STATG019 – Selected Topics in Statistics 2015

Lecture 4

Kernel Methods for Big Data

Dr Franz J. Király

Course organization

In-Course-Assessment

Two take-home ICA, one on kernels, one on point processes

Each counts 50% towards your final grade

Handing out: no.1 on Feb 9, no.2 on Mar 23 (on moodle)
Submission: no.1 on Mar 4, no.2 on Apr 29 (via moodle/TurnitIn)

Submission details for ICA no.1 will be announced

Tutorials and/or practical sessions

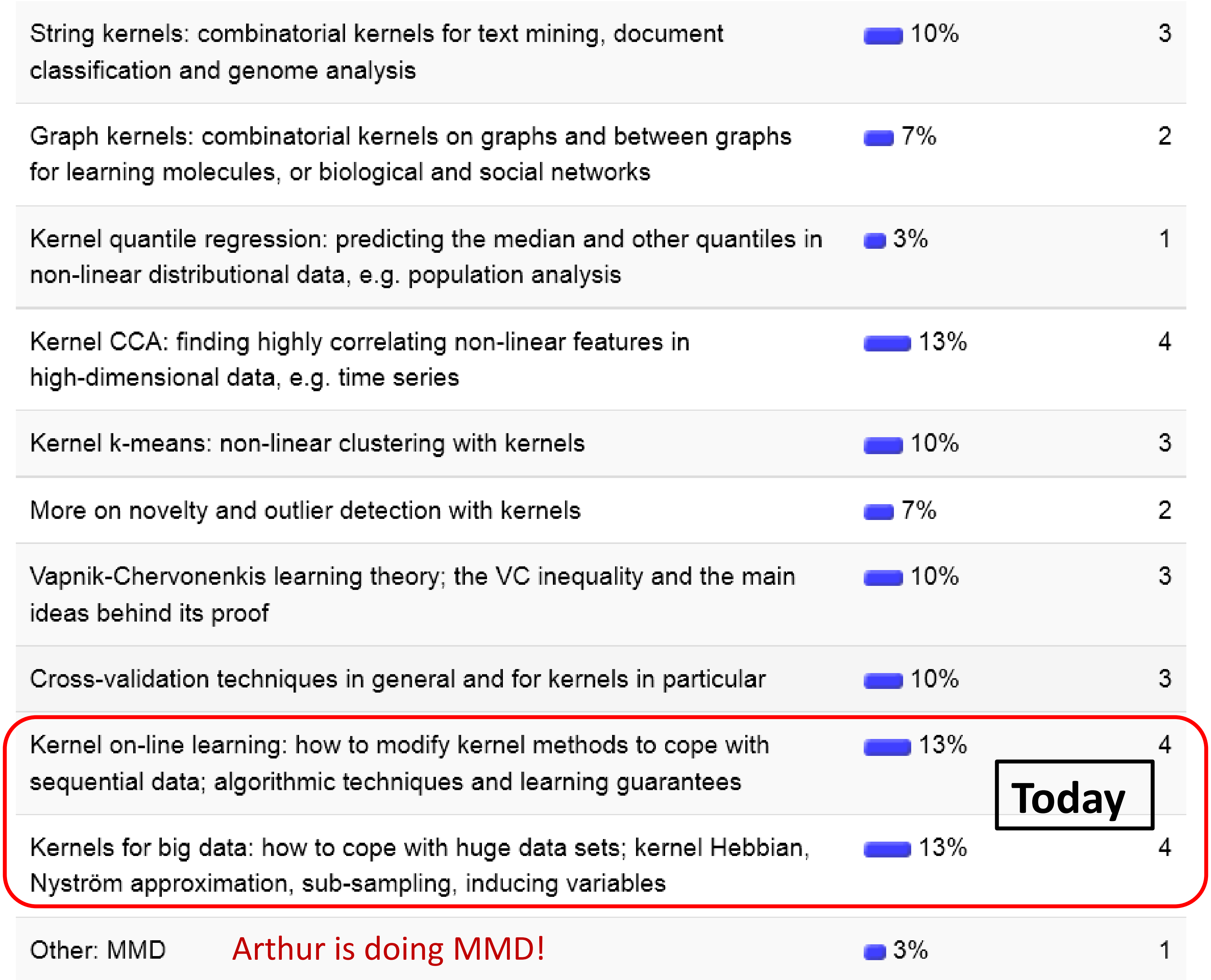
Thursday, 11am - 1pm, February 26

Tutorial will be mainly on mathematical concepts and the exercises

Also a bit of R and programming issues, if we have the time

Thursday, 11am - 1pm, February 12?

R programming, kernlab, help on getting started with the ICA?

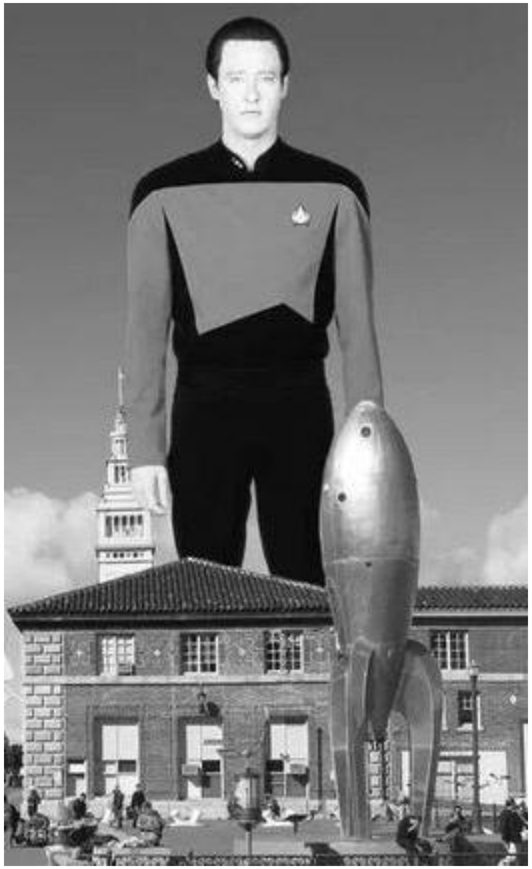


String kernels: combinatorial kernels for text mining, document classification and genome analysis	10%	3
Graph kernels: combinatorial kernels on graphs and between graphs for learning molecules, or biological and social networks	7%	2
Kernel quantile regression: predicting the median and other quantiles in non-linear distributional data, e.g. population analysis	3%	1
Kernel CCA: finding highly correlating non-linear features in high-dimensional data, e.g. time series	13%	4
Kernel k-means: non-linear clustering with kernels	10%	3
More on novelty and outlier detection with kernels	7%	2
Vapnik-Chervonenkis learning theory; the VC inequality and the main ideas behind its proof	10%	3
Cross-validation techniques in general and for kernels in particular	10%	3
Kernel on-line learning: how to modify kernel methods to cope with sequential data; algorithmic techniques and learning guarantees	13%	4
Kernels for big data: how to cope with huge data sets; kernel Hebbian, Nyström approximation, sub-sampling, inducing variables	13%	4
Other: MMD	3%	1

Today

Arthur is doing MMD!

Kernel learning in the presence of BIG DATA



Input: independent data points $x_1, \ldots, x_N \in \mathbb{R}^n$
when supervised: labels $y_1, \ldots, y_N \in \mathbb{R}$

Kernel ridge regression:

$$f(x) = y^\top (K + \lambda I)^{-1} \cdot \kappa(x) = \widehat{\alpha}^\top \kappa(x)$$

$$K = (k(x_i, x_j))_{ij} \qquad \kappa(x) = (k(x_i, x))_i$$

Kernel PCA: $(U, \lambda) = \text{eig}(K_\mu) \qquad \tau_i(x) = \tau_i(U, \lambda, \kappa(x), K)$

Kernel SVM:

$$f(x) = \text{sgn}\left(b + \sum_{i=1}^{N} \alpha_i k(x_i, x)\right) = \text{sgn}\left(\alpha^\top \kappa(x)\right)$$

$$\max_\alpha W(\alpha) = \|\alpha\|_1 - \frac{1}{2} \cdot \alpha^\top \tilde{K} \alpha \qquad \text{s.t.} \quad \alpha \geq 0, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

Naïve implementation costs $O(N^2)$ to $O(N^3)$

anything but $O\left(N \cdot \text{polylog}(N)\right)$ is infeasible for $N \approx 10^6$

Idea no.1: low-rank approximation

main idea: $K \approx Q \cdot Q^\top$, with $Q \in \mathbb{R}^{N \times M}$

operate with tall matrices only, e.g. Q

yields low-dimensional feature vectors

no.2: iterative/incremental methods

main idea: add x_i incrementally

update f, τ, etc, for each addition

directly applicable to on-line setting

Low-Rank Approximation

The Pseudo-inverse and row-span projectors

Definition (Penrose-Moore-pseudoinverse):

Let $A \in \mathbb{R}^{m\times n}$ be a matrix.

Pseudoinverse of A is matrix A^+ such that

$$AA^+A = A \qquad A^+AA^+ = A^+ \qquad (AA^+)^\top = AA^+ \qquad (A^+A)^\top = A^+A$$

Proposition:

Given $A \in \mathbb{R}^{m\times n}$, the pseudo-inverse A^+ exists and is unique.

Existence: Let $A = USV^\top$ the SVD of A, take $A^+ := VS^+U^\top$

with diagonal matrix S^+, where $S^+_{ii} = \begin{cases} 1/S_{ii}, & \text{if } S_{ii} \neq 0, \\ 0, & \text{if } S_{ii} = 0. \end{cases}$

Uniqueness: For two pseudo-inverses B, C of A:

$$C = CAC = CC^\top \cdot (ABA)^\top = C \cdot (ACAB) = CAB = BAB = B$$

Remark: For a data matrix $X \in \mathbb{R}^{N\times n}$

projection matrix onto the row-span of X is given as

$$P_X := X^+X = X^\top(XX^\top)^+X$$

since $P_XP_X = P_X$ and $P_Xv = 0$ for any $v \in \mathbb{R}^n$ with $Xv = 0$

in the middle: (pseudo-)inverse of Gram matrix!

The Nyström approximation (E. Nyström, 1928; idea in the context of integral equations)

Remark: projection matrix onto the row-span of X is given as $P_X := X^\top (XX^\top)^+ X$

Main idea: maybe rowspan X is well-approximated by rowspan Z

where $Z \in \mathbb{R}^{M\times n}$ consists of M rows of $X \in \mathbb{R}^{N\times n}$

so $XX^\top \approx (XP_Z)(XP_Z)^\top = XP_ZX^\top = (XZ^\top)(ZZ^\top)^+(ZX^\top)$

Rewriting with $K_{AB} := AB^\top$: $K_{XX} \approx K_{XZ} \cdot K_{ZZ}^+ \cdot K_{ZX} =: K_{XX|Z}$

Proposition (deterministic approximation):

(i) the residual matrix $K_{XX} - K_{XX|Z}$ is positive semi-definite

(ii) $\lambda_i(K_{XX}) \geq \lambda_i(K_{XX|Z}) \geq \lambda_{i+\Delta}(K_{XX})$ where $\Delta = \text{rank } X - \text{rank } Z$
(by Weyl's theorem) and λ_i denotes the i-th largest eigenvalue

(iii) $\|K_{XX} - K_{XX|Z}\|_F \leq \text{Tr}(K_{XX}) - \text{Tr}(K_{ZZ})$ (use positive semi-definiteness)

Theorem (Kumar et al, NIPS 2009): **(one example for probabilistic approximation)**

Denote by $K_{XX}^{(r)}$ the Frobenius-best rank r approximation to K_{XX}.

Assume Z is uniformly subsampled. If $M \geq 64\frac{r}{\varepsilon^4}$, then

$$\mathbb{E}\|K_{XX} - K_{XX|Z}\|_F \leq \|K_{XX} - K_{XX}^{(r)}\|_F + \varepsilon \cdot \max_i (N \cdot k(x_i, x_i))$$

Remark: Data dimension n does not enter the proofs!
so statements are fine for arbitrary kernels

(more precisely: use existence of Cholesky decomposition)

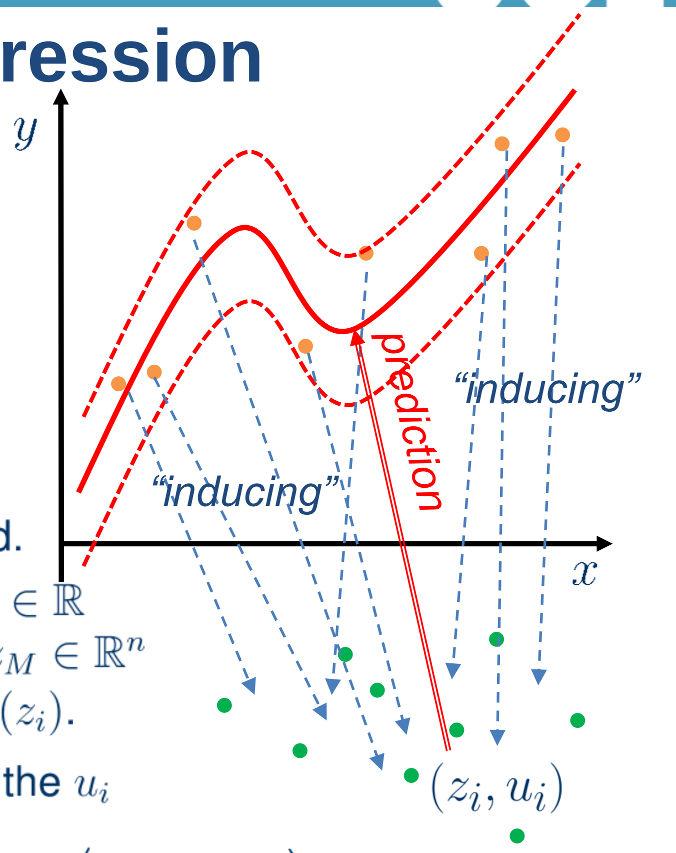
Speeding up kernels with Nytröm

Nytsröm-approximation: $K_{XX} \approx K_{XZ} \cdot K_{ZZ}^{+} \cdot K_{ZX} =: K_{XX|Z}$

Idea for speed-up: $K_{XX|Z} = \left(K_{XZ} \cdot K_{ZZ}^{-1/2}\right) \cdot \left(K_{XZ} \cdot K_{ZZ}^{-1/2}\right)^{\top} =: K_{X|Z} \cdot K_{X|Z}^{\top}$

		time cost
kPCA	**Vanilla:** compute eigenpairs of $K_{\mu} := (I - \mathbb{1}_N) \cdot K_{XX} \cdot (I - \mathbb{1}_N)$	$O(N^2 m)$
	Nyström: compute left singular pairs of $(I - \mathbb{1}_N) \cdot K_{X\vert Z}$	$O(NMm)$ for m pairs
SVM	**Vanilla:** $\max_{\alpha} W(\alpha) = \lVert\alpha\rVert_1 - \frac{1}{2} \cdot \alpha^{\top} \tilde{K} \alpha$ $\quad \tilde{K} = \text{diag}(y) K_{XX} \text{diag}(y)$	$O(h(N))$
	Nyström: restrict α to column-span of diag$(y) K_{X\vert Z}$	
	$\tilde{\alpha} := K_{X\vert Z}^{\top} \text{diag}(y) \cdot \alpha$ so $\alpha = \left(K_{X\vert Z}^{\top} \text{diag}(y)\right)^{+} \tilde{\alpha}$	$O(\frac{N}{M} h(M))$
Ridge regression	**Vanilla:** compute $\widehat{\alpha} = (K_{XX} + \lambda I)^{-1} y$ then $f(x) = \widehat{\alpha}^{\top} \kappa(x)$	$O(N^3)$
	Nyström: $(K_{XX} + \lambda I)^{-1} \approx (K_{XX\vert Z} + \lambda I)^{-1} = (K_{X\vert Z} K_{X\vert Z}^{\top} + \lambda I)^{-1}$	
	observe, going backwards in the derivation of ridge regression	
	$(K_{X\vert Z} K_{X\vert Z}^{\top} + \lambda I)^{-1} = \lambda^{-1} I - \lambda^{-1} K_{X\vert Z} \left(K_{X\vert Z}^{\top} K_{X\vert Z} + \lambda I\right)^{-1} K_{X\vert Z}^{\top}$	
	alternative: the Woodbury identity (later today)	
	so compute $\widehat{\alpha} = \lambda^{-1} y - \lambda^{-1} K_{X\vert Z} \left(K_{X\vert Z}^{\top} K_{X\vert Z} + \lambda I\right)^{-1} K_{X\vert Z}^{\top} \cdot y$	$O(NM^2)$

Sparse Gaussian processes regression



Input: independent data points $x_1, \dots, x_N \in \mathbb{R}^n$
dependent data points/labels $y_1, \dots, y_N \in \mathbb{R}$

Output: Regressor $f : \mathbb{R}^n \to \mathbb{R}$ such that $y_i \approx f(x_i)$
and which predicts well unseen labels $f(x)$

Main assumptions:

The "true" f is outcome of Gaussian process
jointly Gaussian $y_i = f(x_i) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \lambda)$ i.i.d.

There are jointly Gaussian "inducing variables" $u_1, \dots, u_M \in \mathbb{R}$
corresponding to (hypothetical) pseudo-inputs $z_1, \dots, z_M \in \mathbb{R}^n$
(hypothetially) observed without noise, i.e., $u_i = f(z_i)$.

test label $f(x)$ and $f(x_i)$ are independent conditioned on the u_i

Derivation of posterior distribution: (in the case $x \neq x_1, \dots, x_N$)

By assumption: $y_1, \dots, y_N | u_1, \dots, u_M \sim \mathcal{N}\left(K_{XZ}K_{ZZ}^{+}u, K_{XX} - K_{XX|Z}\right)$

$$f(x) | u_1, \dots, u_M \sim \mathcal{N}\left(K_{xZ}K_{ZZ}^{+}u, k(x,x) - K_{xx|Z}\right)$$

Marginalization over u_i yields

$$f(x)|y_1, \dots, y_N \sim \mathcal{N}\left(K_{xX|Z}(K_{XX|Z} + \lambda \cdot I)^{-1}y,\ K_{xx|Z} - K_{xX|Z}(K_{XX|Z} + \lambda \cdot I)^{-1}K_{Xx|Z}\right)$$

equivalent to GP regression with covariance function $k(x,y) = K_{xy|Z} = K_{xZ}K_{ZZ}^{+}K_{Zy}$

mean and variance efficiently computable in analogy to ridge regression $O(NM^2)$

how to choose Z for the Nyström approximation $K_{XX} \approx K_{XX|Z}$

Williams, Seeger (2001): uniformly random sub-sample

will be fine due to $\|K_{XX} - K_{XX|Z}\|_F \leq \mathrm{Tr}(K_{XX}) - \mathrm{Tr}(K_{ZZ})$

and $\mathbb{E}\|K_{XX} - K_{XX|Z}\|_F \leq \|K_{XX} - K^{(r)}_{XX}\|_F + \varepsilon \cdot \max_i (N \cdot k(x_i, x_i))$

Pros: works very often, fast **Cons:** slow on degenerate/clustered data

Fine et al, Bach et al (2001/2): Greedy for approximation (unsupervised)

Bach et al (2005): Greedy for goodness of prediction (supervised)

due to factorization method called “incomplete Cholesky decomposition”

Pros: informed, thus converges quickly **Cons:** greedy $\neq$ optimal

Snelson et al (2006): Bayesian inference on inducing variables

Pros: Bayesian **Cons:** Bayesian

Kiraly et al (2014): sample rows of Z i.i.d. from random variable $\mathcal{Z}$

Theorem: $\|K_{XX} - K_{XX|Z}\|_F = O(M^{-1/4})$ (if support of $\mathcal{Z}$ covers X) (constants depending on $\mathcal{Z}, k$)

Pros: independent of data degeneracies **Cons:** slower convergence (constants)

An overview can also be found in Kumar (2012): Sampling methods for the Nyström method

Iterative kernel methods

Low-rank updates for inversion

Proposition (Sherman, Morrison, Woodbury):

"the Woodbury formula"

(despite Sherman & Morrison being earlier)

$$(A+UV)^{-1} = A^{-1} - A^{-1}U(I+VA^{-1}U)^{-1}VA^{-1}$$

for matrices of the right size where all the above is invertible

Proof: verify that $(A+UV)\cdot$ the right hand side $= I$ (or use the SVD argument)

On-line learning update for ridge regression/Gaussian processes:

$X \in \mathbb{R}^{N\times n}$ old data batch $y \in \mathbb{R}^{N}$ old labels $Z \in \mathbb{R}^{M\times n}$ row-subsample

$Y \in \mathbb{R}^{\ell\times n}$ new data batch $y' \in \mathbb{R}^{\ell}$ new labels

old coefficients: $\widehat{\alpha} = \lambda^{-1}y - \lambda^{-1}BK_{X|Z}^{\top}\cdot y$ where $B = \left(K_{X|Z}^{\top}K_{X|Z} + \lambda I\right)^{-1}$

naive recomputation comes at a cost of $O(NM^2)$

new coeff's: $\widehat{\alpha}_{new} = \lambda^{-1}(y,y') - \lambda^{-1}K_{X,Y|Z}\cdot B'\cdot K_{X,Y|Z}^{\top}\cdot(y,y')$ where $K_{X,Y|Z} = \begin{bmatrix} K_{X|Z} \\ K_{Y|Z} \end{bmatrix}$

and where $B' = \left(K_{X,Y|Z}^{\top}K_{X,Y|Z} + \lambda I\right)^{-1} = \left(K_{X|Z}^{\top}K_{X|Z} + K_{Y|Z}^{\top}K_{Y|Z} + \lambda I\right)^{-1}$

SMW-formula yields $B' = B - BK_{Y|Z}^{\top}(I + K_{Y|Z}\cdot B\cdot K_{Y|Z}^{\top})^{-1}K_{Y|Z}B$

so B-matrix and $\widehat{\alpha}$ can be updated at time cost of $O(NM\ell)$

Similar updates can be derived for posterior variance and Z

Note that repeated updating may lead to cumulative time cost of $O(N^2M)$

The (kernel) Hebbian algorithm

or Sanger's rule (1989)
(for neural networks)

Goal: Given data matrix $X \in \mathbb{R}^{N \times n}$, whose rows are presented at times $t = 1, 2, \ldots$ as an n-variate time series $x(t)$ compute eigenvectors of $X^\top X =$ Cov(X, X)

Idea (Sanger): sequential update with candidate matrix $W(t) \in \mathbb{R}^{r \times n}$ (rows=eigenvectors)

$$W(t+1) = W(t) + \gamma(t) \left(y(t)x(t)^\top - \text{LT}\left[y(t)y(t)^\top\right] W(t) \right) \quad \text{where} \quad y(t) = W(t)x(t)$$

$\gamma(t)$ is "learning rate" LT sets everyting above diagonal to zero

Sanger's rule combines gradient descent (Oja's rule) and Gram-Schmidt

Theorem (Oja, Sanger, 1982/1989): converges under very mild conditions

Kernelization: compute eigenvectors of $K = XX^\top$

by a search in the feature span of the data: $W(t) = A(t)X^\top$ (compare representer thm)

$$A(t+1)X^\top = A(t)X^\top + \gamma(t) \left(y(t)x(t)^\top - \text{LT}\left[y(t)y(t)^\top\right] A(t)X^\top \right)$$

If the columns of X are linearly independent,
then right multiplication with $X^\top$ is injective, so

$$A(t+1) = A(t) + \gamma(t) \left(y(t)e_{i(t)}^\top - \text{LT}\left[y(t)y(t)^\top\right] A(t) \right)$$

where $e_{i(t)}$ denotes the standard unit vector selecting $x(t)$

Theorem (Kim et al, 2003): converges under very mild conditions

Günter et al, 2007: even faster (empirically) for a good $\gamma(t)$

$O(Nr)$ per iteration

Stochastic gradient descent on regularized risk

Setting: Given sequential data $x(t) \in \mathbb{R}^n$, labels $y(t) \in \mathbb{R}$

Learn predictor/classifier f such that $f(x(t)) \approx y(t)$

Idea (various): sequential update of candidate predictor f_t

$$f_{t+1} = f_t - \gamma(t) \cdot \frac{\partial}{\partial f} R_{reg,t}(f)|_{f=f_t} = f_t - \gamma(t) \cdot \frac{\partial}{\partial f} \ell(f, x(t), y(t)) + \frac{\lambda}{2} \|f\|^2|_{f=f_t}$$

$\gamma(t)$ is "learning rate" ℓ is some loss, e.g. $\ell = (y(t) - f_t(x(t)))^2$

by representer theorem, current minimizer $f_t(.) = \sum_{\tau=1}^{t} \alpha_\tau k(x(\tau), .)$

so $f_{t+1} = (1 - \gamma(t)\lambda) \cdot f_t - \gamma(t) \cdot \ell'(f_t(x(t)), y(t)) \cdot k(x(t), .)$

Observation: f_t is "phased out" in f_{t+m} by factor $(1 - \gamma(t)\lambda)^m$

(so $\gamma(t)$ is rather the "forgetting rate")

Theorem (Kivinen et al, 2004):

Let N be any number of seen data, let g be any function (in the RKHS)

then $$\sum_{t=1}^{N} R_{reg,t}(f_t) \leq \left(\sum_{t=1}^{N} R_{reg,t}(g) \right) + a\sqrt{N} + b$$

where a, b are constants depending only on λ, γ, k, ℓ

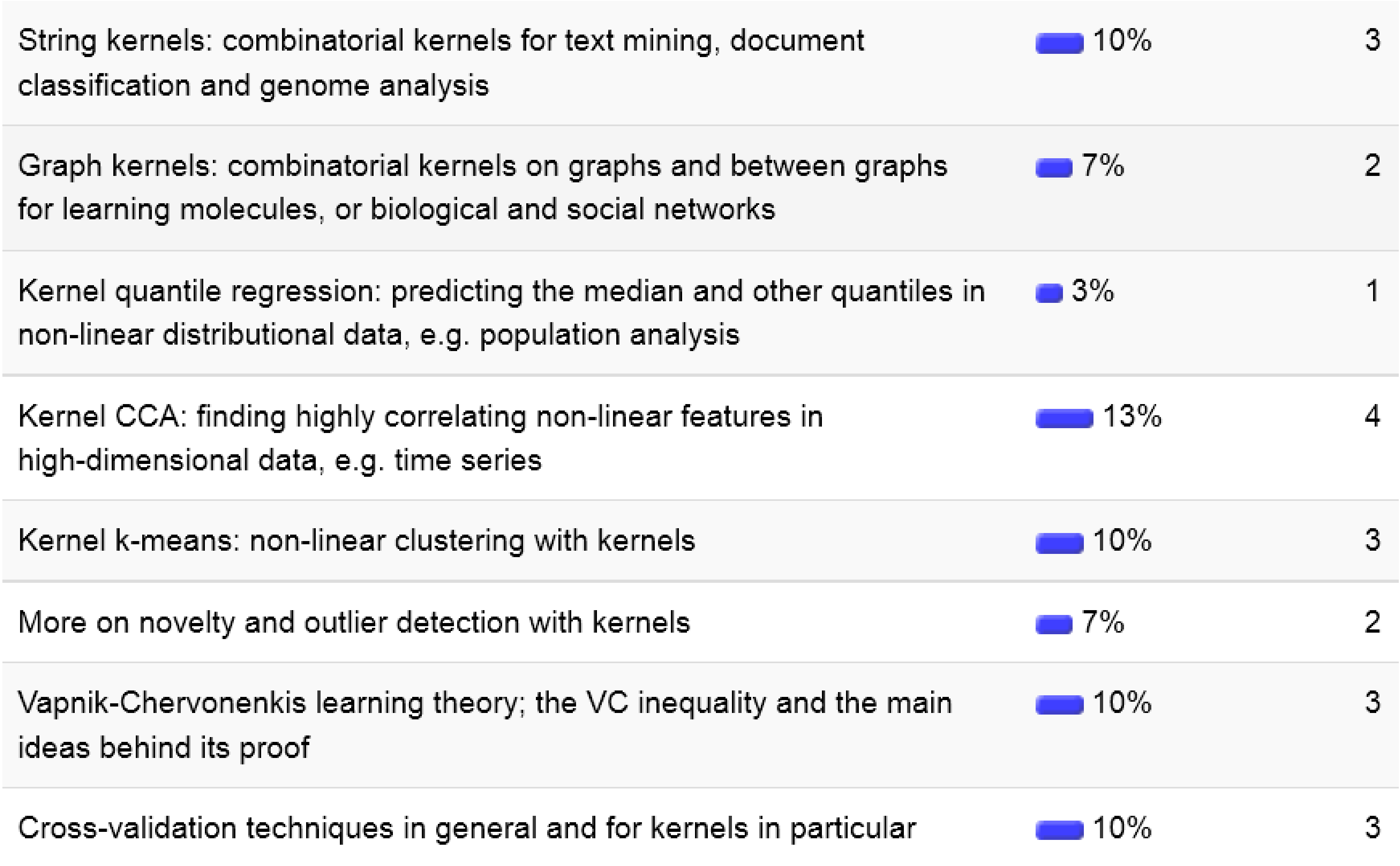
And there is much more…

… but not in kernlab

Some methods are implemented in

scikit, Shogun, LibSVM, etc

Next week: ?



String kernels: combinatorial kernels for text mining, document classification and genome analysis	10%	3
Graph kernels: combinatorial kernels on graphs and between graphs for learning molecules, or biological and social networks	7%	2
Kernel quantile regression: predicting the median and other quantiles in non-linear distributional data, e.g. population analysis	3%	1
Kernel CCA: finding highly correlating non-linear features in high-dimensional data, e.g. time series	13%	4
Kernel k-means: non-linear clustering with kernels	10%	3
More on novelty and outlier detection with kernels	7%	2
Vapnik-Chervonenkis learning theory; the VC inequality and the main ideas behind its proof	10%	3
Cross-validation techniques in general and for kernels in particular	10%	3