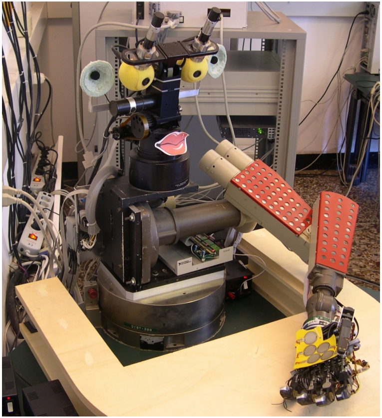
Efficient and Principled Online Classification Algorithms for Lifelong Learning

Francesco Orabona

Toyota Technological Institute at Chicago
Chicago, IL USA

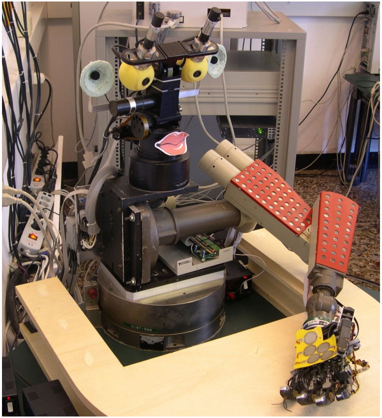
Talk @ Lifelong Learning for Mobile Robotics Applications Workshop, IROS 2012, Vila Moura, Portugal

Something about me



- PhD in Robotics at LIRA-Lab, University of Genova
- PostDocs in Machine Learning

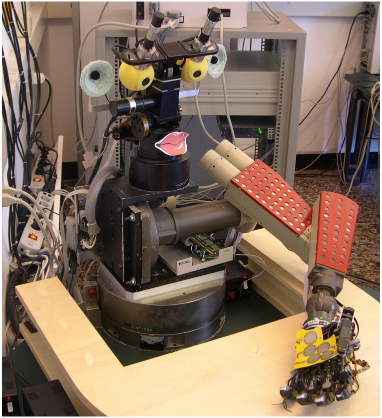
Something about me



- PhD in Robotics at LIRA-Lab, University of Genova
- PostDocs in Machine Learning

I like theoretical motivated algorithms

Something about me



- PhD in Robotics at LIRA-Lab, University of Genova
- PostDocs in Machine Learning

I like theoretical motivated algorithms
But they must work well too! ;-)

Lifelong Learning: why?

In standard Machine Learning we have always two steps

- training on some data
- stop train and use the system on some other data

Lifelong Learning: why?

In standard Machine Learning we have always two steps

- training on some data
- stop train and use the system on some other data

This approach is known to be doomed to fail: environment changes continuously over time and it is impossible to predict how.

Lifelong Learning: how?

Continuous adaptation and learning is the only possibility!

My contributions to the field

- learning with bounded memory on infinite samples (ICML08, JMLR09)
- transfer learning, to bootstrap new classifiers (ICRA09, CVPR10, BMVC12, IEEE trans. Robotics "Soon")
- selective sampling, to know when to ask for more data (ICML09, ICML11)

Lifelong Learning: how?

Continuous adaptation and learning is the only possibility!

My contributions to the field

- learning with bounded memory on infinite samples (ICML08, JMLR09)
- transfer learning, to bootstrap new classifiers (ICRA09, CVPR10, BMVC12, IEEE trans. Robotics "Soon")
- selective sampling, to know when to ask for more data (ICML09, ICML11)

Outline

1. Know that you don't know
2. Selective Sampling
   - Problem definition
   - Coins and hyperplanes
   - BBQ
   - DGS-Mod
3. Experimental Results

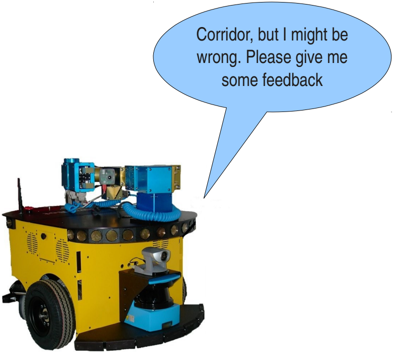
Outline

Know that you don't know

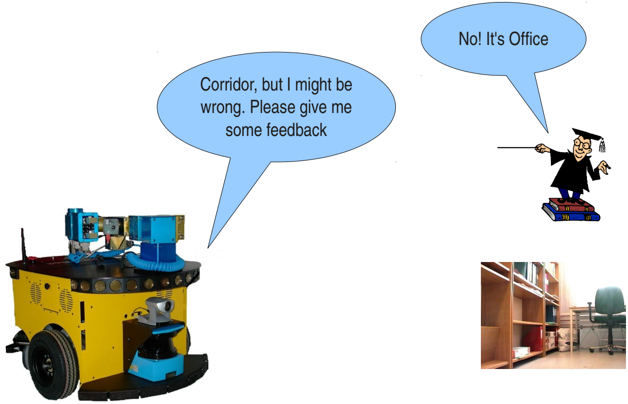
Know that you don't know

Know that you don't know

Know that you don't know



Know that you don’t know



Know that you don't know

Outline

1. Know that you don't know

2. Selective Sampling
   - Problem definition
   - Coins and hyperplanes
   - BBQ
   - DGS-Mod

3. Experimental Results

Selective Sampling

- Selective sampling is a well-known semi-supervised online learning setting (Cohn et al., 1990).
- At each step $t = 1, 2, \ldots$ the learner receives an instance $\boldsymbol{x}_t \in \mathbb{R}^d$ and outputs a binary prediction for the associated unknown label $y_t \in \{0, 1\}$.
- After each prediction the learner may observe the label y_t only by issuing a *query*. If no query is issued at time t, then y_t remains unknown.
- Since one expects the learner's performance to improve if more labels are observed, our goal is to trade off predictive accuracy against number of queries.

Selective Sampling in the Real World

- We want to solve the Selective Sampling problem in the Real World
- This means:
  - Efficient algorithms
  - No unrealistic assumptions
  - No parameters to tune to obtain convergence and/or good performance
  - The order of the samples must be *adversarial*

Warming up: the Coin toss

- Somebody asks us the estimate what is the probability to obtain "head" on a loaded coin

Warming up: the Coin toss

- Somebody asks us the estimate what is the probability to obtain "head" on a loaded coin
- We can toss it N times and estimate the probability as $\frac{\text{number of heads}}{N}$.

Warming up: the Coin toss

- Somebody asks us the estimate what is the probability to obtain "head" on a loaded coin
- We can toss it N times and estimate the probability as $\frac{\text{number of heads}}{N}$.
- How do we choose N?

Warming up: the Coin toss

- Somebody asks us the estimate what is the probability to obtain "head" on a loaded coin
- We can toss it N times and estimate the probability as $\frac{\text{number of heads}}{N}$.
- How do we choose N?
- If we toss the coin at least $\frac{1}{2\epsilon^2} \log \frac{2}{\delta}$ times, then with probability at least $1 - \delta$ the true probability will be in $[\frac{\text{number of heads}}{N} - \epsilon, \frac{\text{number of heads}}{N} + \epsilon]$.

Warming up: the Coin toss

- Somebody asks us the estimate what is the probability to obtain “head” on a loaded coin
- We can toss it N times and estimate the probability as $\frac{\text{number of heads}}{N}$.
- How do we choose N?
- If we toss the coin at least $\frac{1}{2\epsilon^2} \log \frac{2}{\delta}$ times, then with probability at least $1 - \delta$ the true probability will be in $[\frac{\text{number of heads}}{N} - \epsilon, \frac{\text{number of heads}}{N} + \epsilon]$.
- Example: $\epsilon = 0.1$, Probability$= 95\% \Rightarrow\ N \geq 35$

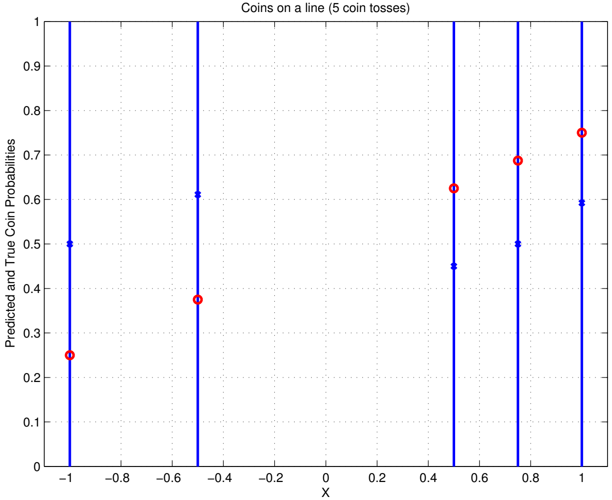
Why do we care about coin toss?

- Suppose there is a probability $P(Y|\boldsymbol{X} = \boldsymbol{x}_t)$
- Each time we receive a sample, the label is given by a coin toss
- Asking for the same label many times, we could estimate it

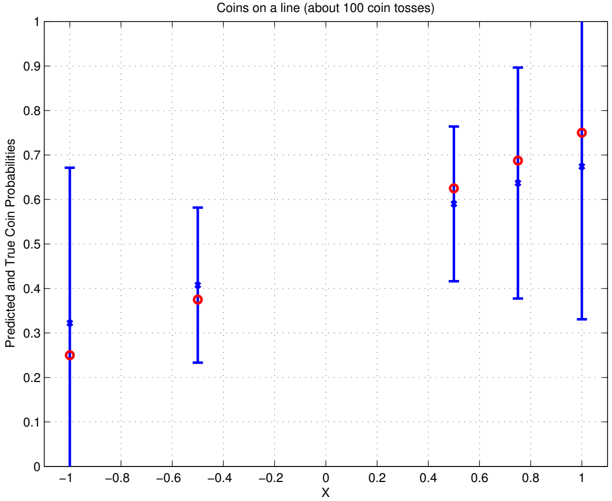
Why do we care about coin toss?

- Suppose there is a probability $P(Y|\boldsymbol{X} = \boldsymbol{x}_t)$
- Each time we receive a sample, the label is given by a coin toss
- Asking for the same label many times, we could estimate it
- Main idea: assume that the function $P(Y|\boldsymbol{X} = \boldsymbol{x})$ has some regularities, so that close points have similar probabilities.
- Gathering points in a neighboorhood will increase the confidence in the estimate of all the points in it!

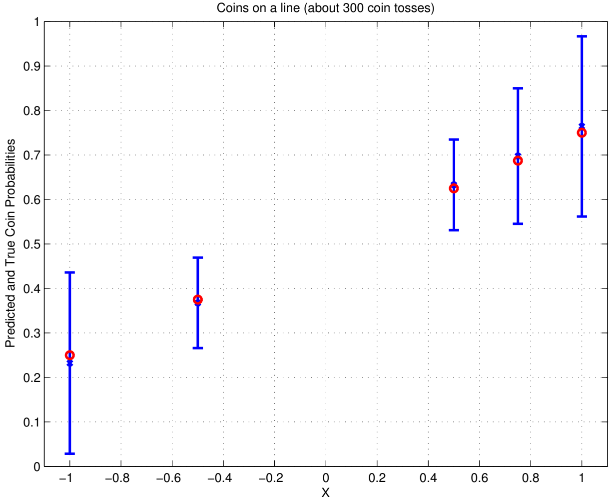
A Step Further: Coins on a Line



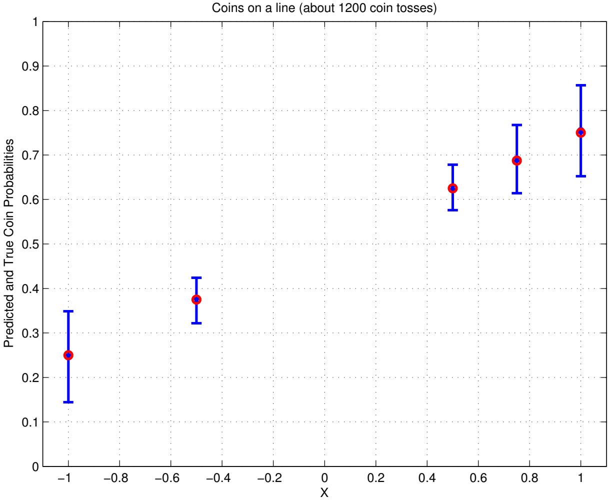
A Step Further: Coins on a Line



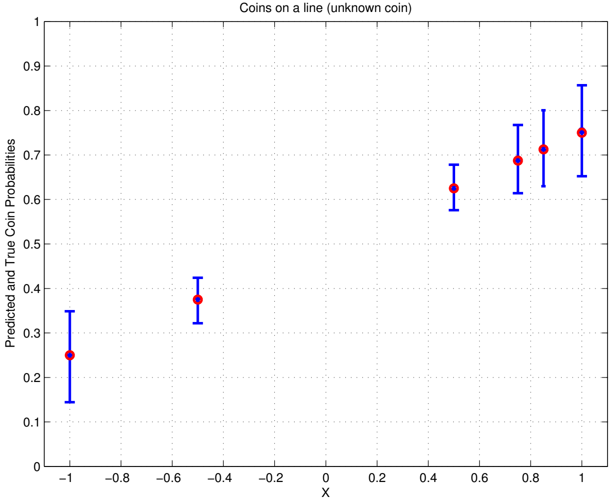
A Step Further: Coins on a Line



A Step Further: Coins on a Line



A Step Further: Coins on a Line



An easy case

- Very easy assumption on $P(Y|\boldsymbol{X} = \boldsymbol{x})$: it is linear in $\boldsymbol{x}$
- Of course this model is a bit restrictive, but...

An easy case

- Very easy assumption on $P(Y|\boldsymbol{X}=\boldsymbol{x})$: it is linear in $\boldsymbol{x}$
- Of course this model is a bit restrictive, but...
- ...we can use the kernel trick!
- $P(Y|\boldsymbol{X}=\boldsymbol{x})=\langle \boldsymbol{u}, \phi(\boldsymbol{x})\rangle$

An easy case

- Very easy assumption on $P(Y|\boldsymbol{X}=\boldsymbol{x})$: it is linear in $\boldsymbol{x}$
- Of course this model is a bit restrictive, but...
- ...we can use the kernel trick!
- $P(Y|\boldsymbol{X}=\boldsymbol{x})=\langle\boldsymbol{u},\phi(\boldsymbol{x})\rangle$
- Is this model powerful enough?

An easy case

- Very easy assumption on $P(Y|\boldsymbol{X}=\boldsymbol{x})$: it is linear in $\boldsymbol{x}$
- Of course this model is a bit restrictive, but...
- ...we can use the kernel trick!
- $P(Y|\boldsymbol{X}=\boldsymbol{x})=\langle\boldsymbol{u},\phi(\boldsymbol{x})\rangle$
- Is this model powerful enough?
- Yes, if the $P(Y|\boldsymbol{X}=\boldsymbol{x})$ is continuos and the kernel *universal* (e.g. Gaussian Kernel)

An easy case

- Very easy assumption on $P(Y|\boldsymbol{X}=\boldsymbol{x})$: it is linear in $\boldsymbol{x}$
- Of course this model is a bit restrictive, but...
- ...we can use the kernel trick!
- $P(Y|\boldsymbol{X}=\boldsymbol{x})=\langle\boldsymbol{u},\phi(\boldsymbol{x})\rangle$
- Is this model powerful enough?
- Yes, if the $P(Y|\boldsymbol{X}=\boldsymbol{x})$ is continuos and the kernel *universal* (e.g. Gaussian Kernel)
- It implies that is possible to approximate $P(Y|\boldsymbol{X}=\boldsymbol{x})$, with $\langle\boldsymbol{u},\phi(\boldsymbol{x}_t)\rangle$

To summarize

- Assume that the outputs y_t comes from a distribution $P(Y|\boldsymbol{X} = \boldsymbol{x})$, continuos w.r.t. $\boldsymbol{x}_t$
- Assume to use a universal kernel, e.g. the gaussian kernel.
- Note that the order of the $\boldsymbol{x}_t$ is *adversarial*, i.e. the samples can arrive in any possible order.

These hypothesis are enough to solve our problem, and the algorithm is even efficient :-)

The Bound on Bias Query (BBQ) Algorithm

Parameters: κ
for $t = 1, 2, \ldots, T$ **do**
  Receive new instance $\boldsymbol{x}_t$
  Predict $\hat{y}_t = \mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_t \rangle)$
  $r = \boldsymbol{x}_t^\top \left(I + A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top \right)^{-1} \boldsymbol{x}_t$
  if $r \geq t^{-\kappa}$ **then**
    Query label y_t
    Update $\boldsymbol{w}$ with a Regularized Least Square
    $A_{t+1} = A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top$
  end if
end for

- Every time a query is not issued the predicted output is not too far from the correct one

N. Cesa-Bianchi, C. Gentile and F. Orabona. Robust Bounds for Classification via Selective Sampling. ICML 2009

The Bound on Bias Query (BBQ) Algorithm

Parameters: κ
for $t = 1, 2, \ldots, T$ **do**
  Receive new instance $\boldsymbol{x}_t$
  Predict $\hat{y}_t = \mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_t \rangle)$
  $r = \boldsymbol{x}_t^\top \left(I + A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top \right)^{-1} \boldsymbol{x}_t$
  if $r \geq t^{-\kappa}$ **then**
    Query label y_t
    Update $\boldsymbol{w}$ with a Regularized Least Square
    $A_{t+1} = A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top$
  end if
end for

- Every time a query is not issued the predicted output is not too far from the correct one

N. Cesa-Bianchi, C. Gentile and F. Orabona. Robust Bounds for Classification via Selective Sampling. ICML 2009

How Does It Work?

- The RLS prediction is an estimate of the true probablity, with a certain bias and variance.
- BBQ issues a query when a r is small, beacause r is proportional to a common upper bound on bias and variance of the current RLS estimate.
- Hence, when r is small the learner can safely avoid issuing a query on that step, because it knows it will be close enough to the true probability.

Theoretical Analysis

- How do we measure the performance of this algorithm?
- We use a relative comparison: how good is it w.r.t. the best classifier ever?
- The best classifier is the one that is trained with the knowledge of all the samples, and same kernel.
- We define the cumulative regret after T steps
  $R_T = \sum_{t=1}^{T} \left(\mathbb{P}(Y_t \,\widehat{\Delta}_t < 0) - \mathbb{P}(Y_t \,\Delta_t < 0) \right)$

Regret Bound of BBQ (2011 Version)

Theorem

If BBQ is run with input $0 < \kappa < 1$ *then*
– the number of queried labels is $N_T \leq T^{\kappa} \log |A_T|$
– its cumulative regret R_T*, with probability at least* $1 - \delta$*, is less than*

$$\min_{0<\varepsilon<1} \varepsilon\, T_\varepsilon + \mathcal{O}\left(\frac{\|\boldsymbol{u}\|^2}{\varepsilon} \log \frac{N_T}{\delta} \log |A_T|\right) + \mathcal{O}\left(\left(\frac{\|\boldsymbol{u}\|}{\varepsilon}\right)^{2/\kappa}\right)$$

where $T_\varepsilon = \left|\{1 \leq t \leq T \,:\, |\Delta_t| < \varepsilon\}\right|$, $A_T = \sum_{queries} \boldsymbol{x}_t \boldsymbol{x}_t^\top$.

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

Regret Bound of BBQ (2011 Version)

Theorem

If BBQ is run with input $0 < \kappa < 1$ *then*
– the number of queried labels is $N_T \leq T^{\kappa} \log |A_T|$
– its cumulative regret R_T*, with probability at least* $1 - \delta$*, is less than*

$$\min_{0<\varepsilon<1} \varepsilon\, T_\varepsilon + \mathcal{O}\left(\frac{\|\boldsymbol{u}\|^2}{\varepsilon} \log \frac{N_T}{\delta} \log |A_T|\right) + \mathcal{O}\left(\left(\frac{\|\boldsymbol{u}\|}{\varepsilon}\right)^{2/\kappa}\right)$$

where $T_\varepsilon = \left|\{1 \leq t \leq T \,:\, |\Delta_t| < \varepsilon\}\right|$, $A_T = \sum_{queries} \boldsymbol{x}_t \boldsymbol{x}_t^\top$.

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

Regret Bound of BBQ (2011 Version)

Theorem

If BBQ is run with input $0 < \kappa < 1$ *then*
– the number of queried labels is $N_T \leq T^\kappa \log |A_T|$
– its cumulative regret R_T*, with probability at least* $1 - \delta$*, is less than*

$$\min_{0<\varepsilon<1} \varepsilon\, T_\varepsilon + \mathcal{O}\left(\frac{\|\boldsymbol{u}\|^2}{\varepsilon} \log \frac{N_T}{\delta} \log |A_T|\right) + \mathcal{O}\left(\left(\frac{\|\boldsymbol{u}\|}{\varepsilon}\right)^{2/\kappa}\right)$$

where $T_\varepsilon = \left|\{1 \leq t \leq T \,:\, |\Delta_t| < \varepsilon\}\right|$, $A_T = \sum_{queries} \boldsymbol{x}_t \boldsymbol{x}_t^\top$.

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

Regret Bound of BBQ (2011 Version)

Theorem

If BBQ is run with input $0 < \kappa < 1$ then
– the number of queried labels is $N_T \leq T^\kappa \log |A_T|$
– its cumulative regret R_T, with probability at least $1 - \delta$, is less than

$$\min_{0<\varepsilon<1} \text{Regret classifier that asks all the labels} + \mathcal{O}\left(\left(\frac{\|\boldsymbol{u}\|}{\varepsilon}\right)^{2/\kappa}\right)$$

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

Regret Bound of BBQ (2011 Version)

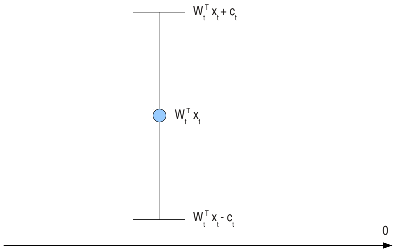
Theorem

If BBQ is run with input $0 < \kappa < 1$ *then*
– the number of queried labels is $N_T \leq T^\kappa \log |A_T|$
– its cumulative regret R_T*, with probability at least* $1 - \delta$*, is less than*

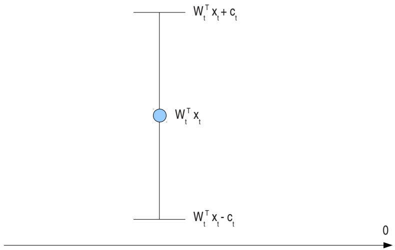
$$\min_{0<\varepsilon<1} \textit{Regret classifier that asks all the labels} + \mathcal{O}\left(\left(\frac{\|\boldsymbol{u}\|}{\varepsilon}\right)^{2/\kappa}\right)$$

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

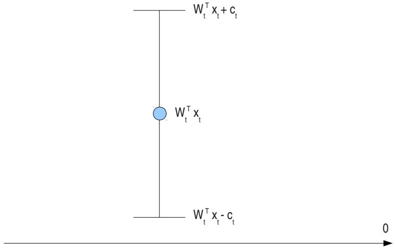
- r does not depend on the labels, hence the query condition ignores all the labels.
- Probably we are loosing precious information.



- r does not depend on the labels, hence the query condition ignores all the labels.
- Probably we are loosing precious information.
- Why the margin is important?



- r does not depend on the labels, hence the query condition ignores all the labels.
- Probably we are loosing precious information.
- Why the margin is important?
- Suppose $\boldsymbol{w}^\top \boldsymbol{x}_t > c_t$, we know that $\boldsymbol{u}^\top \boldsymbol{x}_t \geq \boldsymbol{w}^\top \boldsymbol{x}_t - c_t > 0$



DGS-Mod Algorithm

Parameter: $\alpha > 0, 0 < \delta < 1$
Initialization: Vector $\boldsymbol{w} = \mathbf{0}$, matrix $A_1 = I$
for each time step $t = 1, 2, \dots$ **do**
  Observe instance $\boldsymbol{x}_t$ and set $\widehat{\Delta}_t = \boldsymbol{w}^\top \boldsymbol{x}_t$
  Predict label with $\textsc{sgn}(\hat{\Delta}_t)$
  if $\hat{\Delta}_t^2 \leq 2\alpha\left(\boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t\right)\left(4 \sum_{s=1}^{t-1} Z_s r_s + 36 \ln(t/\delta)\right) \log t$ **then**
    Query label y_t
    $\boldsymbol{w} = \boldsymbol{w} - \textsc{sgn}(\widehat{\Delta}_t) \left| \frac{|\hat{\Delta}_t| - 1}{\boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t} \right|_+ A_t^{-1} \boldsymbol{x}_t$
    $A_{t+1} = A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top \quad \text{and} \quad r_t = \boldsymbol{x}_t^\top A_{t+1}^{-1} \boldsymbol{x}_t$
    Update $\boldsymbol{w}$ with a Regularized Least Square
  else
    $A_{t+1} = A_t, \quad r_t = 0$
  end if
end for

Kernels can be used, formulating the algorithm in dual variables.

DGS-Mod Algorithm

Parameter: $\alpha > 0, 0 < \delta < 1$
Initialization: Vector $\boldsymbol{w} = \mathbf{0}$, matrix $A_1 = I$
for each time step $t = 1, 2, \ldots$ **do**
  Observe instance $\boldsymbol{x}_t$ and set $\widehat{\Delta}_t = \boldsymbol{w}^\top \boldsymbol{x}_t$
  Predict label with $\mathrm{SGN}(\hat{\Delta}_t)$
  if $\hat{\Delta}_t^2 \leq 2\alpha\left(\boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t\right)(4 \sum_{s=1}^{t-1} Z_s r_s + 36 \ln(t/\delta)) \log t$ **then**
    Query label y_t
    $\boldsymbol{w} = \boldsymbol{w} - \mathrm{SGN}(\widehat{\Delta}_t) \left| \frac{|\hat{\Delta}_t| - 1}{\boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t} \right|_+ A_t^{-1} \boldsymbol{x}_t$
    $A_{t+1} = A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top$ and $r_t = \boldsymbol{x}_t^\top A_{t+1}^{-1} \boldsymbol{x}_t$
    Update $\boldsymbol{w}$ with a Regularized Least Square
  else
    $A_{t+1} = A_t, \quad r_t = 0$
  end if
end for

Kernels can be used, formulating the algorithm in dual variables.

DGS-Mod Guarantees

Theorem

After any number T of steps, with probability at least $1-\delta$, the cumulative regret R_T of DGS-Mod with $\alpha > 0$ is less than

$$\min_{0<\varepsilon<1}\left(\varepsilon\, T_\varepsilon + \mathcal{O}\left(\frac{\|\boldsymbol{u}\|^2 + \ln|A_{T+1}| + \ln\frac{T}{\delta}}{\varepsilon} + e^{\frac{\|\boldsymbol{u}\|^2+1}{\alpha}}\right)\right)$$

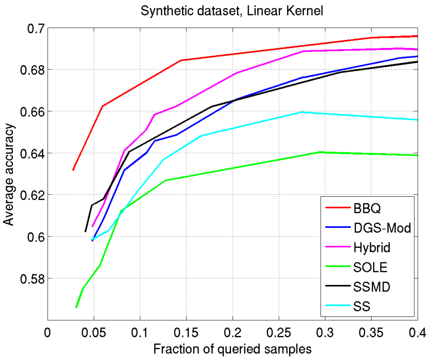
and the number N_T of queries is less than

$$\min_{0<\varepsilon<1}\left(T_\varepsilon + \mathcal{O}\left(\frac{\ln|A_{T+1}|\left[\|\boldsymbol{u}\|^2 + (1+\alpha\ln T)\left(\ln|A_{T+1}| + \ln\frac{T}{\delta}\right)\right]}{\varepsilon^2}\right)\right).$$

F. Orabona, N. Cesa-Bianchi. Better Algorithms for Selective Sampling. ICML 2011

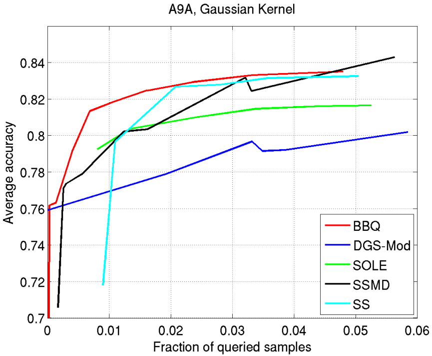
Outline

Synthetic Experiment



- 10,000 random examples on the unit circle in $\mathbb{R}^{100}$, $\|\boldsymbol{x}_t\|_\infty \le 1$.
- The labels were generated according to our noise model.
- BBQ has the best trade-off between performance and queries.

Real World Experiments



- Adult dataset with 32000 samples and gaussian kernel.

Summary

- Real world problems often deviate from the standard "training/test IID data"
- Theory can (and should?) be used to model the real world problems with real world hypothesis
- In particular
    - Continuity of the marginal distribution and universal kernels allow us to design efficient and principled algorithms to solve the selective sampling problem.

Summary

- Real world problems often deviate from the standard "training/test IID data"
- Theory can (and should?) be used to model the real world problems with real world hypothesis
- In particular
  - Continuity of the marginal distribution and universal kernels allow us to design efficient and principled algorithms to solve the selective sampling problem.
  - What problems can *you* solve with these tools? :-)

Thanks for your attention

Code: `http://dogma.sourceforge.net`
My website: `http://francesco.orabona.com`

Minimal Bibliography:

Cesa-Bianchi, Gentile, Orabona. ICML'09
Dekel, Gentile, Sridharan. COLT'10
Orabona, Cesa-Bianchi. ICML'11