time-limited support vector machines model selection
margins

time-limited svms
first problem:
how to solve the SVM (=maximize margin) problem?
first problem: how to solve the SVM (=maximize margin) problem?

this is a quadratic minimization problem which in its most basic form looks like

\[ \min_{w,b} \frac{1}{2}||w||^2 \]

such that \( y_i(<w,x_i>-b) \geq 1 \).
\[ \varphi : (x, y) \mapsto (x, y, \sqrt{x^2 + y^2}) \]
second problem: how to choose the kernel?
second problem: how to choose the kernel?

we fix here the universal ’RBF’ kernel which depends on a parameter $\gamma > 0$

so choosing a kernel means choosing a ’good’ $\gamma$. 

noise
noise
third problem: how to choose the 'trust' $C$ for the data?

(low $C$ = assume data is noisy
high $C$ = assume data is correct)
solving the svm problem

there are different approaches for solving the optimization problem:

· stochastic gradient descent (BSGD)
· cutting planes (SVMperf)
· sequential minimal optimization (LASVM/LIBSVM)
· (minimal) enclosing ball (CVM/BVM)

...
solving the svm problem

there are different approaches for solving the optimization problem:

- stochastic gradient descent (BSGD)
- cutting planes (SVMperf)
- sequential minimal optimization (LASVM/LIBSVM)
  - (minimal) enclosing ball (CVM/BVM)

each solver has its own virtues and problems.
choosing the right \((C, \gamma)\) is called model selection.
choosing the right \((C, \gamma)\) is called model selection.

the quality of the solution depends heavily on this choice!
choosing the right \((C, \gamma)\) is called model selection.

the quality of the solution
depends \textit{heavily} on this choice!

\textit{but: no }’theory’ on how to choose \(C\) or \(\gamma\).
grid search

easiest way out: use a systematic 'grid'-search—i.e. simply solve svm problem for all combinations of say

\[ C = \{2^{-15}, \ldots, 2^{15}\} \text{ and } \gamma = \{2^{-15}, \ldots, 2^{15}\}. \]
grid search

easiest way out: use a systematic ’grid’-search—i.e. simply solve svm problem for all combinations of say

\[ C = \{2^{-15}, \ldots, 2^{15}\} \quad \text{and} \quad \gamma = \{2^{-15}, \ldots, 2^{15}\}. \]

thus, one needs to solve the problem 961 times.

to know which one is the best, validate the model on a validation set (or use cross-validation).
but solving a single SVM can even take days–
suppose you are given a data set with 1,000,000 points
and solving an SVM takes \( \approx 4 \) hours.
then grid search takes
\[ 961 \times 4 \text{ hours} \approx 160 \text{ days}(!). \]
grid search

even worse: information get ’lost’ as from these 961 models you only take the best one– so 99.89% of all computed solutions are thrown away.
even worse: information get ’lost’
as from these 961 models you only take the best one– so
99.89% of all computed solutions are thrown away.

can we do better?
improving grid search

yes!
improving grid search

yes!

a) apply a more sophisticated search!
improving grid search

yes!

a) apply a more sophisticated search!

b) subsample your data!
improving grid search

yes!

a) apply a more sophisticated search!

b) subsample your data!

c) do not throw away models, but create an ensemble!
improving grid search

yes!

a) apply a more sophisticated search!
b) subsample your data!
c) do not throw away models, but create an ensemble!
d) be realistic!
being realistic

what do you do if...

... you're in a soccer stadium and...

after 3 minutes your favourite team (of course fc oberpusemuckelhausen) is behind, 0:5 against fc barcelona.

do you really stay there and watch till the end?

you really expect fc barcelona to loose? probably not. you switch off the tv or go home.
being realistic

what do you do if...

... you’re in a soccer stadium and...
being realistic

what do you do if...

... you’re in a soccer stadium and...

after 3 minutes your favourite team
(of course fc oberpusemuckelhausen)
is behind, 0:5 against...
being realistic

what do you do if...
... you’re in a soccer stadium and...
after 3 minutes your favourite team
(of course fc oberpusementuckelhausen)
is behind, 0:5 against...
fc barcelon.
being realistic

what do you do if...

... you’re in a soccer stadium and...

after 3 minutes your favourite team
(of course fc oberpusemuckelhausen)
is behind, 0:5 against...

fc barcelona.

do you really stay there and watch till the end?
you really expect fc barcelona to loose?
being realistic

what do you do if...

... you’re in a soccer stadium and...

after 3 minutes your favourite team
(of course fc oberpusemuckelhausen)
is behind, 0:5 against...

fc barcelona.

do you really stay there and watch till the end?
you really expect fc barcelona to loose?

probably not. you switch off the tv or go home.
being realistic

can we apply this to model selection?
being realistic

can we apply this to model selection?

why?
being realistic

can we apply this to model selection?

why?

main insight: good models train faster than bad ones.
can we apply this to model selection?

why?
main insight: good models train faster than bad ones.

how?
can we apply this to model selection?

why?
main insight: good models train faster than bad ones.

how?
simply stop training after say 60 seconds
better models will have fewer errors on the validation set.
can we apply this to model selection?

why?

main insight: good models train faster than bad ones.

how?

simply stop training after say 60 seconds
better models will have fewer errors on the validation set.

simple idea, could have tremendous effect:
model search would be an order of magnitude faster.
(160 days $\rightarrow$ 20 hours)
test this idea:

- try six different solvers.
- heuristic time limit $T = 2 \log_{10}(n) + 1$.
- use EGO instead of grid search.
  thus at each point train with $T$ seconds on the train set.
  retrain a full model on $(C, \gamma)$ with the lowest validation error.
test this idea:

- try six different solvers.
experiment

test this idea:

- try six different solvers.
- heuristic time limit $T = 2^{\log_{10}(n)+1}$. 
test this idea:

- try six different solvers.
- heuristic time limit $T = 2^{\log_{10}(n)+1}$.
- use EGO instead of grid search.
test this idea:

- try six different solvers.
- heuristic time limit $T = 2^{\log_{10}(n)+1}$.
- use EGO instead of grid search.

thus at each point train with $T$ seconds on the train set.
test this idea:

- try six different solvers.
- heuristic time limit $T = 2^{\log_{10}(n)+1}$.
- use EGO instead of grid search.

thus at each point train with $T$ seconds on the train set.
retrain a full model on $(C, \gamma)$
with the lowest validation error.
results: accuracy

- BSGD
- LASVM
- LIBSVM
- CVM
- BVM
- SVMperf

Absolute Error

- aXa
- cod−rna
- mnist
- poker

- time-limited svms

23/26
results: timing

time-limited svms
for LASVM we can use a time-limited model selection—this is not true for the other solvers.
for LASVM we can use a time-limited model selection—this is not true for the other solvers.

the approach looses a bit accuracy but is over an order of magnitude faster.
thank you.