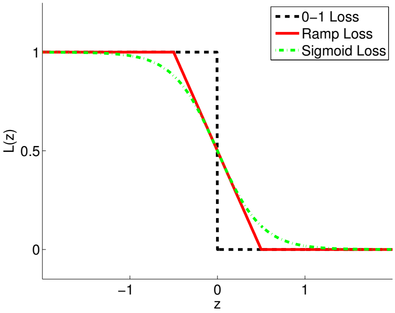
Non-convexity of (P)

1. Either J or $\mathcal{C}$ is non-convex

Convex loss function L



Non-Convex loss L



$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

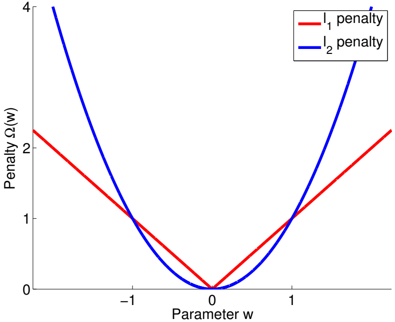
Convexity of Problem (P)

1. J is convex, and
2. Set $\mathcal{C}$ is convex

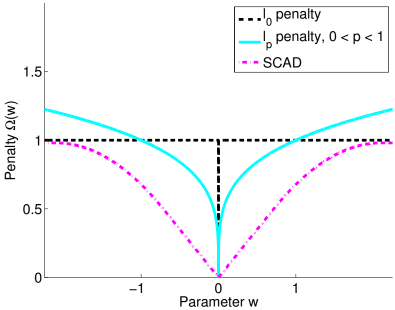
Non-convexity of (P)

1. Either J or $\mathcal{C}$ is non-convex

Convex Penalty Ω



Non-Convex Penalty Ω



Features of the learning problem

$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

Convexity of Problem (P)

1. J is convex, and
2. Set $\mathcal{C}$ is convex

Non-convexity of (P)

1. Either J or $\mathcal{C}$ is non-convex

Pros and Cons

- Any local solution is globally optimal
- Efficient computation
- Initialization does not matter

Pros and Cons

- Difficult to solve
- Find all local minima to get global solution
- Initialization really matters

Features of the learning problem

$$(P) \quad \min_{f \in \mathcal{C}} J(f,S) \quad \text{with} \quad J(f,S) = L(f,S) + \lambda\, \Omega(f), \quad \mathcal{C} \subseteq \mathcal{H}$$

Convexity of Problem (P)

1. J is convex, and
2. Set $\mathcal{C}$ is convex

Non-convexity of (P)

1. Either J or $\mathcal{C}$ is non-convex

Pros and Cons

- Any local solution is globally optimal
- Efficient computation
- Initialization does not matter

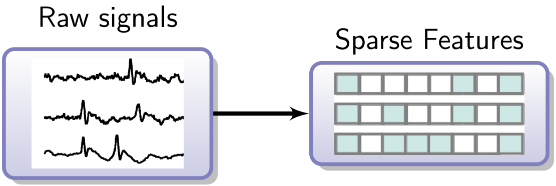
Pros and Cons

- Difficult to solve
- Find all local minima to get global solution
- Initialization really matters

Convexity of (P) is a blessing.
However non-convexity can pay off. Why to prefer it?

Motivation (1)

Sparse representation (Compressive sensing)



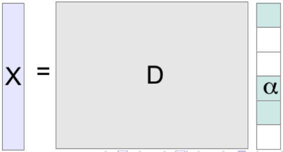
- Dictionary $D \in \mathbb{R}^{N \times d}$
- $N \ll d$ (more variables than samples)
- Signal $X \in \mathbb{R}^{N}$

Goal

Find a sparse decomposition of signal $X \in \mathbb{R}^{N}$ over D

Need of sparsity

- computation
- interpretation
- accuracy



Motivation (1)

$$\min_{\alpha \in \mathbb{R}^d} \|X - D\alpha\|^2 + \lambda \Omega(\alpha)$$

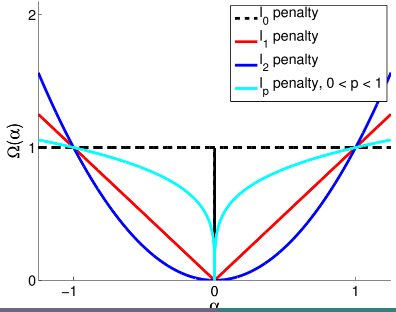
$\Omega(\alpha)$: sparsity inducing penalisation

Non-convex formulation

1. Count: $\Omega(\alpha) = \sum_{j=1}^{d} \mathbb{I}_{\alpha_j \neq 0}$
2. A Concave relaxation
   $\Omega(\alpha) = \sum_{j=1}^{d} |\alpha_j|^p$, $0 < p < 1$

Convex relaxation

1. ℓ_1-norm $\Omega(\alpha) = \|\alpha\|_1$
2. ℓ_2-norm $\Omega(\alpha) = \|\alpha\|_2^2$



- Convex formulations lead to biased estimation of α
- Concave relaxation: better approximation of $\|\cdot\|_0$

Dynamical system modelling under stability constraint

Input $u(t)$ $\longrightarrow$ **Black-Box model $\mathcal{M}$** $\longrightarrow$ Output $\hat{y}(t)$

Model

$$\begin{cases} X(t+1) = AX(t) + Bu(t) + \psi(t) \\ \hat{y}(t) = CX(t) + \varepsilon(t) \end{cases}$$

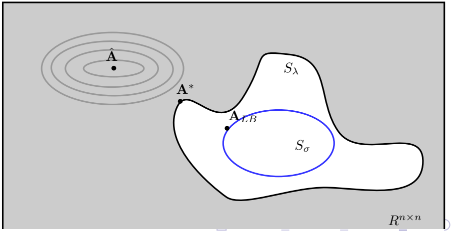
Learning Problem

Find A, B, C

s.t. A is stable

Non-Convex formulation	Convex relaxation
$\rho(A) \leq 1$	$\rho(A^\top A) \leq 1$

$\rho(M)$: spectral radius of M

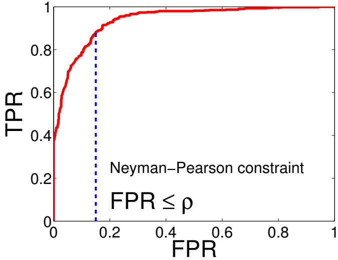


Neyman-Pearson classification
(Binary imbalanced classification)

Learning problem

Find decision function f

$$\max_{f} \quad \text{TPR}$$
$$\text{s.t.} \quad \text{FPR} \leq \rho$$



- TPR: True Positives Rate
- FPR : False Positives Rate

- Probability constraint is generally non-convex
- Convex relaxation is tedious

Convex or Non-convex?

- Convex problems not subject to initialization issue
- Efficient solver for convex problems
- Non-Convex problems difficult to solve ...
- ... but can provide better results if carefully solved

Adopted approach

- Solve efficiently the non-convex problem by successive refinements of convex relaxation
- Leverage convex solvers
- Handle non-smooth cases

Algorithm 1 Synopsis to solve $\min_{f \in \mathcal{C} \subseteq \mathcal{H}} J(f, S)$

Set $t = 0$, initialize f

repeat

Find J_{Conv} and $\mathcal{C}_{Conv}$, convex relaxations of J and $\mathcal{C}$ at f_t

Solve the convex problem $f_{t+1} = \text{argmin}_{f \in \mathcal{C}_{Conv}} J_{Conv}(f, S)$

until termination

Algorithm 2 Synopsis to solve $\min_{f \in \mathcal{C} \subseteq \mathcal{H}} J(f, S)$

Set $t = 0$, initialize f

repeat

Find J_{Conv} and $\mathcal{C}_{Conv}$, convex relaxations of J and $\mathcal{C}$ at f_t

Solve the convex problem $f_{t+1} = \text{argmin}_{f \in \mathcal{C}_{Conv}} J_{Conv}(f, S)$

until termination

How to find a convex relaxation?

- Majoration-Minimization [Wu, 2010]
  - DC (difference of convex functions) programming [Tao, 1998]
  - Concave relaxation [Zhan, 2010]

Algorithm 3 Synopsis to solve $\min_{f \in \mathcal{C} \subseteq \mathcal{H}} J(f, S)$

Set $t = 0$, initialize f

repeat

Find J_{Conv} and $\mathcal{C}_{Conv}$, convex relaxations of J and $\mathcal{C}$ at f_t

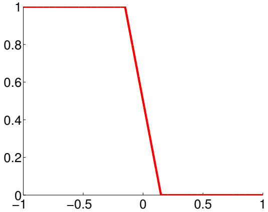
Solve the convex problem $f_{t+1} = \text{argmin}_{f \in \mathcal{C}_{Conv}} J_{Conv}(f, S)$

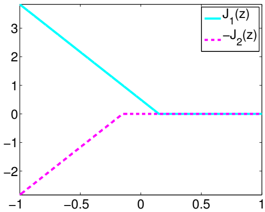
until termination

Example: DC Decomposition

$$\begin{aligned} J(f) &= J_1(f) - J_2(f) \\ J_{Conv} &= J_1(f) + \langle \beta_t, f \rangle + cte \end{aligned}$$

with $\beta_t \in \partial J_2(f_t)$





Algorithm 4 Synopsis to solve $\min_{f \in \mathcal{C} \subseteq \mathcal{H}} J(f, S)$

Set $t = 0$, initialize f

repeat

Find J_{Conv} and $\mathcal{C}_{Conv}$, convex relaxations of J and $\mathcal{C}$ at f_t

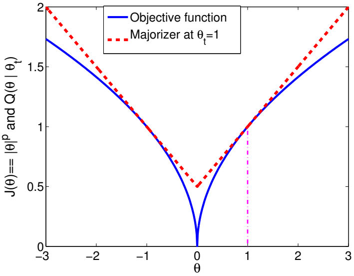
Solve the convex problem $f_{t+1} = \operatorname{argmin}_{f \in \mathcal{C}_{Conv}} J_{Conv}(f, S)$

until termination

Example: concave relaxation

$$
\begin{aligned}
J(\alpha) &= \|\alpha\|^p \\
J_{Conv} &= p|\alpha_t|^{p-1}|\alpha| + (1-p)|\alpha_t|^p
\end{aligned}
$$

with α_t the current solution



Brief summary

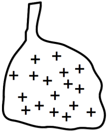
- Convex problems are "easy to solve"
- however most of learning issues are natively non-convex
- Promote Multi-stage convex relaxation to address them
- Does it work?

Outline

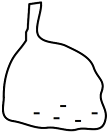
1. Introduction
   - General learning problem
   - Discussion of convexity and non-convexity of learning problem
   - Multi-stage convex relaxation
2. Case study
   - Learning under probability constraint
   - Multitask learning

Neyman Pearson classification

- Binary classification with samples $(\mathbf{x}, y) \in \mathcal{X} \times \{1, -1\}$
- Imbalanced data (medical diagnosis, surveillance system, ...)



vs



$S_+ = \{(\mathbf{x}_i, y_i = 1)\}_{i=1}^{n_+}$ $\quad$ $S_- = \{(\mathbf{x}_i, y_i = 1)\}_{i=1}^{n_-}$ $\quad$ with $n_+ \gg n_-$

Two types of errors

- False Alarm (FA) rate)
  $\mathbf{P}_{\mathrm{fa}}(f) = \mathbb{P}\big(f(\mathbf{x}) \geq 0 \,|\, y = -1\big)$
- Non-Detection (ND) Rate
  $\mathbf{P}_{\mathrm{nd}}(f) = \mathbb{P}\big(f(\mathbf{x}) \leq 0 \,|\, y = 1\big)$

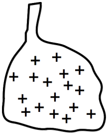
Control of FA rate

- Because of $n_+ \gg n_-$
- $\min_f \;\mathbf{P}_{\mathrm{nd}}(f)$ st
- Constraint: $\mathbf{P}_{\mathrm{fa}}(f) \leq \rho$

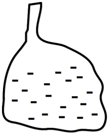
Learning under probability constraint: motivation (2)

q-value constraint

$$\min_f \; \mathrm{P}_{\mathrm{nd}}(f) \quad \text{s.t.} \quad \mathrm{P}_{\mathrm{fa}}(f) \leq q(1 - \mathrm{P}_{\mathrm{nd}}(f)) \qquad (q \ll 1 : \text{confidence level})$$



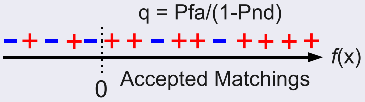
vs



Possible positives vs Reliable Negatives

Application

- Matching spectrum with peptides (pieces of proteins)
- Fake spectra are well known (randomly generated)
- True spectra are conjectured



- Assume $q = 0.01$ and $n_+ = n_-$
- Expecting $TP = 1000 \rightarrow FA \leq 10$

How to solve these problems?

Remark

1. Search for the saddle point of the lagrangian $\mathcal{L}(f, \lambda \geq 0)$
   - Neyman-Person: $\mathcal{L}(f, \lambda) = \mathbf{P}_{\mathrm{nd}}(f) + \lambda \left(\mathbf{P}_{\mathrm{fa}}(f) - \rho\right)$
   - q-value constraint: $\mathcal{L}(f, \lambda) = (1 + \lambda q)\, \mathbf{P}_{\mathrm{nd}}(f) + \lambda \mathbf{P}_{\mathrm{fa}}(f)$
2. Asymmetric Costs (AC) classification: $\min_f \; C_+ \, \mathbf{P}_{\mathrm{nd}}(f) + C_- \, \mathbf{P}_{\mathrm{fa}}(f)$
   - Costs specification not easy (while dealing with surrogate convex losses)

Problem involved by probability constraints

Find the appropriate costs asymmetry; Non-convexity

Solution

Guide the search by checking the probability constraint

Empirical risk Neyman-Pearson formulation

Estimation of probabilities of error

- Data set $S_+ = \{(\mathbf{x}_i, y_i = 1)\}_{i=1}^{n_+}$, $\quad S_- = \{(\mathbf{x}_i, y_i = -1)\}_{i=1}^{n_-}$
- Empirical Neyman-Pearson problem

$$\min_f \ \Omega(f) + C\hat{\mathrm{P}}_{\mathrm{nd}}(f) \quad \text{subject to} \quad \hat{\mathrm{P}}_{\mathrm{fa}}(f) \leq \rho$$

- Empirical probability errors ($0-1$ errors)

$$\hat{\mathrm{P}}_{\mathrm{nd}}(f) = \frac{1}{n_+}\sum_{i\in S_+} \mathbb{I}_{f(\mathbf{x}_i)\leq 0}, \quad \hat{\mathrm{P}}_{\mathrm{fa}}(f) = \frac{1}{n_-}\sum_{i\in S_-} \mathbb{I}_{f(\mathbf{x}_i)\geq 0}$$

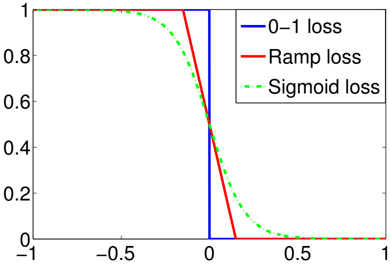
Using $0-1$ errors leads to NP hard problem

Our Proposal

- Non-convex approximation of the 0-1 errors

$$\hat{\mathbf{P}}_{\text{nd}}(f) = \frac{1}{n_+}\sum_{i\in S_+} \ell(y_i f(\mathbf{x}_i)) \;, \quad \hat{\mathbf{P}}_{\text{fa}}(f) = \frac{1}{n_-}\sum_{i\in S_-} \ell(y_i f(\mathbf{x}_i)) \;.$$

- Used approximation ℓ depends on the model family (kernel method, deep network) and optimization algorithm



Non-convex Neyman-Pearson classifier

Proposed Algorithms

- Kernel machine (SVM)
  - Ramp loss approximation
    $\ell(z) = \max\left\{0,\ \frac{1}{2}(1-z)\right\} - \max\left\{0,\ -\frac{1}{2}(1+z)\right\}$
  - Remark: non-convex and non-differentiable
  - Batch learning for non-linear SVM: tool = DC programming
  - Online learning for linear SVM (large scale datasets): tool = stochastic gradient
- Deep network
  - Sigmoid loss approximation $\ell(z) = \frac{1}{1+e^{z}}$
  - Online learning with stochastic gradient

$$\min_{f\in\mathcal{H}} \quad \Omega(f) + C\,\hat{\mathsf{P}}_{\mathrm{nd}}(f) \quad \text{s.t.} \quad \hat{\mathsf{P}}_{\mathrm{fa}}(f) \leq \rho$$

Step0 Augmented Lagrangian at iteration t

$$\mathcal{L}_A(f, \lambda \geq 0; \lambda_t) = \Omega(f) + C\,\hat{\mathsf{P}}_{\mathrm{nd}}(f) + \lambda\,(\hat{\mathsf{P}}_{\mathrm{fa}}(f) - \rho) + \frac{1}{\nu}(\lambda - \lambda_t)^2$$

Step1 f fixed $\rightarrow$ force λ to stay at the proximal of λ_t

$$\lambda \leftarrow \max\left\{0, \lambda_t + \nu(\hat{\mathsf{P}}_{\mathrm{fa}}(f) - \rho)\right\}$$

Step2 For λ fixed, solve the non-convex problem

$$f \leftarrow \mathrm{argmin}_{f\in\mathcal{H}}\Omega(f) + C\,\hat{\mathsf{P}}_{\mathrm{nd}}(f) + \lambda\,\hat{\mathsf{P}}_{\mathrm{fa}}(f)$$

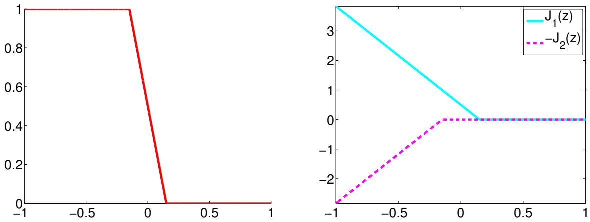
Batch learning of Neyman-Pearson SVM

Solving Step 2 at iteration t

- $\mathcal{L} = \frac{1}{2}\|f\|^2_{\mathcal{H}} + C_+ \sum_{i \in S_+} \ell(y_i f(\mathbf{x}_i)) + C_- \sum_{i \in S_-} \ell(y_i f(\mathbf{x}_i)) - \lambda\rho$

  with $C_+ = C/n_+$ and $C_- = \lambda/n_-$
- ℓ is the non-convex Ramp loss function
- Step 2 = Non-convex Asymmetric Costs SVM
- Apply Multi-stage Convex relaxation using a DC decomposition of ℓ

- $\ell(z) = \max\left\{0,\ \frac{1}{2}(1-z)\right\} - \max\left\{0,\ -\frac{1}{2}(1+z)\right\} = \ell_1(z) - \ell_2(z)$



- Decomposition of $\mathcal{L}(f,\lambda) = J_1(f) - J_2(f)$

$$\begin{aligned} J_1(f) &= \frac{1}{2}\|f\|^2_{\mathcal{H}} + \sum_i C_{y_i}\, \ell_1\big(y_i f(\mathbf{x}_i)\big)\,, \\ J_2(f) &= \sum_i C_{y_i}\, \ell_2(y_i f(\mathbf{x}_i)) \quad \text{where} \quad C_{y_i} \in \{C_+, C_-\} \end{aligned}$$

Batch learning of Neyman-Pearson SVM

Solving Step 2 at iteration t (cont'd)

- Convex majorization of $\mathcal{L}$

$$\mathcal{L}_{Conv} = \frac{1}{2}\|f\|^2_{\mathcal{H}} + \sum_i C_{y_i}\, \ell_1(y_i f(\mathbf{x}_i)) + \sum_i C_{y_i}\, \langle \nabla_f \ell_2(y_i f_t(\mathbf{x}_i)), f - f_t \rangle_{\mathcal{H}}$$

- We obtain classical SVM-like problem
- Solve the Non-convex Asymmetric Costs SVM with DC $\equiv$ solve iteratively SVM-type problem

Solving Neyman-Pearson SVM problem

1. For λ fixed, solve Non-convex SVM with $C_+ = C/n_+$, $C_- = \lambda/n_-$
2. Update λ according to Neyman-Pearson constraint satisfaction

Algorithm derivation

- Model $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$
- Reformulation of Neyman-Pearson problem

$$\min_f \frac{\lambda_c}{2}\|\mathbf{w}\|^2 + \frac{1}{n_+}\sum_{i \in S_+} \ell(y_i f(\mathbf{x}_i)) \quad \text{s.t.} \quad \frac{1}{n_-}\sum_{i \in S_-} \ell(y_i f(\mathbf{x}_i)) \leq \rho$$

- Lagrangian

$$\mathcal{L}(f, \lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\lambda_c}{2}\|\mathbf{w}\|^2 + a_i\, \ell(y_i f(\mathbf{x}_i)) - \lambda\rho\right)$$

with the coefficients $a_i = \begin{cases} n/n_+ & \forall i \in S_+ \\ \lambda n/n_- & \forall i \in S_- \end{cases}$

Algorithm 5 Stochastic algorithm

Initialize λ, $\mathbf{w}$, b.

repeat

Pick a random training example $(\mathbf{x}_t, y_t)$

Update $\mathbf{w}$ and b in the following ways

$$\begin{aligned} \mathbf{w} &\leftarrow (1 - \gamma_t \lambda_c)\mathbf{w} - \gamma_t a_t \nabla_{\mathbf{w}} \ell(y_t f(\mathbf{x}_t)) \\ b &\leftarrow b - \gamma_t a_t \nabla_b \ell(y_t f(\mathbf{x}_t)) \end{aligned}$$

If $y_t = -1$, set

$$\lambda \leftarrow \max\left(0, \lambda + \nu_t \left(\ell(y_t, f(\mathbf{x}_t)) - \rho\right)\right)$$

until convergence

- γ_t, ν_t: learning rates
- Neyman-Pearson constraint being related to negative samples, update of λ occurs if the current sample has a negative label

Remarks

Straightforward Extensions

- Online algorithm for deep network
- Batch and online algorithms for q-value constraint

$$\min_{f \in \mathcal{H}} \quad \Omega(f) + C\,\hat{\mathsf{P}}_{\mathsf{nd}}(f) \quad \text{subject to} \quad \hat{\mathsf{P}}_{\mathsf{fa}}(f) \leq q(1 - \hat{\mathsf{P}}_{\mathsf{nd}}(f))$$

  - Use the lagrangian

$$\begin{aligned} \mathcal{L}(f, \lambda) &= \Omega(f) + C\,\hat{\mathsf{P}}_{\mathsf{nd}}(f) + \lambda \left(\hat{\mathsf{P}}_{\mathsf{fa}}(f) - q(1 - \hat{\mathsf{P}}_{\mathsf{nd}}(f)) \right) \\ &= \Omega(f) + (C + \lambda q)\,\hat{\mathsf{P}}_{\mathsf{nd}}(f) + \lambda \hat{\mathsf{P}}_{\mathsf{fa}}(f) - \lambda q \end{aligned}$$

Performance evaluation: Neyman-Pearson

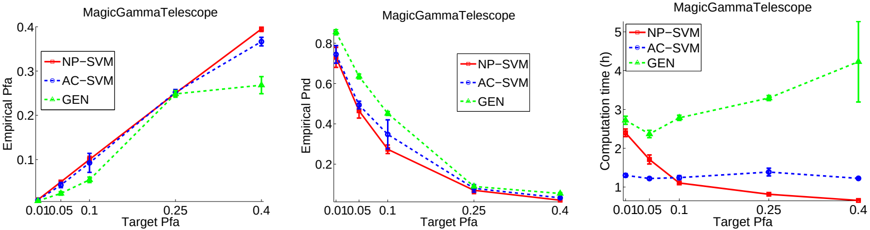
Compared methods

- Batch Neyman-Pearson (NP-SVM)
- Online Neyman-Pearson(ONP-SVM)
- Convex Asymmetric Costs SVM (AC-SVM)
  - Solve a convex SVM with costs (C_+, C_-). Check if the solution satisfies Neyman-Pearson constraint, otherwise look for another pair of costs.
- Generative approach (GEN)
  - Conditional distribution of each class $\equiv$ Gaussian distribution

Validation criterion

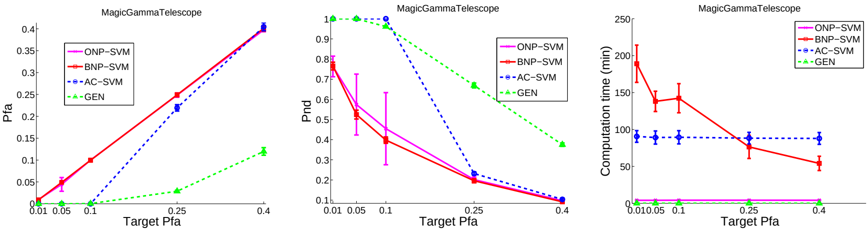
$$J_{val} = \hat{\mathsf{P}}_{\text{nd}} + \max(0, \hat{\mathsf{P}}_{\text{fa}} - \rho)/\rho$$

Results for nonlinear SVM model (medium scale $\approx$ 20,000 samples)



- Batch Neyman-Pearson (NP-SVM)
- Convex Asymmetric Costs SVM (AC-SVM)
- Generative approach (GEN)

Results for linear SVM model (medium scale ≈ 20,000 samples)



- Batch Neyman-Pearson (NP-SVM)
- Online Neyman-Pearson(ONP-SVM)
- Convex Asymmetric Costs SVM (AC-SVM)
- Generative approach (GEN)

Performance evaluation: Neyman-Pearson

Results for linear SVM (large scale $\approx$ 800,000 samples)

Table: Performances on test set (19700 positives and 3449 negatives) of RCV1-V2 for different values of ρ. Top row: left) $\rho = 0.1\%$, right) $\rho = 0.5\%$. Bottom Row: left) $\rho = 5\%$ and right) $\rho = 10\%$. Performances are percentages of errors.

	ONP-SVM	AC-SVM
$\hat{\mathbf{P}}_{fa}$	0.029	0
$\hat{\mathbf{P}}_{nd}$	**76.8**	93.26

	ONP-SVM	AC-SVM
$\hat{\mathbf{P}}_{fa}$	0.31	0.145
$\hat{\mathbf{P}}_{nd}$	60	**59.35**

	ONP-SVM	AC-SVM
$\hat{\mathbf{P}}_{fa}$	4.69	5.01
$\hat{\mathbf{P}}_{nd}$	11.84	**9.53**

	ONP-SVM	AC-SVM
$\hat{\mathbf{P}}_{fa}$	10	8.3
$\hat{\mathbf{P}}_{nd}$	**4.63**	7.9

Online NP-SVM (ONP-SVM) is in average 6 times faster than Convex Asymmetric Cost SVM (AC-SVM)

Performance evaluation: q-value

Setup

- Peptides-spectrum matching (PSM) verification
- Goal: identify consistently true positive matchings
- Models investigated : non-linear SVM (qSVMOpt), deep network (qNNOpt)

q	qRanker	qSVMOpt	qNNOpt
0.0025	4,449	4,947	**5,005**
0.01	5,462	5666	**5,707**
0.1	7,473	**7,954**	7,491

Table: Number of true positives correctly identified (over 34,852).

So far ...

- Learning with probability constraint
- The non-convex formulation leads to better results
- State-of-art results for PSM using q-value
- It is competitive in terms of computation time
- Online learning is strikingly fast ...

  ... but should be controlled carefully

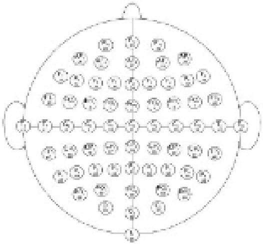
Outline

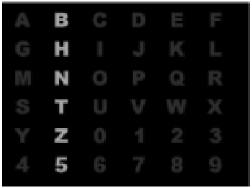
1. Introduction
   - General learning problem
   - Discussion of convexity and non-convexity of learning problem
   - Multi-stage convex relaxation
2. Case study
   - Learning under probability constraint
   - Multitask learning

Position of the problem

Brain computer interface Application

- P300 Speller System
- Characteristics: appearance of a deflection in the EEG signals 300*ms* (P300) after submitting a subject to a stimulus (visual stimulus)
- This deflection corresponds to an evoked potential (P300) to be detected
- M acquisition channels





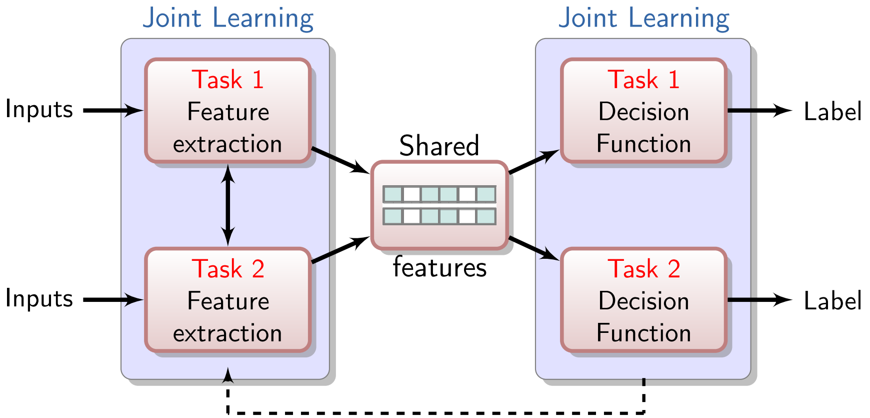
Problem setup

Issues

- Identify positive signals (with P300) from negative signals
- Select the useful channels or variables
- Handle the variability of the signals over different sessions and subjects

Workaround

- Define acquisition sessions as (nearly) similar tasks
- Learn jointly the tasks to improve performances
- Joint selection of discriminative features for the tasks



$$\min_{f_1, f_2 \in \mathcal{H}_1 \bigoplus \cdots \bigoplus \mathcal{H}_M} \sum_{t=1}^{2} \sum_{i=1}^{n_t} L\left(y_i^{(t)}, f_t\left(x_i^{(t)}\right)\right) + \lambda \, \Omega(f_1, f_2)$$

Group sparsity penalization

Joint sparsity penalization

- Two tasks with $f_1(x) = \langle \mathbf{w}_1, \mathbf{x} \rangle + b_1$ and $f_2(x) = \langle \mathbf{w}_2, \mathbf{x} \rangle + b_2$
- Penalization

$$\Omega(f_1, f_2) = \sum_j \mathbb{I}_{\mathbf{w}_{1,j} \neq 0 \,\wedge\, \mathbf{w}_{2,j} \neq 0} \qquad \text{NP hard !}$$

- Relaxation using mixed-norm $\|\cdot\|_{p,q}$

$$\begin{aligned} \Omega_{p,q}(f_1, f_2) &= \sum_j \sum_{t=1}^{2} \left(\left(|\mathbf{w}_{t,j}|^q \right)^{1/q} \right)^p \\ &= \sum_j \left(\|\mathbf{W}(:,j)\|_q \right)^p \quad \text{with} \quad \mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2]^\top \end{aligned}$$

  - $\|\mathbf{W}(:,j)\|_q$ encodes relation between tasks (if it is small, variable j is irrelevant for both tasks)
  - ℓ_p-norm encodes joint sparsity level
  - $0 < p < 1$ enforces sparsity but problem is non-convex

Joint sparsity penalization: non-linear case

- Three kernel spaces $\mathcal{H}_m$, with kernels k_m
- Decision function $f_t(x)$

$$f_t(x) = f_{t,1}(x) + f_{t,2}(x) + f_{t,3}(x) + b_t \quad \text{with} \quad f_{t,m} \in \mathcal{H}_m$$

- Penalization

$$\Omega_{p,q}(f_1, f_2) = \sum_{m=1}^{3} \left(\sum_{t=1}^{2} \|f_{t,m}\|_{\mathcal{H}_m}^q \right)^{p/q} = \color{red}{\sum_{m=1}^{3} (\color{blue}{\|f_{\cdot,m}\|}\color{red}{)^p}}$$

- $\color{blue}{\|f_{\cdot,m}\|} = \left(\sum_{t=1}^{2} \|f_{t,m}\|_{\mathcal{H}_m}^q \right)^{1/q}$ measures the importance of kernel k_m across the tasks.

Multiple kernel Multi-task Learning

Optimization problem: general case

$$\min_{f_1,\cdots,f_T \in \mathcal{H}_1 \bigoplus \cdots \bigoplus \mathcal{H}_M} \sum_{t=1}^{T} \sum_{i=1}^{n_t} L\left(y_i^{(t)}, f_t\left(x_i^{(t)}\right)\right) + \lambda\, \Omega_{p,q}(f_1,\cdots,f_T)$$

with $\Omega_{p,q}(f_1,\cdots,f_T) = \sum_{m=1}^{M}\left(\sum_{t=1}^{T} \|f_{t,m}\|_{\mathcal{H}_m}^{q}\right)^{p/q}$

Elements of solution

- Convex case ($p = 1$): equivalent penalization with $s = (2-q)/q$

$$\Omega_{p,q}(f_1,\cdots,f_T)^2 = \min_{d_{t,m}\geq 0} \sum_{m=1}^{M} \frac{\|f_{t,m}\|^2}{d_{t,m}} \quad \text{s.t} \quad \sum_m \left(\sum_t d_{t,m}^{1/s}\right)^s \leq 1$$

- Efficient solvers exist (multiple kernel learning)

Multiple kernel Multi-task Learning

Optimization problem: general case

$$\min_{f_1,\cdots,f_T \in \mathcal{H}_1 \bigoplus \cdots \bigoplus \mathcal{H}_M} \sum_{t=1}^{T} \sum_{i=1}^{n_t} L\left(y_i^{(t)}, f_t\left(x_i^{(t)}\right)\right) + \lambda \, \Omega_{p,q}(f_1, \cdots, f_T)$$

with $\Omega_{p,q}(f_1, \cdots, f_T) = \sum_{m=1}^{M} \left(\sum_{t=1}^{T} \|f_{t,m}\|_{\mathcal{H}_m}^{q} \right)^{p/q}$

Elements of solution

- Non-Convex case ($0 < p < 1$) for enhanced sparsity
- Use Multi-Stage Convex Refinements
- Notice that $\Omega_{p,q}(f_1, \cdots, f_T) = \sum_{m=1}^{M} g(\|f_{\cdot,m}\|)$ with $g(u) = |u|^p$
- Convex relaxation at iteration t: $g(u) \leq p|u_t|^{p-1}|u| + (1-p)|u_t|^p$

Application on BCI data

- 9 subjects $\rightarrow$ 9 tasks
- 256 features, training sets of size $n = 300$

	$MTL_{1,2}$	$MTL_{p,2}$	$MTL_{1,q}$	SepSVM	Sepℓ_1SVM
AUC	**76.5 ± 0.6**	76.1 ± 0.5	76.5 ± 0.6	75.6 ± 0.8	73.4 ± 1.3
# Var	191 ± 26	134 ± 33	201 ± 23	256	**118 ± 30**

SepSVM: tasks are trained separately using classical SVM

Sepℓ_1SVM: tasks are trained separately using penalised ℓ_1-norm SVM

Application on Multiclass problem

- Proteins classification
- Tasks: pairwise binary classification in 1-vs-all fashion
- Two datasets
  - Dataset 1 : PSORT+ (4 classes, 541 samples)
  - Dataset 2 : PSORT- (5 classes, 1444 samples)
- Initial number of kernels: 69

Data	$\text{MTL}_{1,2}$	$\text{MTL}_{p,2}$	$\text{MTL}_{1,q}$	MCMKL
PSORT +	93.87 ± 2.82	93.62 ± 3.04	93.88 ± 2.73	93.8
# Kernels	15.4 ± 1.17	7.4 ± 1.42	15.9 ± 1.05	18
PSORT -	95.92 ± 1.35	95.90 ± 1.12	96.02 ± 1.33	96.1
# Kernels	12.9 ± 0.31	7.5 ± 0.85	12.8 ± 0.42	14

... So good

- Group sparsity based on kernels and using mixed-norm
- Sharing information across tasks helps
- Non-convex solutions: better or similar performances with reduced complexity
- Why does it work ?
  - Convex approaches provide sub-optimal solutions when dealing with sparsity
  - Non-convex penalizations can alleviate these drawbacks
  - They trade convexity for enhanced sparsity
  - Some theoretical guarantees are emerging (at least for regression) [Zhan 2010]

References

[Coll 2006] R. Collobert, F. Sinz, J. Weston, and L. Bottou. "Trading convexity for scalability". In: Proceedings of the 23rd international conference on Machine learning (ICML 2006), pp. 201-208, Pennsylvania, USA, 2006.

[Wu 2010] T. T. Wu and K. Lange. "The MM Alternative to EM". Statistical Science, Vol. 25, No. 4, pp. 492-505, 2010.

[Zhan 2010] T. Zhang. "Analysis of Multi-stage Convex Relaxation for Sparse Regularization". Journal of Machine Learning Research, Vol. 11, pp. 1081-1107, March 2010.

[Tao, 1998] P. D. Tao and L. T. H. An. "DC optimization algorithms for solving the trust region subproblem". SIAM Journal of Optimization, Vol. 8, No. 2, pp. 476-505, 1998.