Convex sparse methods for feature hierarchies

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

ICML Workshop, June 2009

Learning with kernels is not dead

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

ICML Workshop, June 2009

Learning with kernels is not dead
Learning kernels is not dead either

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

ICML Workshop, June 2009

Smart shallow learning

Francis Bach

Willow project, INRIA - Ecole Normale Supérieure

ICML Workshop, June 2009

Outline

- Supervised learning and regularization
  - *Kernel methods vs. sparse methods*
- MKL: Multiple kernel learning
  - *Non linear sparse methods*
- HKL: Hierarchical kernel learning
  - *Feature hierarchies - non linear variable selection*

Supervised learning and regularization

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$
- Minimize with respect to function $f : \mathcal{X} \to \mathcal{Y}$:

$$\sum_{i=1}^{n} \ell(y_i, f(x_i)) \qquad + \qquad \frac{\mu}{2} \|f\|^2$$

Error on data + Regularization

Loss & function space ? Norm ?

- Two theoretical/algorithmic issues:

1. **Loss / energy**
2. **Function space / norm / architecture**

Regularizations

- Main goal: avoid overfitting

- Two main lines of work:

1. Euclidean and Hilbertian norms (i.e., ℓ^2-norms)
   - Non linear kernel methods

Regularizations

- Main goal: avoid overfitting

- Two main lines of work:

  1. Euclidean and Hilbertian norms (i.e., ℓ^2-norms)
     – Non linear kernel methods
  2. Sparsity-inducing norms
     – Usually restricted to linear predictors on vectors $f(x) = w^\top x$
     – Main example: ℓ_1-norm $\|w\|_1 = \sum_{i=1}^p |w_i|$
     – Perform model selection as well as regularization

Kernel methods: regularization by ℓ^2-norm

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, with **features** $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$
  - Predictor $f(x) = w^\top \Phi(x)$ linear in the features

- Optimization problem: $$\boxed{\min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\mu}{2} \|w\|_{\mathbf{2}}^2}$$

Kernel methods: regularization by ℓ^2-norm

- Data: $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $i = 1, \dots, n$, with **features** $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$
  - Predictor $f(x) = w^\top \Phi(x)$ linear in the features

- Optimization problem: $$\boxed{\min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top \Phi(x_i)) + \frac{\mu}{2}\|w\|_{\mathbf{2}}^2}$$

- **Representer theorem** (Kimeldorf and Wahba, 1971): solution must be of the form $w = \sum_{i=1}^n \alpha_i \Phi(x_i)$

  - Equivalent to solving: $$\boxed{\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i) + \frac{\mu}{2}\alpha^\top K \alpha}$$

  - Kernel matrix $K_{ij} = k(x_i, x_j) = \Phi(x_i)^\top \Phi(x_j)$

Kernel methods: regularization by ℓ^2-norm

- Running time $O(n^2\kappa + n^3)$ where κ complexity of one kernel evaluation (often much less) - **independent from p**

- **Kernel trick**: implicit mapping if $\kappa = o(p)$ by using only $k(x_i, x_j)$ instead of $\Phi(x_i)$

- Examples:
  - Polynomial kernel: $k(x,y) = (1 + x^\top y)^d \Rightarrow \mathcal{F} =$ polynomials
  - Gaussian kernel: $k(x,y) = e^{-\alpha\|x-y\|_2^2} \quad \Rightarrow \mathcal{F} =$ smooth functions
  - Kernels on structured data (see Shawe-Taylor and Cristianini, 2004)

Kernel methods: regularization by ℓ^2-norm

- Running time $O(n^2\kappa + n^3)$ where κ complexity of one kernel evaluation (often much less) - **independent from p**

- **Kernel trick**: implicit mapping if $\kappa = o(p)$ by using only $k(x_i, x_j)$ instead of $\Phi(x_i)$

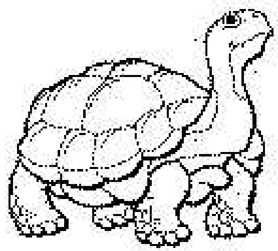
- Examples:
  - Polynomial kernel: $k(x,y) = (1 + x^\top y)^d \Rightarrow \mathcal{F} =$ polynomials
  - Gaussian kernel: $k(x,y) = e^{-\alpha\|x-y\|_2^2} \Rightarrow \mathcal{F} =$ smooth functions
  - Kernels on structured data (see Shawe-Taylor and Cristianini, 2004)

- $+$: Implicit non linearities and high-dimensionality

- $-$: Problems of interpretability, dimension really high?

Kernel methods are "not" infinite-dimensional

- Usual message: "learning with infinite dimensions in finite time"
- But infinite number of features of **rapidly decaying magnitude**
  - Mercer expansion: $k(x,y) = \sum_{p=1}^{\infty} \lambda_i \varphi_i(x) \varphi_i(y)$
  - $(\lambda_i)_i$ convergent series
- Zenon's paradox (Achilles and the tortoise)



ℓ_1-norm regularization (linear setting)

- Data: covariates $x_i \in \mathbb{R}^p$, responses $y_i \in \mathcal{Y}$, $i = 1, \dots, n$

- Minimize with respect to loadings/weights $w \in \mathbb{R}^p$:

$$\sum_{i=1}^{n} \ell(y_i, w^\top x_i) \quad + \quad \mu \|w\|_1$$

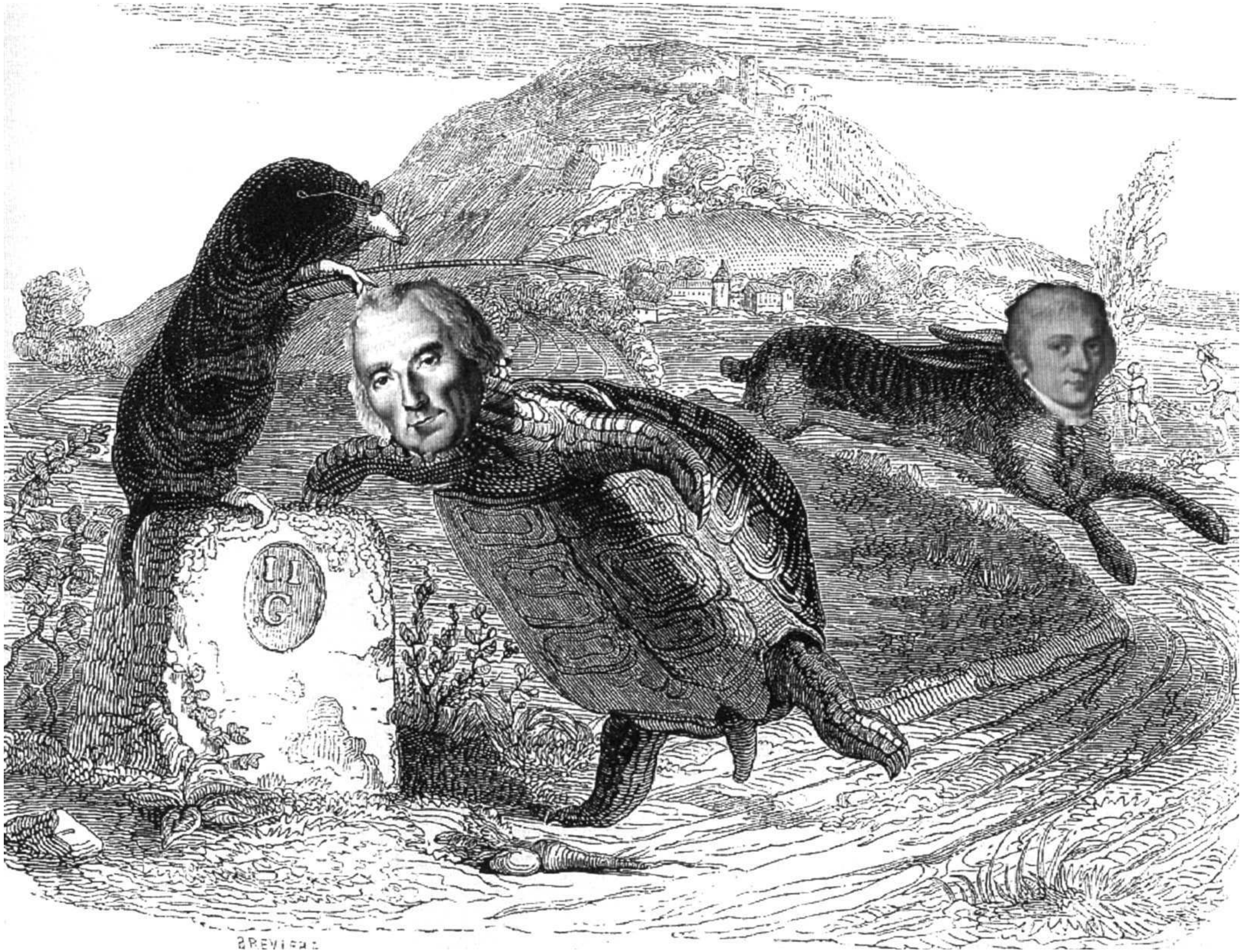
$$\text{Error on data} \quad + \quad \text{Regularization}$$

- square loss $\Rightarrow$ basis pursuit (signal processing) (Chen et al., 2001), Lasso (statistics/machine learning) (Tibshirani, 1996)

ℓ^2-norm vs. ℓ^1-norm

- ℓ^1-norms lead to interpretable models
- ℓ^2-norms can be run implicitly with "very large" feature spaces
- **Algorithms**:
  - Smooth convex optimization vs. nonsmooth convex optimization
- **Theory**:
  - better predictive performance?

ℓ^2 vs. ℓ^1 - Gaussian hare vs. Laplacian tortoise



- First-order methods (Fu, 1998; Wu and Lange, 2008)
- Homotopy methods (Markowitz, 1956; Efron et al., 2004)

Lasso - Two main recent theoretical results

1. **Consistency condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007)

2. **Exponentially many irrelevant variables** (Zhao and Yu, 2006; Wainwright, 2006; Bickel et al., 2008; Lounici, 2008; Meinshausen and Yu, 2009): under appropriate assumptions, consistency is possible as long as

$$\log p = O(n)$$

Lasso - Two main recent theoretical results

1. **Consistency condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007)

2. **Exponentially many irrelevant variables** (Zhao and Yu, 2006; Wainwright, 2006; Bickel et al., 2008; Lounici, 2008; Meinshausen and Yu, 2009): under appropriate assumptions, consistency is possible as long as

$$\log p = O(n)$$

- Question: is it possible to build a sparse algorithm that can learn from more than 10^{80} features?

Lasso - Two main recent theoretical results

1. **Consistency condition** (Zhao and Yu, 2006; Wainwright, 2006; Zou, 2006; Yuan and Lin, 2007)

2. **Exponentially many irrelevant variables** (Zhao and Yu, 2006; Wainwright, 2006; Bickel et al., 2008; Lounici, 2008; Meinshausen and Yu, 2009): under appropriate assumptions, consistency is possible as long as

$$\log p = O(n)$$

- Question: is it possible to build a sparse algorithm that can learn from more than 10^{80} features?
  - **Some type of recursivity/factorization is needed!**

Outline

- Supervised learning and regularization
  - *Kernel methods vs. sparse methods*
- MKL: Multiple kernel learning
  - *Non linear sparse methods*
- HKL: Hierarchical kernel learning
  - *Feature hierarchies - non linear variable selection*

Multiple kernel learning - MKL (Lanckriet et al., 2004; Bach et al., 2004)

- Kernels $k_v(x,x') = \Phi_v(x)^\top \Phi_v(x')$ on the same input space, $v \in V$
- Concatenation of features $\Phi(x) = (\Phi_v(x))_{v \in V}$ equivalent to summing kernels

$$k(x,x') = \Phi(x)^\top \Phi(x') = \sum_{v \in V} \Phi_v(x)^\top \Phi_v(x') = \sum_{v \in V} k_v(x,x')$$

- If predictors $w = (w_v)_{v \in V}$, then penalizing by $\left(\sum_{v \in V} \|w_v\|_2\right)^2$
  - will induce sparsity at the kernel level (many w_v equal to zero)
  - is equivalent to learn a sparse positive combination $\displaystyle\sum_{v \in V} \eta_v k_v(x,x')$
- NB: penalizing by $\sum_{v \in V} \|w_v\|_2^2$ is equivalent to uniform weights

Hierarchical kernel learning - HKL (Bach, 2008)

- Many kernels can be decomposed as a sum of many "small" kernels

$$k(x,x') = \sum_{v \in V} k_v(x,x')$$

- Example with $x = (x_1, \dots, x_q) \in \mathbb{R}^q$ ($\Rightarrow$ non linear variable selection)
  - Gaussian/ANOVA kernels: $p = \#(V) = 2^q$

$$\prod_{j=1}^{q} \left(1 + e^{-\alpha(x_j - x_j')^2}\right) = \sum_{J \subset \{1,\dots,q\}} \prod_{j \in J} e^{-\alpha(x_j - x_j')^2} = \sum_{J \subset \{1,\dots,q\}} e^{-\alpha \|x_J - x'_J\|_2^2}$$

- **Goal**: learning sparse combination $\sum_{v \in V} \eta_v k_v(x,x')$

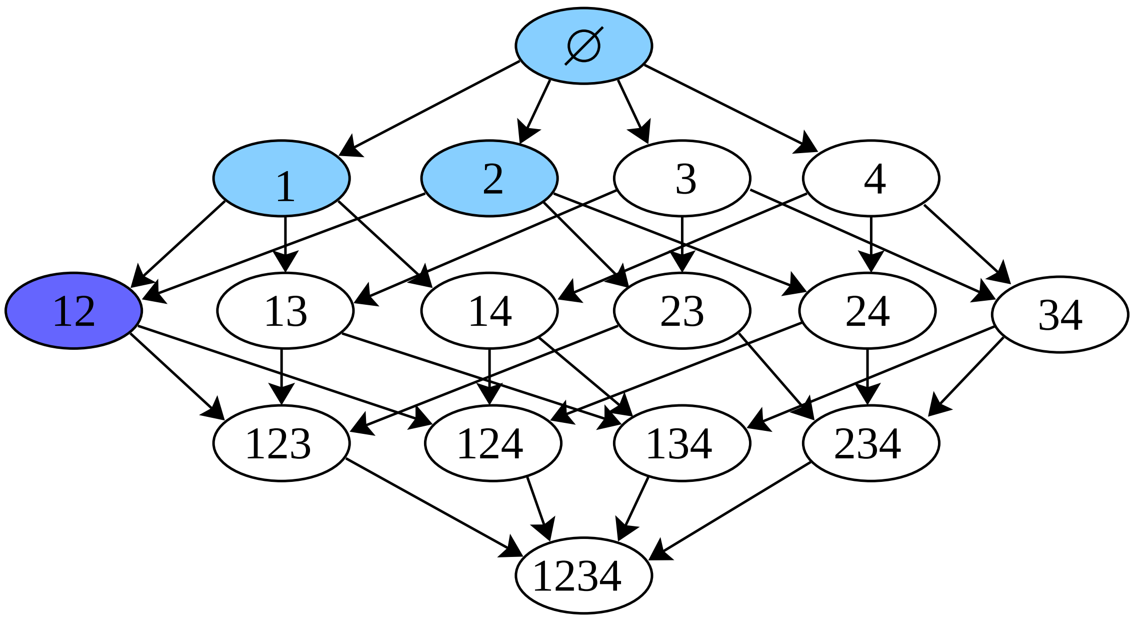
Restricting the set of active kernels

- With flat structure
  - Consider block ℓ^1-norm: $\sum_{v \in V} \|w_v\|_2$
  - cannot avoid being linear in $p = \#(V)$

- Using the structure of the small kernels
  - for computational reasons
  - to allow more irrelevant variables

Restricting the set of active kernels

- V is endowed with a directed acyclic graph (DAG) structure: **select a kernel only after all of its ancestors have been selected**
- Gaussian kernels: $V =$ power set of $\{1, \dots, q\}$ with inclusion DAG
  - Select a subset only after all its subsets have been selected

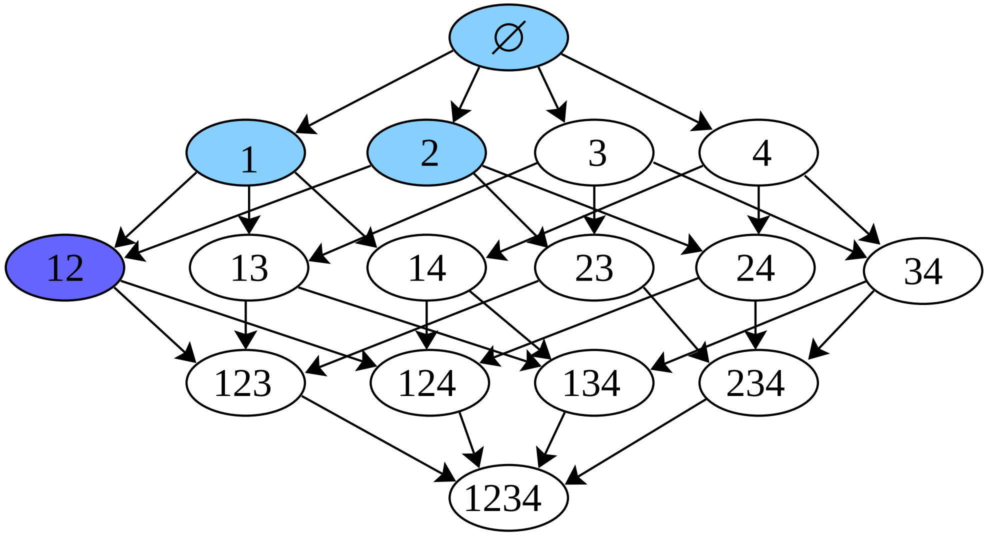


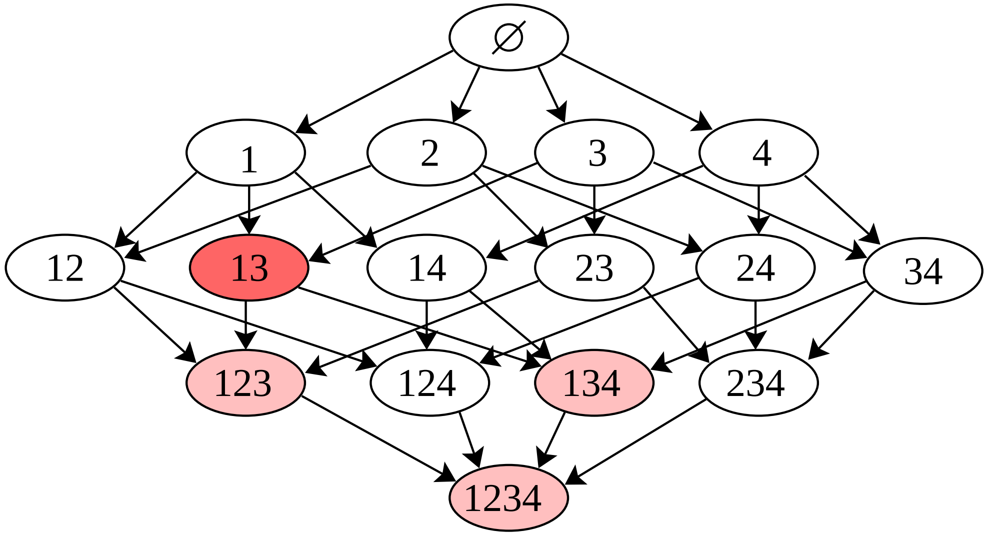
DAG-adapted norm (Zhao & Yu, 2008)

- Graph-based structured regularization
  - $\mathrm{D}(v)$ is the set of descendants of $v \in V$:

$$\sum_{v \in V} \|w_{\mathrm{D}(v)}\|_2 = \sum_{v \in V} \Big(\sum_{t \in \mathrm{D}(v)} \|w_t\|_2^2 \Big)^{1/2}$$

- Main property: If v is selected, so are all its ancestors





DAG-adapted norm (Zhao & Yu, 2008)

- Graph-based structured regularization
  - $\mathrm{D}(v)$ is the set of descendants of $v \in V$:

$$\sum_{v \in V} \|w_{\mathrm{D}(v)}\|_2 = \sum_{v \in V} \left(\sum_{t \in \mathrm{D}(v)} \|w_t\|_2^2 \right)^{1/2}$$

- Main property: If v is selected, so are all its ancestors

- Questions :
  - **polynomial-time** algorithm for this norm?
  - **necessary/sufficient conditions** for consistent kernel selection?
  - **Scaling between p, q, n** for consistency?
  - **Applications** to variable selection or other kernels?

Active set algorithm for sparse problems

- First assume that the set J of active kernels is known
  - If J is small, solving the reduced problem is easy
  - Simply need to check if the solution is optimal for the full problem
    - If yes, the solution is found
    - If not, add violating variables to the reduced problem

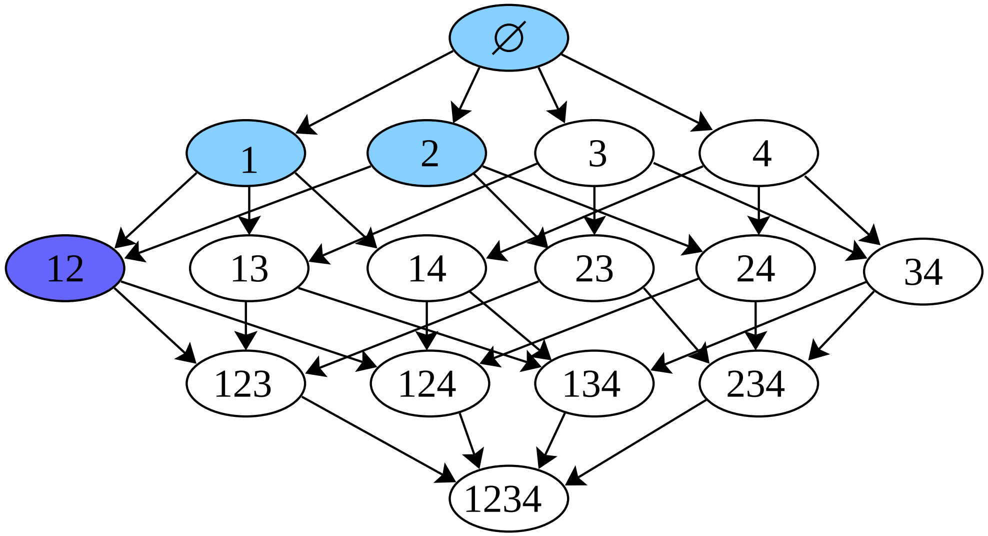
Active set algorithm for sparse problems

- First assume that the set J of active kernels is known
  - If J is small, solving the reduced problem is easy
  - Simply need to check if the solution is optimal for the full problem
    - If yes, the solution is found
    - If not, add violating variables to the reduced problem

- **Technical issue**: computing approximate necessary and sufficient conditions in polynomial time in the out-degree of the DAG
  - NB: with flat structure, this is linear in $p = \#(V)$

- **Active set algorithm**: start with the roots of the DAG and grow
  - Running time polynomial in the number of selected kernels

Consistency of kernel selection (Bach, 2008)



- Because of the selection constraints, getting the exact sparse model is not possible in general

- May only estimate the *hull* of the relevant kernels

- Necessary and sufficient conditions can be derived

Scaling between p, q, n

$$
\begin{aligned}
n &= \text{number of observations} \\
q &= \text{maximum out degree in the DAG} \\
p &= \text{number of vertices in the DAG}
\end{aligned}
$$

- **Theorem**: Assume consistency condition satisfied, Gaussian noise with variance σ^2, and $\lambda = c_1\sigma\left(\frac{\log q}{n}\right)^{1/2} \leqslant c_2$; the probability of incorrect hull selection is less than c_3/q.

Scaling between p, q, n

$$\begin{aligned} n &= \text{number of observations} \\ q &= \text{maximum out degree in the DAG} \\ p &= \text{number of vertices in the DAG} \end{aligned}$$

- **Theorem**: Assume consistency condition satisfied, Gaussian noise with variance σ^2, and $\lambda = c_1\sigma\left(\frac{\log q}{n}\right)^{1/2} \leqslant c_2$; the probability of incorrect hull selection is less than c_3/q.

- Unstructured case: $q = p \Rightarrow \boxed{\log p = O(n)}$

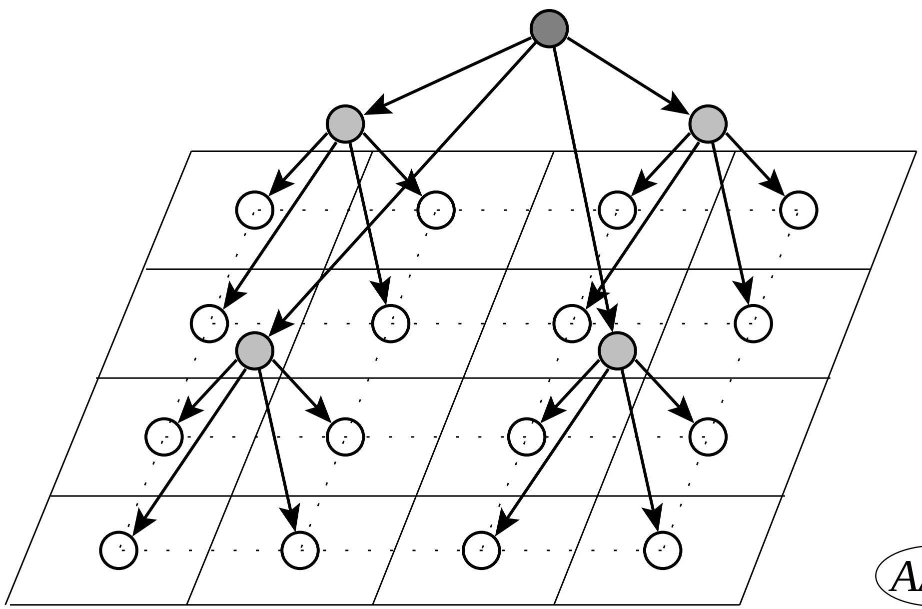
- Power set of q elements: $q \approx \log p \Rightarrow \boxed{\log\log p = \log q = O(n)}$

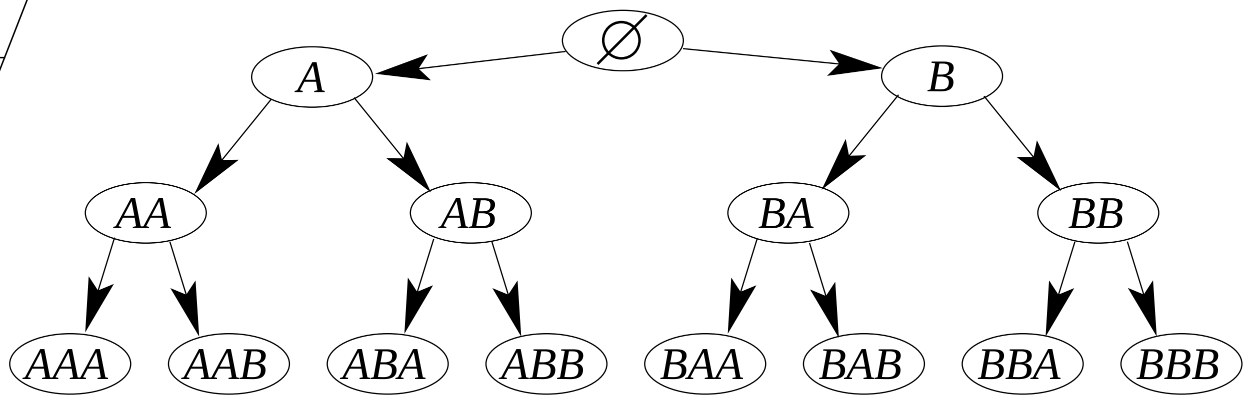
Mean-square errors (regression)

dataset	n	p	k	$\#(V)$	L2	greedy	MKL	HKL
abalone	4177	10	pol4	$\approx 10^7$	44.2±1.3	43.9±1.4	44.5±1.1	**43.3±1.0**
abalone	4177	10	rbf	$\approx 10^{10}$	**43.0±0.9**	45.0±1.7	43.7±1.0	43.0±1.1
boston	506	13	pol4	$\approx 10^9$	**17.1±3.6**	24.7±10.8	22.2±2.2	18.1±3.8
boston	506	13	rbf	$\approx 10^{12}$	**16.4±4.0**	32.4±8.2	20.7±2.1	17.1±4.7
pumadyn-32fh	8192	32	pol4	$\approx 10^{22}$	57.3±0.7	56.4±0.8	**56.4±0.7**	56.4±0.8
pumadyn-32fh	8192	32	rbf	$\approx 10^{31}$	57.7±0.6	72.2±22.5	56.5±0.8	**55.7±0.7**
pumadyn-32fm	8192	32	pol4	$\approx 10^{22}$	6.9±0.1	6.4±1.6	7.0±0.1	**3.1±0.0**
pumadyn-32fm	8192	32	rbf	$\approx 10^{31}$	5.0±0.1	46.2±51.6	7.1±0.1	**3.4±0.0**
pumadyn-32nh	8192	32	pol4	$\approx 10^{22}$	84.2±1.3	73.3±25.4	83.6±1.3	**36.7±0.4**
pumadyn-32nh	8192	32	rbf	$\approx 10^{31}$	56.5±1.1	81.3±25.0	83.7±1.3	**35.5±0.5**
pumadyn-32nm	8192	32	pol4	$\approx 10^{22}$	60.1±1.9	69.9±32.8	77.5±0.9	**5.5±0.1**
pumadyn-32nm	8192	32	rbf	$\approx 10^{31}$	15.7±0.4	67.3±42.4	77.6±0.9	**7.2±0.1**

Extensions to other kernels

- Extension to graph kernels, string kernels, pyramid match kernels





- Exploring large feature spaces with structured sparsity-inducing norms
  - Interpretable models
- Other structures than hierarchies or DAGs

Conclusions - Discussion

Shallow, but not stupid

- **Learning with a flat architecture and exponentially many features is possible**
  - Theoretically
  - Algorithmically

Conclusions - Discussion
Shallow, but not stupid

- **Learning with a flat architecture and exponentially many features is possible**
  - Theoretically
  - Algorithmically

- **Deep vs. Shallow**
  - non-linearities are important
  - multi-task learning is important
  - Problems are non-convex: convexity vs. non convexity
  - Theoretical guarantees vs. empirical evidence
  - Dealing with prior knowledge / structured data - Interpretability
  - Learning / engineering / sampling intermediate representations

References

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Adv. NIPS*, 2008.

F. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

P. J. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 2008. To appear.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001. ISSN 0036-1445.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32:407, 2004.

W. Fu. Penalized regressions: the bridge vs. the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998).

G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applicat.*, 33:82–95, 1971.

G. R. G. Lanckriet, N. Cristianini, L. El Ghaoui, P. Bartlett, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 34(5):2272–2297, 2006.

K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electronic Journal of Statistics*, 2, 2008.

H. M. Markowitz. The optimization of a quadratic function subject to linear constraints. *Naval Research Logistics Quarterly*, 3:111–133, 1956.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Stat.*, 2009. to appear.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Camb. U. P., 2004.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of The Royal Statistical Society Series B*, 58(1):267–288, 1996.

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming. Technical Report 709, Dpt. of Statistics, UC Berkeley, 2006.

T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of The Royal Statistical Society Series B*, 69(2):143–161, 2007.

P. Zhao and B. Yu. On model selection consistency of Lasso. *JMLR*, 7:2541–2563, 2006.

H. Zou. The adaptive Lasso and its oracle properties. *JASA*, 101:1418–1429, 2006.