Learning with infinitely many features

A. Rakotomamonjy
joint work with R. Flamary and F. Yger

LITIS EA 4108
Université de ROUEN

November 2011

Infinitely many features?

When does it occurs?

- Feature extraction with continuous parameters
- Wavelet or Gabor based features of the form

$$\langle \mathbf{x}, \psi_{j,k,\theta} \rangle \qquad \langle \mathbf{x}, \psi_{u,v,\sigma,\lambda} \rangle$$

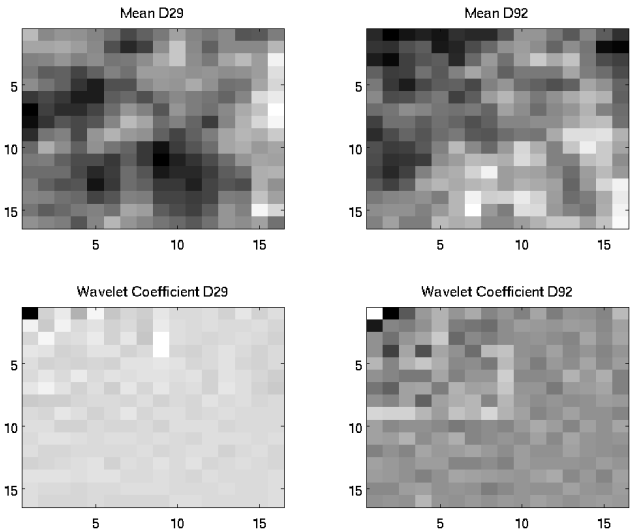
  - Brain Computer Interfaces problem or texture ecognition
- Explicit feature maps with continuous parameters
  - kernel with feature scaling : $k(\mathbf{x}, \mathbf{x}') = e^{-\sum_j \frac{(x_j - x'_j)^2}{2\sigma_j^2}}$

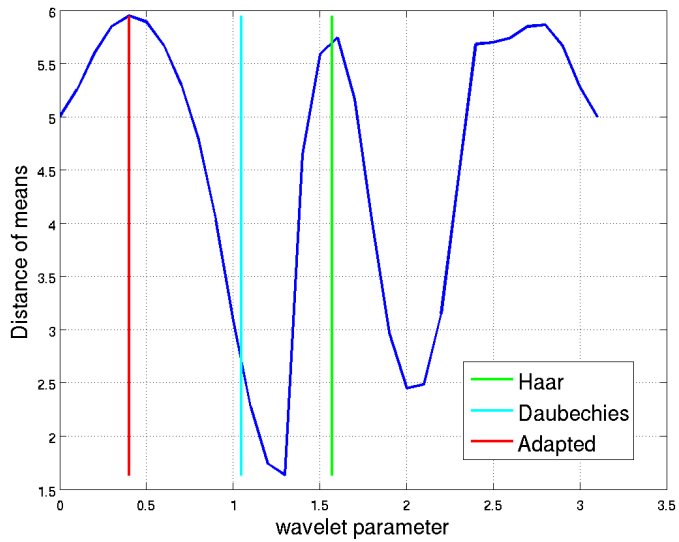
Approach

- Consider a empirical risk minimization framework that selects few features among infinitely many
- sparsity inducing regularizers

Simple illustration using wavelets

- Look for the best discriminative wavelet basis for classifying texture patches
- "discriminative" = maximize distance of means in wavelet decomposition space
- wavelets are parametrized through their QMF by vector of angles
- discretization leads to large amount of features





Related work

- Extension the Lasso to infinite dimension feature space (Rosset, COLT 2004)

$$\min_{p \in \mathcal{P}, p \geq 0} \sum_{i=1}^{n} L\left(y_i, \int \tilde{\Phi}_\theta(\mathbf{x}_i) p(\theta) d\theta\right) st \int p(\theta) d\theta \leq \lambda$$

- ℓ_1 like penalty
- Equivalent to the Lasso if the parameter space is finite
- the solution is still sparse
- LARS-like path-following algorithm for solving the problem
  - works for specific features
  - unstable

Our framework

- Formulation
  - Look for the finite subset of feature that yields to the lowest minimum empirical risk
  - The number of finite subset is still infinite but the ERM applies to a finite number of features.

- Notations
  - $\mathcal{F}$ the set of all possible finite subset of features
  - φ an element of $\mathcal{F}$ composed of d features $\{\Phi_{\theta_j}\}_{i=1}^{d}$, with θ being the feature parameter
  - For an optimal $\varphi^\star$ with optimal parameters $\{\theta_j^\star\}$, the decision function writes:

$$f(\mathbf{x}) = \sum_{j=1} \mathbf{w}_j \Phi_{\theta_j^\star}(\mathbf{x}) = \mathbf{w}^T \Phi_\theta$$

Optimization problem

- Learning examples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$
- Formulation

$$\min_{\varphi \in \mathcal{F}} \min_{\mathbf{w}} \sum_{i=1}^{n} L(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i)) + \lambda \Omega(\mathbf{w})$$

  - $L(\cdot, \cdot)$ convex and differentiable loss function
  - $\Omega(\cdot)$ norm based sparsity-inducing regularizers
  - λ : trade-off hyperparameter
- two-step optimization, bi-level optimization
  - ERM with finite feature set φ
  - optimization over the feature set

Optimality conditions for $\Omega(\mathbf{w}) = \|\mathbf{w}\|_1$

- inner problem

$$\begin{aligned} \sum_i \Phi_{\theta_j}(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i)) + \lambda sign(w_j) &= 0 \quad \text{if } w_j \neq 0 \\ \left|\sum_i \Phi_{\theta_j}(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i))\right| &\leq \lambda \quad \text{if } w_j = 0 \text{ and } \Phi_\theta \in \varphi \end{aligned}$$

- full problem

$$\begin{aligned} \sum_i \Phi_{\theta_j}(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_{\theta_j}(\mathbf{x}_i)) + \lambda sign(w_j) &= 0 \quad \text{if } w_j \neq 0 \\ \left|\sum_i \Phi_{\theta_j}(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i))\right| &\leq \lambda \quad \text{if } w_j = 0 \text{ and } \Phi_{\theta_j} \in \varphi \\ \left|\sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i))\right| &\leq \lambda \quad \text{if } \Phi \notin \varphi \end{aligned}$$

- Intuition : a feature violating constraint in red also violates the optimality condition of the inner problem with augmented feature set $\varphi \cup \Phi$

Remarks

- Violating constraint feature
  - suppose $\mathbf{w}^\star$ solution of the inner problem with the feature set φ.
  - any Φ violating

$$\left| \sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^{\star T} \Phi_\theta(\mathbf{x}_i)) \right| \leq \lambda$$

    would lead to a decrease of the objective function if added to φ.
- Active set Algorithm
  - train with a finite set of feature φ
  - select one violating constraint ϕ and update φ : $\varphi \leftarrow \varphi \cup \phi$
  - re-train

Algorithm properties

- For checking the optimality of the full problem, we have to be able to solve

$$\max \left| \sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i)) \right|$$

- ϵ-approximate solution : if the inner problem can be solved exactly and we can compute the above equation then the algorithm provides an ϵ-approximate solution in finite time.

Violating constraint features

- A key point of the algorithm is the resolution of

$$\max_{\Phi} \left| \sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i)) \right|$$

- Depending on $L(\cdot, \cdot)$ and the structure of Φ_θ, the problem can be very difficult.
- randomization, brute force, or clever search if applicable
  - sample some values of θ
  - select the feature that maximizes $\left| \sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_\theta(\mathbf{x}_i)) \right|$
  - sub-optimal but efficient

Algorithmic implementation

- Randomization for feature searching
- minimization of empirical risk + sparse regularizer for the inner problem
  - fast proximal algorithm or alternate direction methods of multipliers
- Instantiation with square hinge loss of the ADMM approach

$$\min_{\mathbf{w}} \max(0, 1 - \mathbf{y}\Phi\mathbf{w})^T \max(0, 1 - \mathbf{y}\Phi\mathbf{w}) + \lambda\Omega(\mathbf{w})$$

- variable splitting

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v},\mathbf{w}} \quad & \max(0, \mathbf{u})^T \max(0, \mathbf{u}) + \lambda\Omega(\mathbf{v}) \\ & \mathbf{u} = 1 - \mathbf{y}\Phi\mathbf{w} \\ & \mathbf{v} = \mathbf{w} \end{aligned}$$

decouples the influence of the loss and the regularizer in the optimization problem.

Algorithmic implementation : ADMM (2)

- Lagrangian

$$\begin{aligned}\mathcal{L} &= \max(0,\mathbf{u})^T \max(0,\mathbf{u}) + \lambda\Omega(\mathbf{v}) + \alpha^T(\mathbf{u} - 1 + \mathbf{y}\Phi\mathbf{w}) \\ &\quad +\beta^T(\mathbf{v}-\mathbf{w}) + \tfrac{\nu}{2}\|\mathbf{u} - 1 + \mathbf{y}\Phi\mathbf{w}\|^2 + \tfrac{\nu'}{2}\|\mathbf{v}-\mathbf{w}\|^2\end{aligned}$$

- Iteration
  - minimization of the augmented Lagrangian wrt to each single primal variable
  - update of the dual variable α, β
- Steps :
  - linear system for $\mathbf{w}$
  - proximal operator update for $\mathbf{u}$ related to the loss function
  - proximal operator update for $\mathbf{v}$ related to the regularizer
- Nice points
  - simple and generic
  - convergence for inexact proximal operators
  - efficient

Extensions to other paradigms

- non-differentiable norm-based regularization term $\Omega(\mathbf{w})$. The violating constraint condition becomes

$$\Omega^{\star}\left(\sum_i \Phi(\mathbf{x}_i) L'(y_i, \mathbf{w}^T \Phi_{\theta_j}(\mathbf{x}_i))\right) \leq \lambda$$

  with $\Omega^{\star}(\mathbf{w})$ being the dual norm of $\Omega(\mathbf{w})$.

- Multi-task framework with shared and specific norm based regularizers for feature selection *e.g* $\ell_1 - \ell_q$ mixed-norm whose dual is $\ell_\infty - \ell_{q'}$

$$\|\mathbf{W}\|_{1,q} = \sum_{i=1}^{d} \|\mathbf{W}_{\cdot,t}\|_q$$

Application to kernel and multiple kernel approximation

- simple and efficient to kernel method : use explicit features if any.
- Gaussian kernel $k(\mathbf{x}, \mathbf{x}')$

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{m} [\cos(\mathbf{v}_j^T \mathbf{x}) \cos(\mathbf{v}_j^T \mathbf{x}') + \sin(\mathbf{v}_j^T \mathbf{x}) \sin(\mathbf{v}_j^T \mathbf{x}')]$$

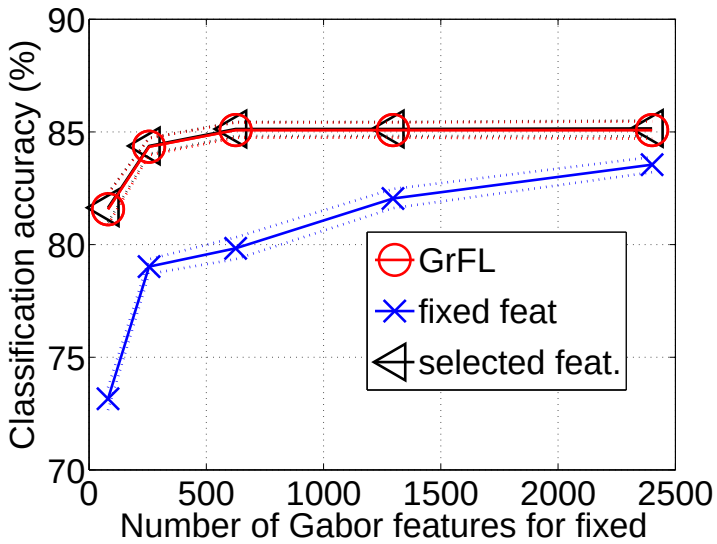
  where $\{\mathbf{v}_j\}$ are random vectors samples according to the FT of the Gaussian kernel
- Application in our framework :
  - sample several values of the Gaussian kernel bandwidth
  - for each value, draw direction vectors $\{\mathbf{v}_j\}$
  - for all bandwidth and direction vectors, compute the constraint violation
  - select the pair of features violating the most their constraints.

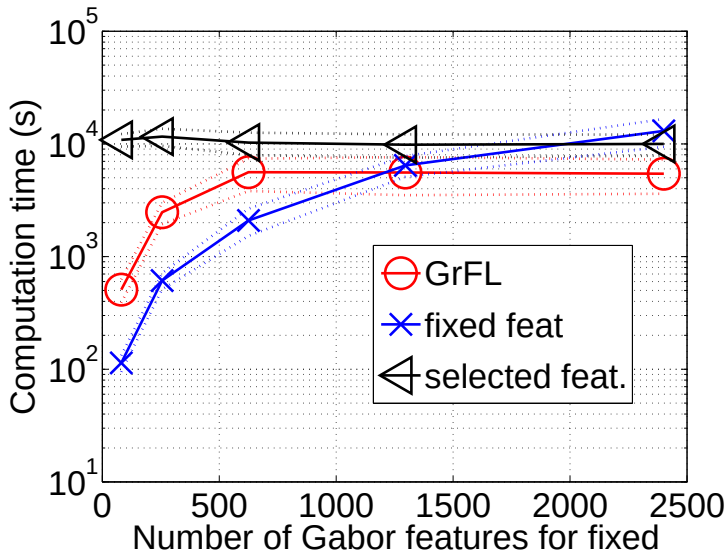
Experimental results

- Gabor features for multiclass texture recognition problems
  - comparison with sampled parameters of feature extraction
- Large scale approximated kernel machines
  - comparison with incomplete choleski decomposition

Gabor feature for texture recognition

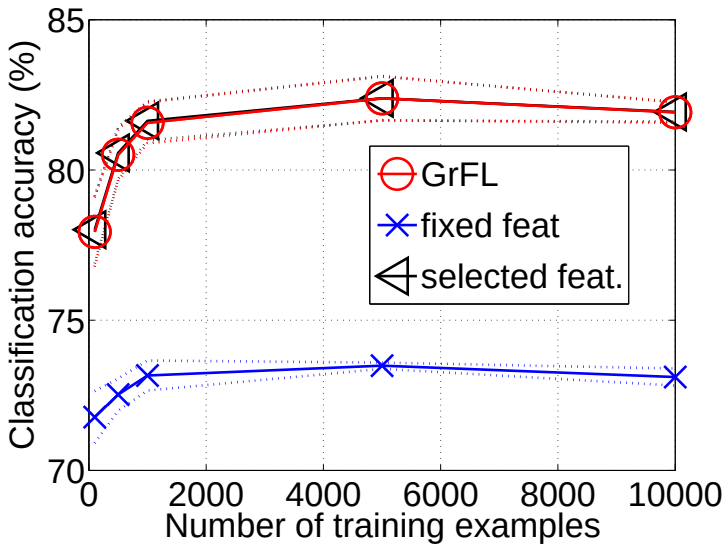
- 3 classes, 16×16 patches from the texture image
- increasing number of features and 1000 examples per class
- approaches
  - GrFL : our method
  - fixed feat : pre-defined features through discretization
  - selected feat: Lasso with 3000 of the features visited by GrFL

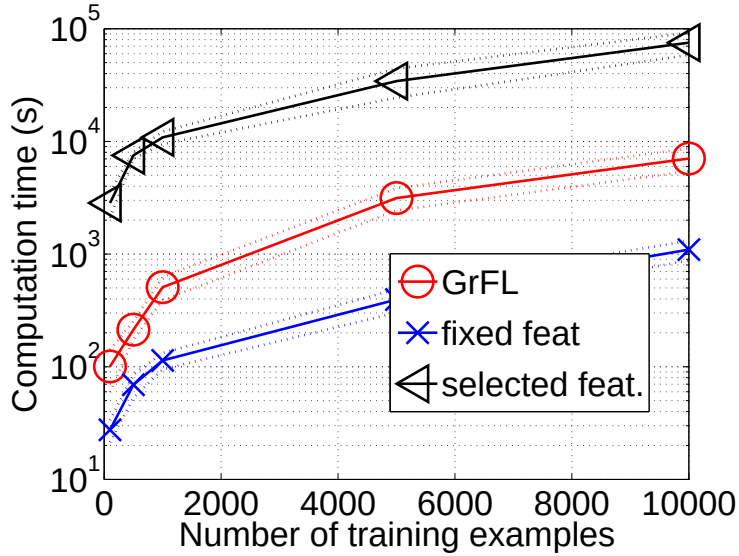




Gabor feature for texture recognition

- increasing number of training samples with 81 Gabor features





Lessons

- learning with infinitely many cheaper than learning with many
- do not sample parameters but take advantage of the continuous parameters

Large scale kernel machines

- Gaussian kernel with explicit and selected feature maps
- datasets : *Adult* and *IJCNN1* (40k and 110k training examples)
- sample kernel bandwidth and then sample vector direction

	Adult			IJCNN1		
# feat	GrFL	GrFL-M	IC	GrFL	GrFL-M	IC
10	**83.82**	83.77	83.38	**92.06**	91.96	91.03
50	84.76	**84.86**	84.58	**97.05**	96.97	92.19
100	84.98	**85.00**	84.84	97.97	**98.02**	93.29
500	85.24	**85.30**	85.04	–	–	–

	Adult			IJCNN1		
ratio	GrFL	GrFL-M	IC	GrFL	GrFL-M	IC
0.1	84.23	84.34	**84.54**	96.27	**96.67**	93.38
0.3	84.78	**84.87**	84.72	97.40	**97.77**	93.23
0.5	84.91	**84.95**	84.74	97.75	**97.96**	93.32
0.7	84.98	**85.00**	84.84	97.97	**98.02**	93.29

- Better performances than Incomplete Choleski decomposition
- Easy multiple Gaussian kernel

Conclusions

- Framework for learning with infinitely features that is generic to loss functions and sparsity inducing regularizers
- work pretty well from an empirical point of view
- Questions
  - Theoretical guarantees when the algorithm stops at non-optimal solution?
  - Are we sure that the selected features are "similar" to the true ones?