Batch and online approaches for constrained classification Neyman-Pearson and q-value classification

Gilles Gasso

gilles.gasso@insa-rouen.fr

LITIS *EA 4108* - INSA de Rouen

Séminaire GDR ISIS

07 Avril 2011

Roadmap

Context

- Binary classification with samples $(\mathbf{x}, y) \in \mathcal{X} \times \{1, -1\}$
- Let $f(\mathbf{x})$ the decision function
- Contingency table

	$y = 1$	$y = -1$
$\text{sign}(f(\mathbf{x})) = 1$	True Positives (TP)	False Alarm (FA)
$\text{sign}(f(\mathbf{x})) = -1$	Non Detection (ND)	True Negatives (TN)

- Two types of errors
  - Type I: probability of false alarm (FA rate)
    $\mathbf{P}_{\text{fa}}(f) = \mathbb{P}\big(f(\mathbf{x}) \geq 0 \,|\, y = -1\big)$
  - Type II: Probability of non detection (ND rate)
    $\mathbf{P}_{\text{nd}}(f) = \mathbb{P}\big(f(\mathbf{x}) \leq 0 \,|\, y = 1\big)$

Motivation

- Need to control one kind of error
- Two ways

Motivation

- Need to control one kind of error
- Two ways

1 - Contingency table based objective functions

- Asymmetric costs

$$\min_f \; C_+ \mathbf{P}_{\text{nd}}(f) + C_- \mathbf{P}_{\text{fa}}(f)$$

  - In practice, costs specification is complicated

- Precision or Recall (with k predicted positives) [Joachims, 2005]

$$Prec_k = \frac{TP}{TP + FA}, \quad Rec_k = \frac{TP}{TP + ND}$$

- F-measure

$$F = \frac{2Prec \times Rec}{Prec + Rec}$$

  - Difficult optimization problem (polynomial time algorithm by [Joachims, 2005])

Motivation

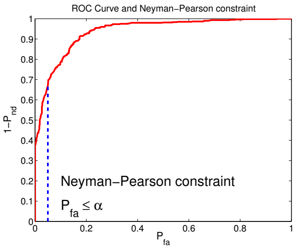
- Need to control one kind of error
- Two ways

2 - Probability constraints

- Neyman-Pearson classifier

$$\min_f \; P_{nd}(f) \quad \text{s.t.} \quad P_{fa}(f) \leq \alpha \qquad (\alpha : \text{maximal false alarm rate})$$

- Typical applications: surveillance, drug screening, medical diagnosis, signal detection against background, imbalanced classification



Motivation

- Need to control one kind of error
- Two ways

2 - Probability constraints

- *q*-value

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

Application: tandem mass spectrometry of proteins mixtures

- Peptides (pieces of proteins) spectrum matching
- Consider True Database $\mathcal{T}$ and Decoys Database
- Matching spectrum with fake peptides $\rightarrow$ true negative samples
- Matching spectrum with peptides in $\mathcal{T}$ $\rightarrow$ possibly positives
- Assign confidently the positive labels, the negatives being sure

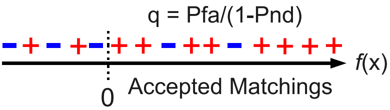
Motivation

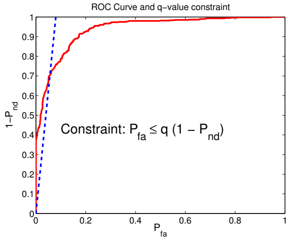
- Need to control one kind of error
- Two ways

2 - Probability constraints

- *q*-value

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





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

Adopted approach

Equivalence between formulations

1. Probability constraints

   Search for the saddle point of the lagrangian $\mathcal{L}(f, \lambda \geq 0)$

   - Neyman-Person: $\mathcal{L}(f, \lambda) = \mathbf{P}_{\mathrm{nd}}(f) + \lambda \left(\mathbf{P}_{\mathrm{fa}}(f) - \alpha\right)$
   - 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

Solution

Guide the search by checking the probability constraint

The framework

- Data set $\mathcal{D} = \mathcal{D}_+ \cup \mathcal{D}_-$

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

- Neyman-Pearson problem

$$\min_f \ \hat{\mathsf{P}}_{\mathrm{nd}}(f) \quad \text{subject to} \quad \hat{\mathsf{P}}_{\mathrm{fa}}(f) \leq \alpha$$

Empirical probability errors ($0-1$ errors)

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

Existing approaches

Generative approach [Kim et al., 2006]

- Linear classifier $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.
- Assumptions: class-conditional distributions are Gaussian with means $\boldsymbol{\mu}_\pm$ and covariances $\boldsymbol{\Sigma}_\pm$
- Solve Neyman-Pearson problem with

$$\hat{\mathrm{P}}_{\mathrm{nd}} = \Phi\left(-\frac{b+\mathbf{w}^\top\hat{\boldsymbol{\mu}}_+}{\sqrt{\mathbf{w}^\top\hat{\boldsymbol{\Sigma}}_+\mathbf{w}}}\right), \quad \hat{\mathrm{P}}_{\mathrm{fa}} = \Phi\left(\frac{b+\mathbf{w}^\top\hat{\boldsymbol{\mu}}_-}{\sqrt{\mathbf{w}^\top\hat{\boldsymbol{\Sigma}}_-\mathbf{w}}}\right)$$

  Φ: cumulative distribution function of standard normal distribution
  $\hat{\boldsymbol{\mu}}_\pm$ and $\hat{\boldsymbol{\Sigma}}_\pm$ are empirical estimations.
- Straightforward Kernelization
- Drawbacks: lack of sparsity when kernelized; gaussian assumption too restrictive
- Relax Gaussian assumption: use instead of Φ, Chebyshev bound $\Psi(u) = [u]_+^2/(1+[u]_+^2)$, $[u]_+ = \max(0, u)$

Existing approaches

Discriminative approach: Convex Asymmetric Cost SVM [Bach et al., 2006, Davenport et al., 2010]

- $\min_{f\in\mathcal{H}} \quad \Omega(f) + C_+ \hat{\mathrm{P}}_{\mathrm{nd}}(f) + C_- \hat{\mathrm{P}}_{\mathrm{fa}}(f)$
- $\Omega(f) = \frac{1}{2}\|f\|^2_{\mathcal{H}}$: regularizer
- Convex surrogate of the 0-1 classification errors using hinge loss

$$\hat{\mathrm{P}}_{\mathrm{nd}}(f) = \frac{1}{n_+}\sum_{i\in\mathcal{D}_+} H_\ell(y_i f(\mathbf{x}_i)), \quad \hat{\mathrm{P}}_{\mathrm{fa}}(f) = \frac{1}{n_-}\sum_{i\in\mathcal{D}_-} H_\ell(y_i f(\mathbf{x}_i))$$

with $\quad H_\ell(y_i f(\mathbf{x}_i)) = \max(0, 1 - y_i f(\mathbf{x}_i))$

- Find the appropriate costs C_+ and C_- to satisfy Neyman-Pearson constraint $\Longrightarrow$ search in costs space (C_+, C_-)
- Because of convex surrogate, signification of the costs is lost

Our proposals

Proposed solutions

- Rely on discriminative approach
- Deal directly with the non-convex probability constraint for Neyman-Pearson
- Extension to q-value constraint

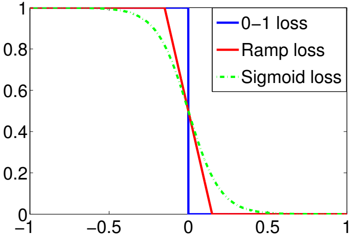
Our proposals

Non-convex Neyman-Pearson classifier

- $\min_{f \in \mathcal{H}} \quad \Omega(f) + C\,\hat{\mathsf{P}}_{\mathrm{nd}}(f) \quad$ subject to $\quad \hat{\mathsf{P}}_{\mathrm{fa}}(f) \leq \alpha$
- Non-convex approximation of the 0-1 errors

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

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



Our proposals

Algorithms for Non-convex Neyman-Pearson classification

- 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

Primitive of the optimization algorithms

Unconstrained augmented lagrangian (Uzawa algorithm)

General Algorithm

Principle

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

- Augmented Lagrangian at iteration t

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

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

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

- λ fixed $\rightarrow \min_{f\in\mathcal{H}} \mathcal{L}_A(f,\lambda) \equiv \min_{f\in\mathcal{H}} \mathcal{L}(f,\lambda)$

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

Algorithm 1 Uzawa Algorithm

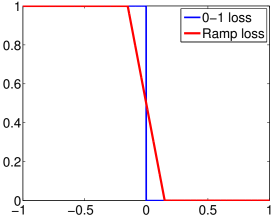
Set initial value for $\lambda \geq 0$. Pick small gain $\nu > 0$.
repeat
STEP 1 : $f \leftarrow \mathrm{argmin}_{f \in \mathcal{H}}\, \mathcal{L}(f, \lambda)$
STEP 2 : $\lambda \leftarrow \max\left\{0, \lambda + \nu\left(\hat{\mathsf{P}}_{\mathrm{fa}}(f) - \alpha\right)\right\}$
until convergence

Trick: Use a multiplicative update to keep $\lambda \geq 0$
$\lambda \leftarrow \lambda(1 + \nu\left(\hat{\mathsf{P}}_{\mathrm{fa}}(f) - \alpha\right))$

Batch learning of Neyman-Pearson SVM

Algorithm derivation

- $\mathcal{L} = \frac{1}{2}\|f\|^2_{\mathcal{H}} + C_+\sum_{i\in\mathcal{D}_+} \ell(y_i f(\mathbf{x}_i)) + C_-\sum_{i\in\mathcal{D}_-} \ell(y_i f(\mathbf{x}_i)) - \lambda\alpha$
  with $C_+ = C/n_+$, $C_- = \lambda/n_-$
- Ramp loss function $\ell(z) = \max\left\{0, \frac{1}{2}(1-z)\right\} - \max\left\{0, -\frac{1}{2}(1+z)\right\}$
- Step 1 of Uzawa Algorithm solves Non-convex Asymmetric Costs SVM



Issue: non-convexity and non-differentiability of the ramp loss

However, problem amenable to DC programming

Difference of Convex (DC) programming [Tao and An, 1998]

- Non-convex (non-differentiable) problem

$$\min_{\theta} J_1(\theta) - J_2(\theta)$$

  J_1 and J_2 are convex functions (1).

- Solve iteratively the linearized convex problem

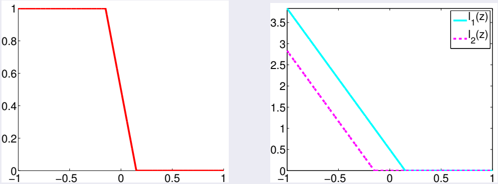
$$\theta_{t+1} = \operatorname*{argmin}_{\theta} J_1(\theta) - \langle \nabla_\theta J_2(\theta^t), \theta - \theta^t \rangle \quad (2)$$

- The objective function $J_1(\theta) - J_2(\theta)$ decreases at each iteration as

$$\begin{aligned} J_1(\theta_{t+1}) + \langle \nabla_\theta J_2(\theta_t), \theta_{t+1} \rangle &\leq J_1(\theta_t) + \langle \nabla_\theta J_2(\theta_t), \theta_t \rangle & (2) \\ -J_2(\theta_{t+1}) &\leq -J_2(\theta_t) + \langle \nabla_\theta J_2(\theta_t), \theta_{t+1} - \theta_t \rangle & (1) \end{aligned}$$

Applying DC to Non-convex Asymmetric Costs SVM

- $\mathcal{L} = \frac{1}{2}\|f\|^2_{\mathcal{H}} + C_+ \sum_{i\in\mathcal{D}_+} \ell(y_i f(\mathbf{x}_i)) + C_- \sum_{i\in\mathcal{D}_-} \ell(y_i f(\mathbf{x}_i))$
- $\ell(z) = \max\left\{0,\ \frac{1}{2}(1-z)\right\} - \max\left\{0,\ -\frac{1}{2}(1+z)\right\} = \ell_1(z) - \ell_2(z)$



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

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

Applying DC to Non-convex Asymmetric Costs SVM (cont'd)

- Convex linearized $\mathcal{L}$

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

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

Applying DC to Non-convex Asymmetric Costs SVM (cont'd)

- Convex linearized $\mathcal{L}$

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

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

Solving Non-Convex Neyman-Pearson 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

Limitations

Computation bulk for non-linear SVM

Remedies to computation bulk

Solution 1: speed-up trick

- Update the Lagrange parameter λ after each iteration of DC
- Avoid solving many times Nonconvex SVM problem

Algorithm 2 Annealed Uzawa algorithm

---

repeat

- Set $C_+ = C/n_+$ and $C_- = \lambda/n_-$.
- Solve for one iteration of DC $\rightarrow f(\mathbf{x})$
- Update $\lambda \leftarrow \lambda(1 + \nu(\hat{\mathbf{P}}_{\text{fa}}(f) - \alpha))$

until convergence.

Solution 2: online learning

- However, non-linear kernel SVM case is challenging
- Focus on linear SVM and deep architecture

Algorithm derivation

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

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

- Lagrangian

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

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

Online learning of Neyman-Pearson SVM

Algorithm 3 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

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

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

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

  - Use the lagrangian

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

Performances evaluation

Dataset	#features	n_+	n_-
Spambase	57	2788	1813
GammaTelescope	10	12332	6688
Covertype	54	211840	20510
RCV1-V2	47152	684494	119920

Compared methods

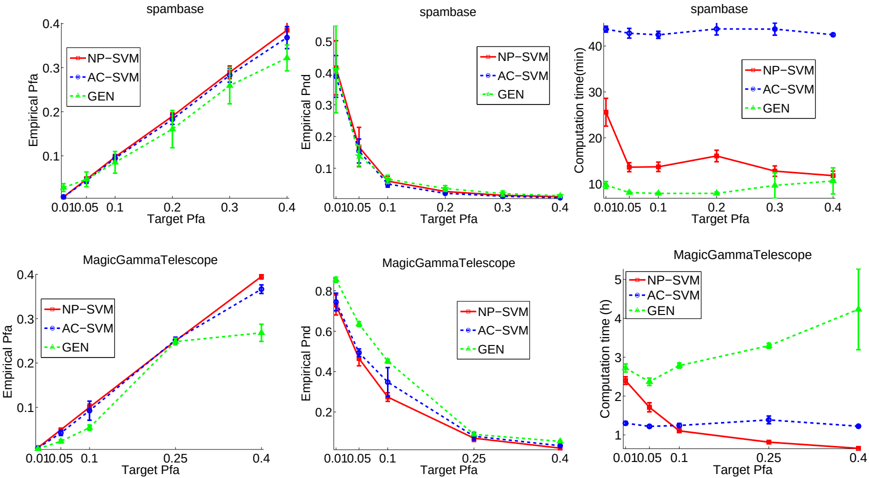
- Batch Neyman-Pearson (NP-SVM) : requires specification of (C_+, σ)
- Online Neyman-Pearson(ONP-SVM) : requires specification of (λ_c, γ)
- Convex Asymmetric Costs SVM (AC-SVM) : triplet (C_+, C_-, σ)
- Generative approach (GEN) : only σ is needed

Validation criterion

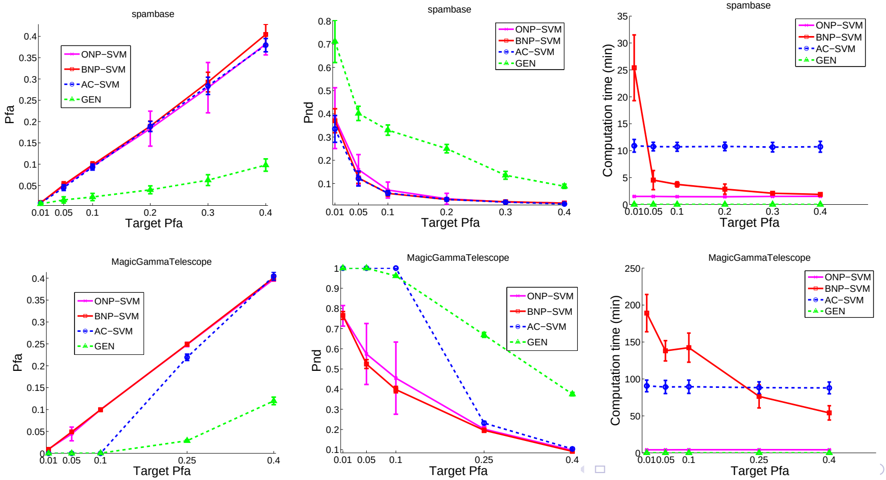
$$J_{val} = \hat{\mathrm{P}}_{\mathrm{nd}} + \max(0, \hat{\mathrm{P}}_{\mathrm{fa}} - \alpha)/\alpha$$

Performances evaluation of proposed algorithms

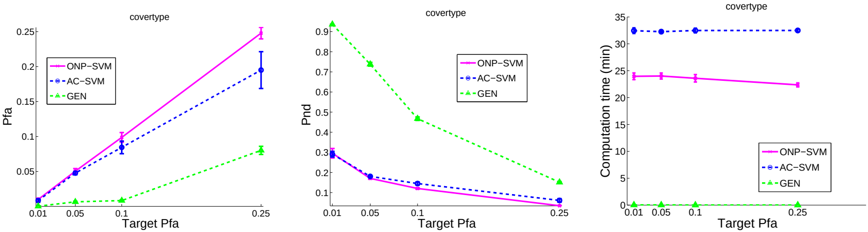
Results for nonlinear SVM model



Results for linear SVM model



Results for linear SVM model



Results for linear SVM model

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

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

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

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

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

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

Conclusion

- Batch and online approaches to tackle NP classification problem
- Framework can be extended to address q-value optimization problem
- Perspective: derive an efficient extension for online kernel learning

q-value optimization results

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

Table: Number of true positives correctly identified (over 34852).

q	qRanker	qSVMOpt	qNNOpt
0.0025	4449	4947	**5005**
0.01	5462	5666	**5707**
0.1	7473	**7954**	7491

That's all folks

Questions ?

F. R. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *The Journal of Machine Learning Research*, 7:1741, 2006.

M. Davenport, R. Baraniuk, and C. Scott. Tuning support vector machines for minimax and neyman-pearson classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2010. ISSN 0162-8828.

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proc. of International Conference on Machine Learning*, 2005.

S.J. Kim, A. Magnani, S. Samar, S. Boyd, and J. Lim. Pareto optimal linear classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 473–480, New York, NY, USA, 2006. ACM.

P. D. Tao and L. T. Hoai An. Dc optimization algorithms for solving the trust region subproblem. *SIAM Journal of Optimization*, 8(2):476–505, 1998.