Tricks of the trade for training SVMs

Gökhan H. Bakır & Olivier Chapelle

Max Planck Institute for Biological Cybernetics

Tübingen, Germany

{goekhan.bakir,olivier.chapelle}@tuebingen.mpg.de

Overview

- Problem Statement
- Tricks in the primal
- Tricks in the dual
- Conjecture

Why are we sitting here?

SVM can not watch TV.

– Leon Bottou –

Problem Statement

Why?

In kernel methods the representer theorem is your **best** friend and your **worst** enemy.

Theorem 1 (Representer Theorem). *Under some mild conditions ($f \in$ RKHS,...) the solution f^* of the problem*

$$f^* = \arg\min_f \sum_{i=1}^{N} \ell(y_i, x_i, f(x_i)) + \Omega[f] \tag{1}$$

has the form:

$$f^*(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x). \tag{2}$$

See (Kimeldorf and Wahba, 1971; Schölkopf et al., 2000)

Remark 2. *Does not imply uniqueness of α.*

Overview

- Problem Statement
- Tricks in the primal
- Tricks in the dual
- Conjecture

Linear SVM in the primal

We want to minimize (with an L_2 penalization of the errors)

$$\mathbf{w}^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))^2.$$

$\longrightarrow$ Forget about dual problem, Lagrange multipliers, Kuhn-Tucker conditions, ...

$\longrightarrow$ Just minimize it directly; for instance Newton steps:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - (H^t)^{-1} \nabla^t$$

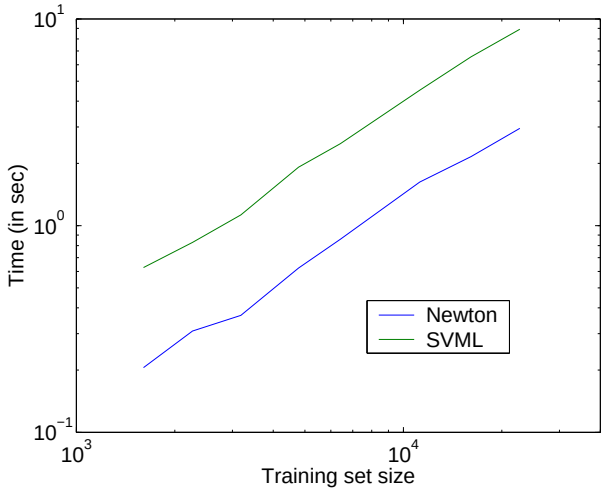
$$\begin{array}{lrll} \text{Gradient:} & \nabla_p^t = & 2w_p & -2C \sum_{i=1}^{n} y_i x_{ip} \max(0, 1 - y_i(\mathbf{w}^t \cdot \mathbf{x}_i + b^t)) \\ \text{Hessian:} & H_{pq}^t = & 2 & +2C \sum_{i=1}^{n} x_{ip} x_{iq} 1_{y_i(\mathbf{w}^t \cdot \mathbf{x}_i + b^t) \leq 1} \end{array}$$

Very simple to implement and efficient: complexity is $O(nd^2)$ per step and usually convergence is reached after only couple of steps.

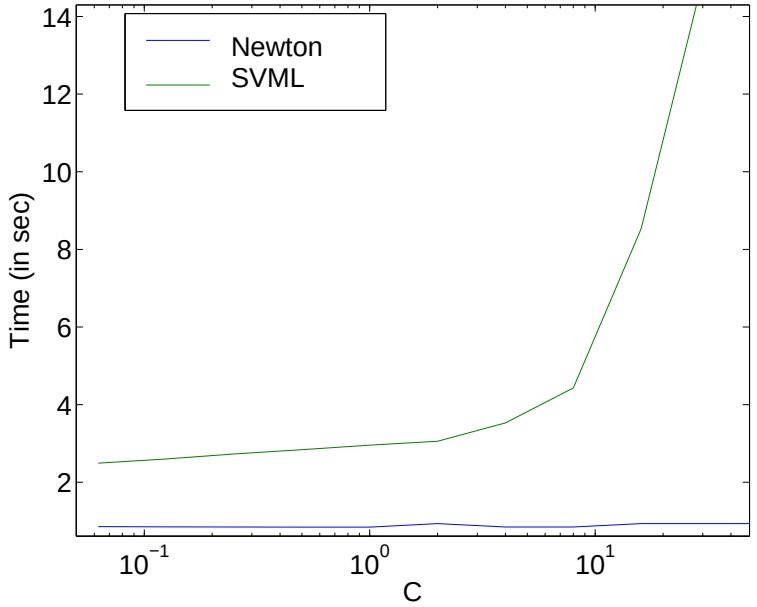
Experimental comparison with LSVM [Mangasarian '00] – fixpoint based approach

- LSVM has been specially designed to train linear SVMs

Adult dataset, stop when precision on the obj fun $< 10^{-7}$.



Fixed $C = 0.05$ (as in original paper).
Complexity is linear for both methods.
Better constant for Newton.



6414 points.
Convergence problems with LSVM
for large values of C

No need of duality theory !

Simpler is better : with a linear SVM, when the number of points is larger than the dimension, straightforward minimization is the best !

Non-linear SVMs

- From the representer theorem, we know that optimal solution is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i k(\mathbf{x}, \mathbf{x}_i) + b$$

- Minimize directly over $\boldsymbol{\beta} \in \mathbb{R}^n, b \in \mathbb{R}$,

$$\underbrace{\boldsymbol{\beta}^\top K \boldsymbol{\beta}}_{\text{regularizer}} + C \underbrace{\sum_{i=1}^{n} \max(0, 1 - y_i([K\boldsymbol{\beta}]_i + b))^2}_{\text{loss}}$$

$\longrightarrow$ Again, no need of duality theory.

Complexity

- Let n_{SV} be the number of support vectors, i.e. the points for which the gradient of the loss is non zero.
- Using the Woodbury formula to invert the Hessian, the complexity of performing one Newton step on the $\boldsymbol{\beta}$ is $O(n_{\mathrm{SV}}(n + {n_{\mathrm{SV}}}^2))$

  $\longrightarrow$ Not surprisingly, exactly the same as standard SVM solvers.
- Can be large if n_{SV} is large.

Trick

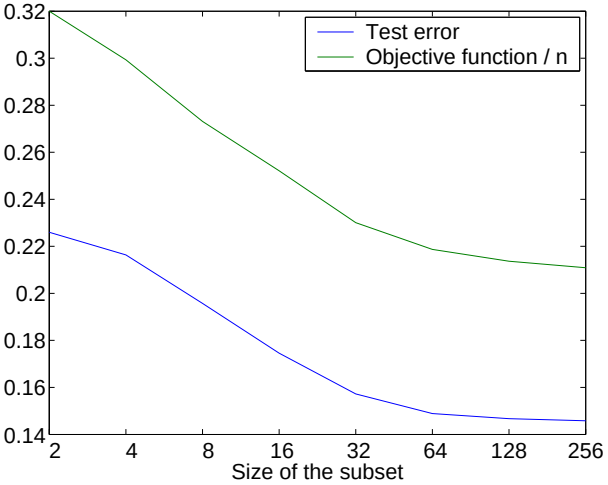
- Fix the complexity (both in the learning and computational sense) by choosing a subset of points on which to expand the solution.
- Let $S \subset [1...n]$. Minimize

$$\boldsymbol{\beta}_S^\top K_{SS} \boldsymbol{\beta}_S + C \sum_{i=1}^{n} \max(0, 1 - y_i (K_{i,S} \boldsymbol{\beta}_S + b))^2.$$

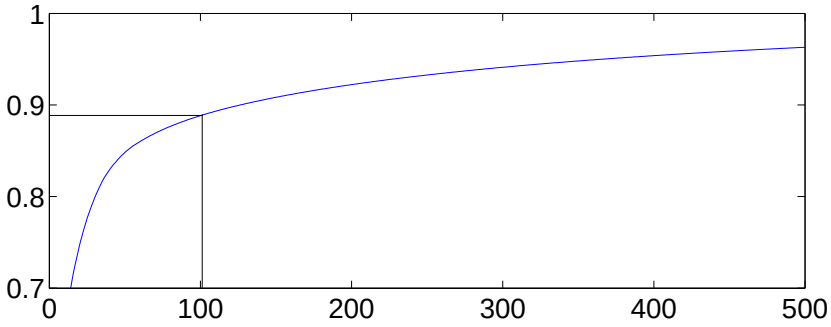
- If $|S| = k$, the complexity is $O(k^2 n)$.
- Very similar to RSVM [Mangasarian '00].

- In feature space, this is equivalent to train a linear SVM after projection of points on the subspace spanned by $\{\Phi(\mathbf{x}_i)\}_{i \in S}$.
- If the "effective dimension" of the feature space (or the effective rank of the Gram matrix) is around $k \longrightarrow$ not a big loss in approximation.
- If it's larger, loss in accuracy, but computational speed-up.
- Good for data mining: n is very large and one does not want to find the best solution, but one within a fixed amount of time.

Experimental results on the Adult dataset, RBF kernel.

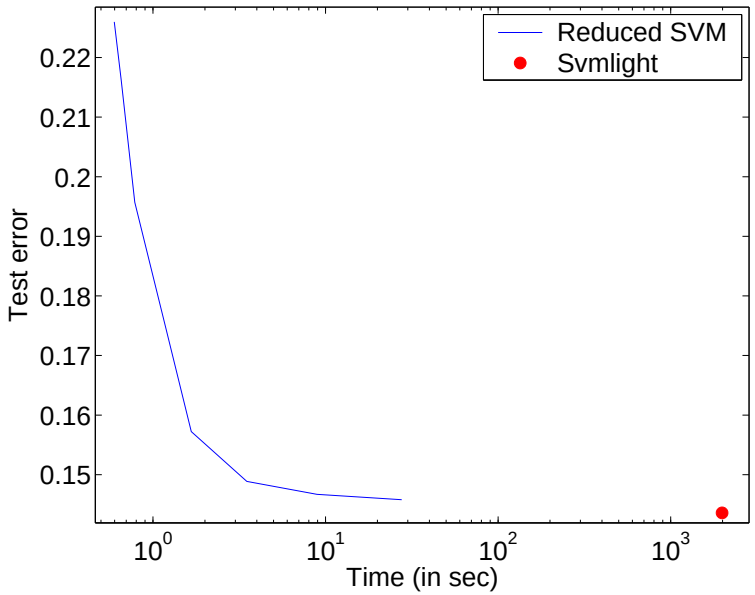


22697 points; results averaged over 10 repetitions
Having more than 100 basis function does not help a lot (note that the dimension of the data is 123).



The first 100 eigenvalues of the kernel matrix retain 89% of the spectrum.

Comparison with standard SVM learning: Can reduce the training time of several orders of magnitude, without too much loss in accuracy.



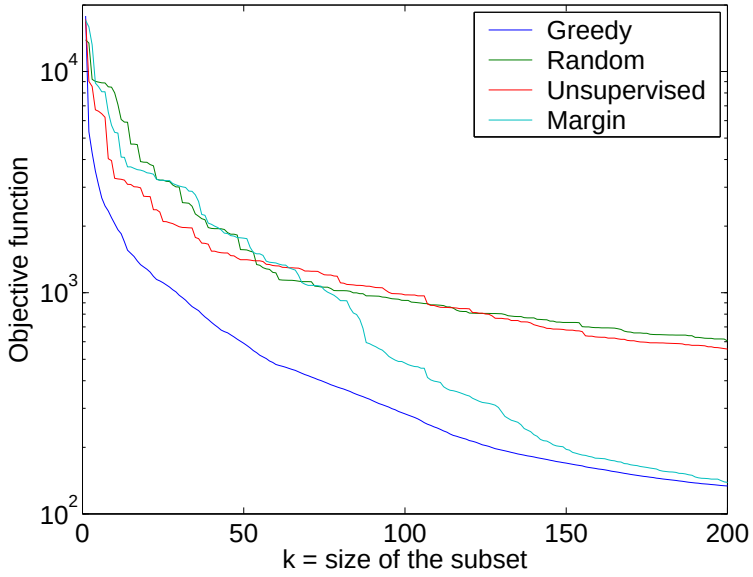
Remarks

- Objective function and test error are strongly related
  $\longrightarrow$ we are in an *underfitting* situation
- The Adult dataset is quite noisy $\longrightarrow$ no need for an accurate solution (and S can be small).
- For some other datasets (e.g. Mnist), need for accuracy.

Greedy choice of the subset

- Idea = find the subset of size k such that the objective function computed with this subset is small as possible.
- First attempt: given the current subset, the corresponding solution and a candidate to be added in the subset, perform a virtual Newton step and see how much the objective function would decrease.
- Select the best candidate among of a pool of p of them and iterate. Complexity is $O(nkp)$ per iteration.

Experiments on Mnist dataset, polynomial kernel of degree 5, 10 000 points, class 0 vs the rest.



At each iteration, the objective function is minimized and a new point is selected to be in the subset according to

`Greedy` the method explained above

`Random`

`Unsupervised` the maximum distance in feature space from the linear subspace spanned by the the current subset.

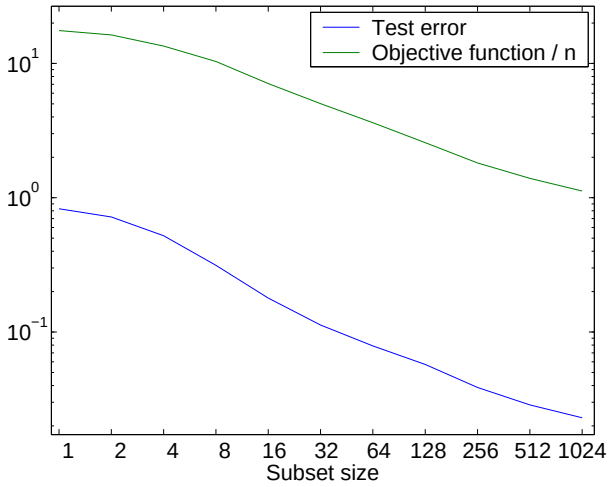
`Margin` the minimum distance from the decision boundary.

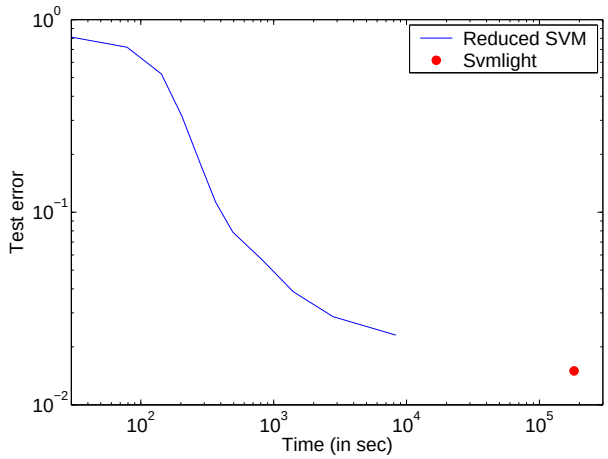
- Speed-up: discarding the points which are not support vectors, (i.e. $y_i(f(\mathbf{x}_i) + b) \geq 1$), the algorithm is the same as kernel ridge regression.
  $\longrightarrow$ A fast rank one update is possible to recompute the solution after a basis point has been added.
- Final algorithm:

  for $i = 1$ to k **do**
    if i is a power of 2 **then**
      Train the SVM on the current subset
    end if
    Choose a random set of p candidates
    Select the one which decreases the most the objective function
    Update the solution (assuming the set of SV does not change)
  end for
- Overall complexity is $O(k^2np))$

Multiclass experiment

- Mnist, full training set (60k points), one vs the rest.
- Use the same subset expansion for the 10 classifiers.
- Points in the subset are chosen such that the sum of the objective function is minimized.





Take home message:

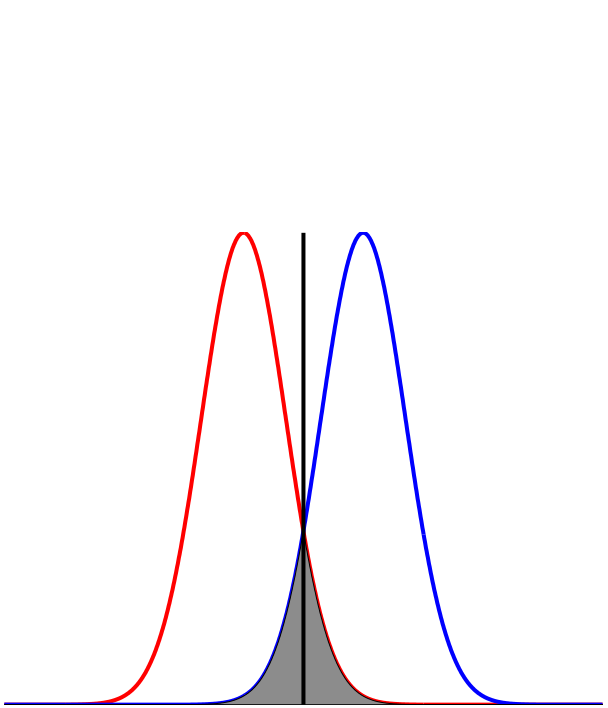
This enables the user to choose the accuracy / time complexity trade-off himself.

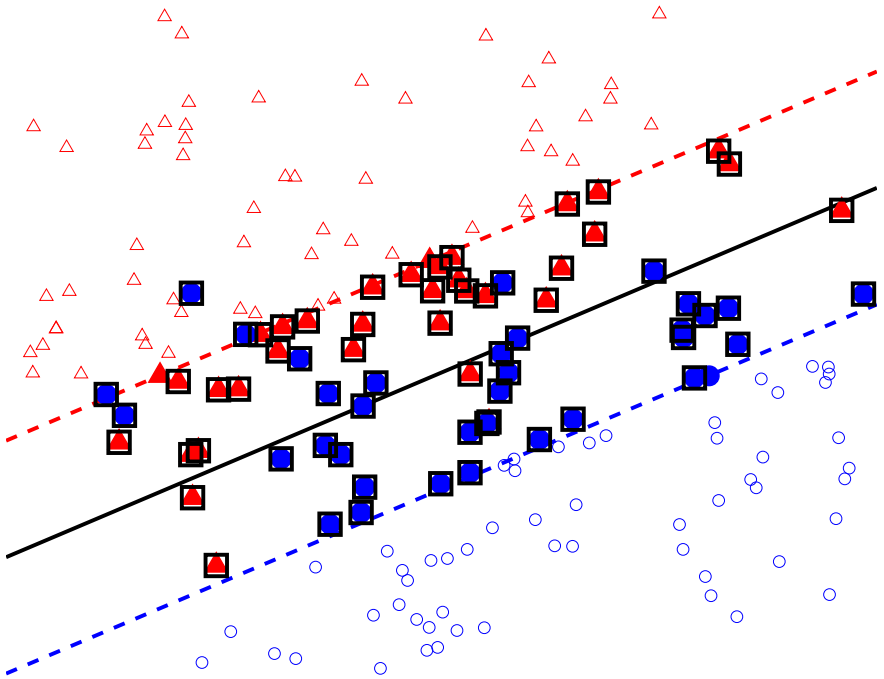
Overview

- Problem Statement
- Tricks in the primal
- Tricks in the dual
- Conjectures

Where do we spend our money in the dual formulation?

Consider a problem with noise.





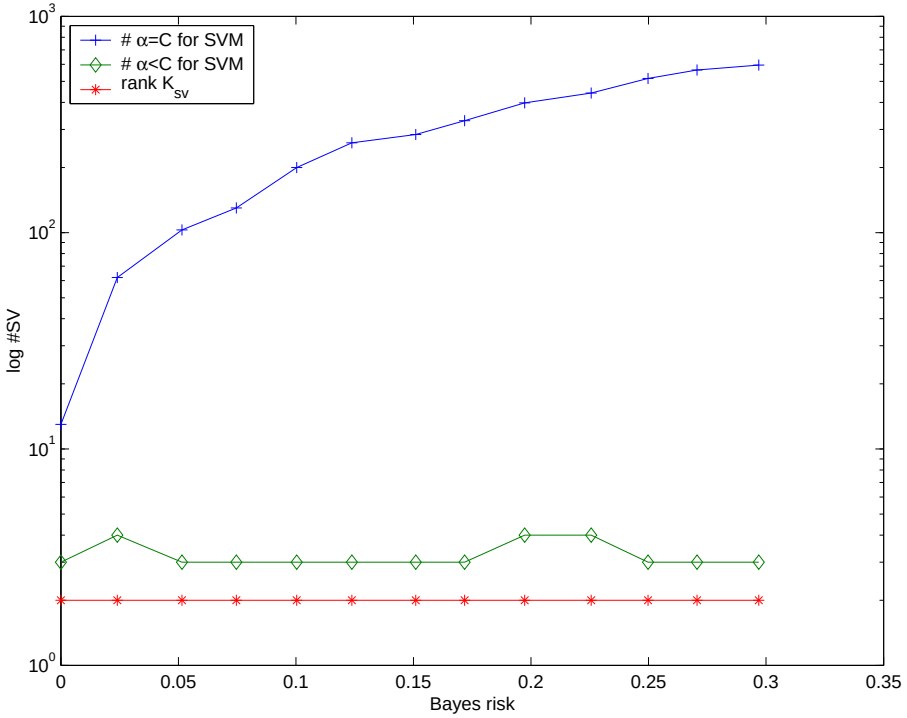
$\Rightarrow$ All outliers and points in overlapping class regions will end up as support vector.

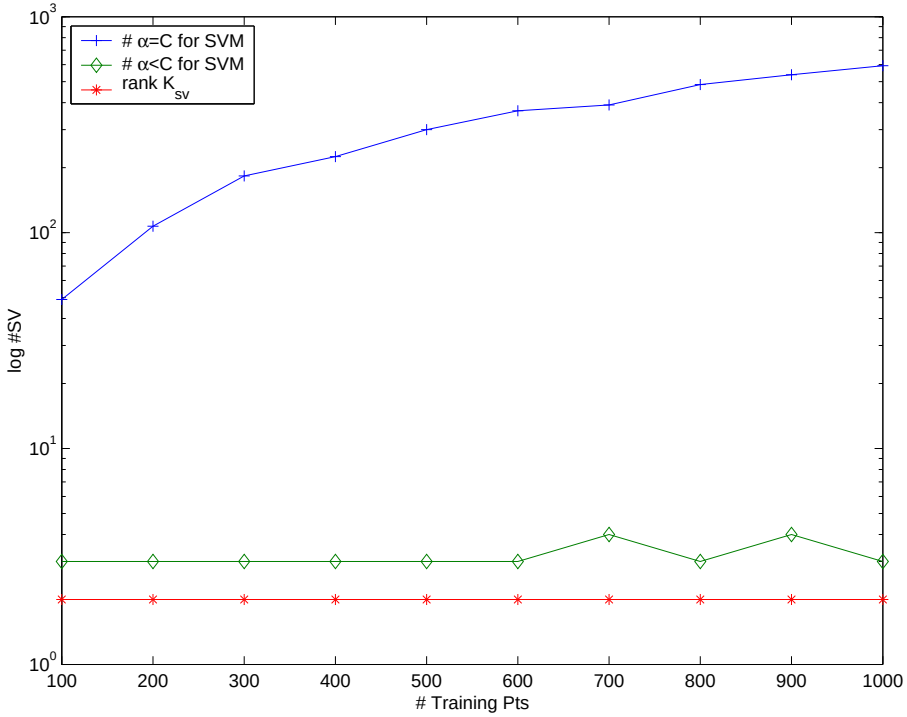
What are the problems if n increases?

Steinwart (Steinwart, 2004) showed that k – the number of SVs increases **linearly** with the number n of training examples. More specifically,

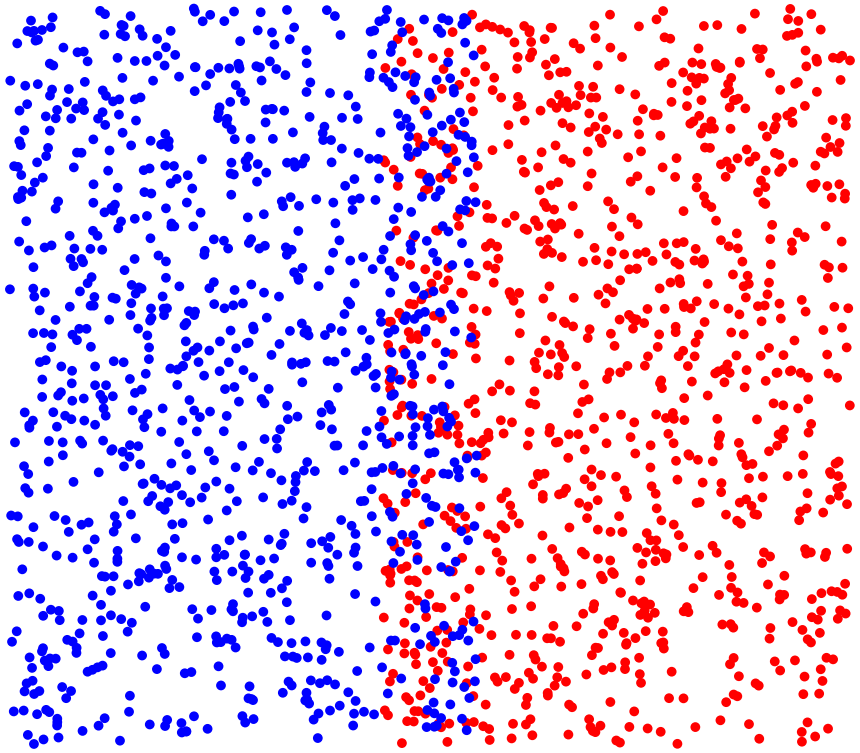
$$k/n \longrightarrow 2\mathcal{B}_K \tag{3}$$

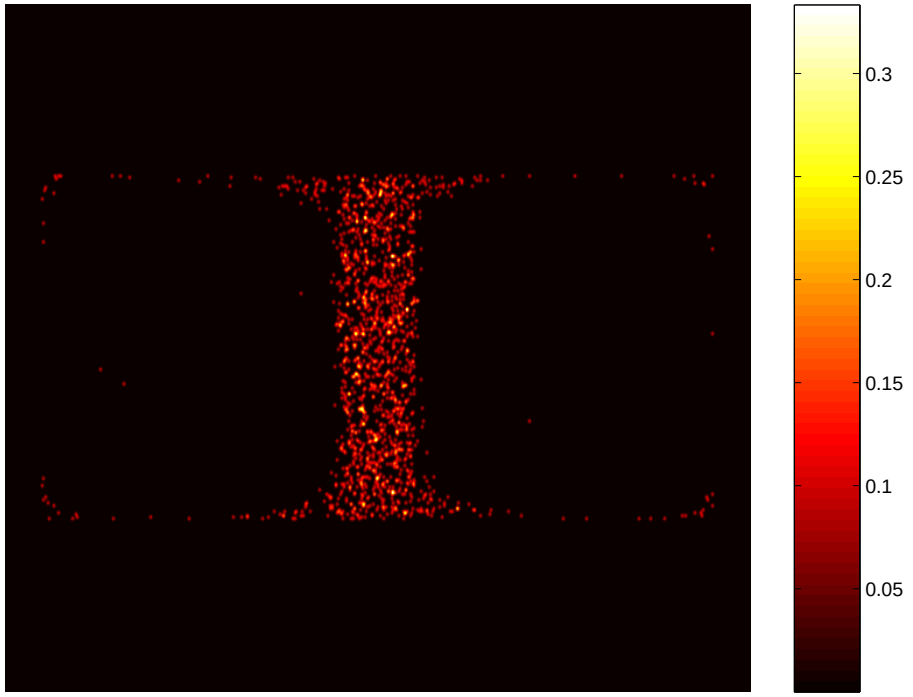
where $\mathcal{B}_K$ is the bayes risk.

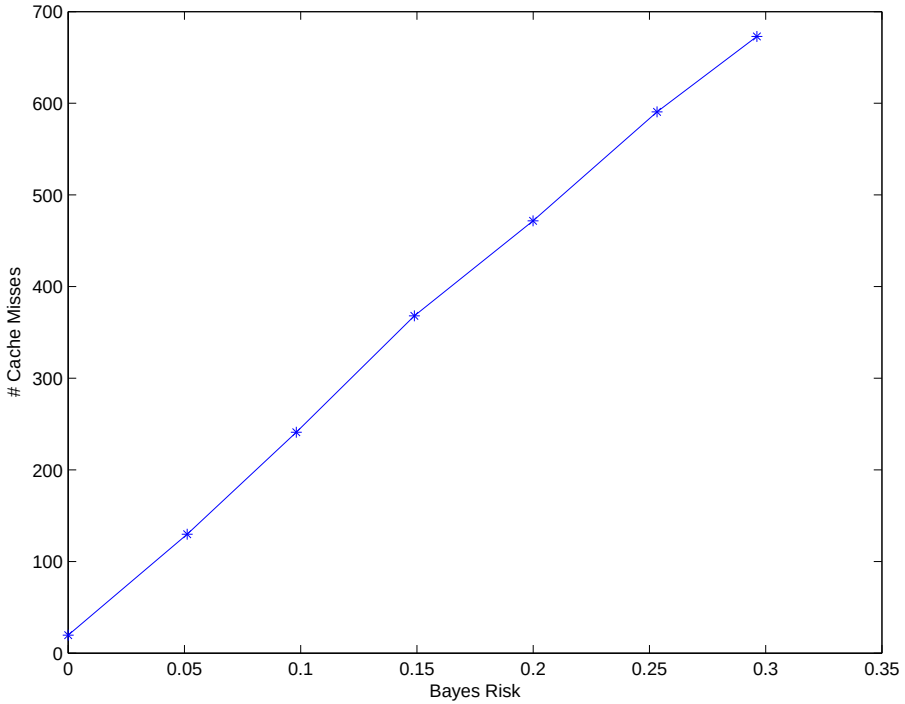


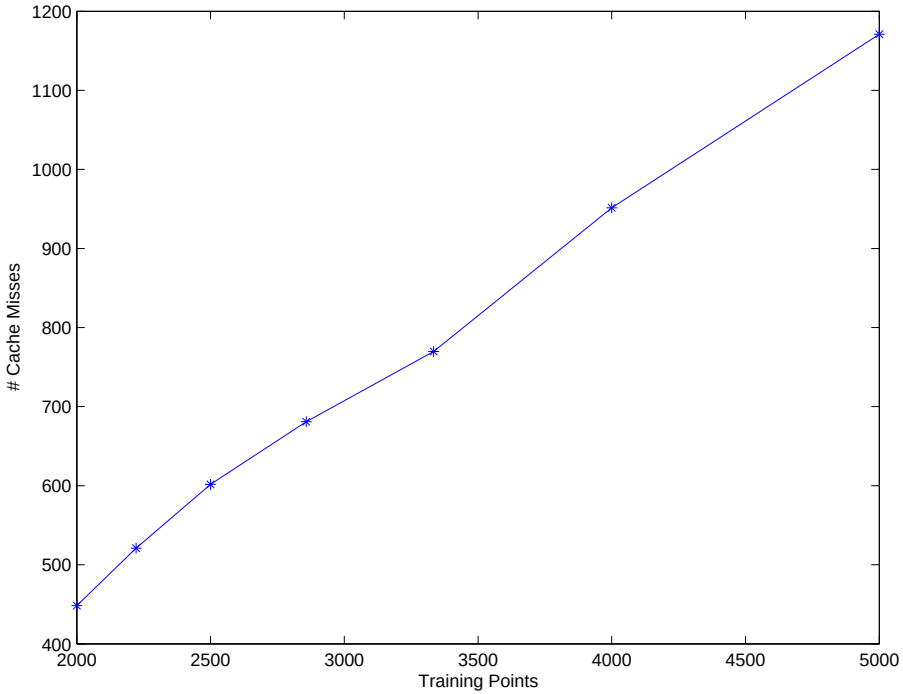


The noise is dominating the cache!









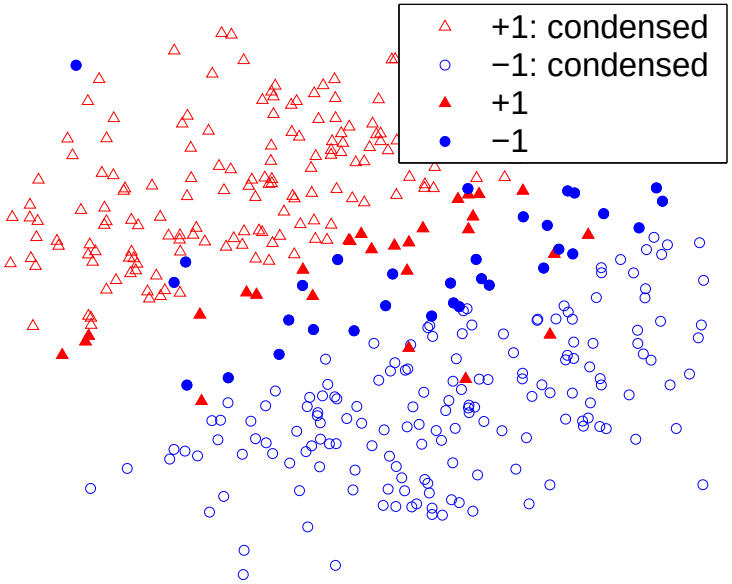
Cache Misses are linear related to the Bayes Risk.

Select patterns

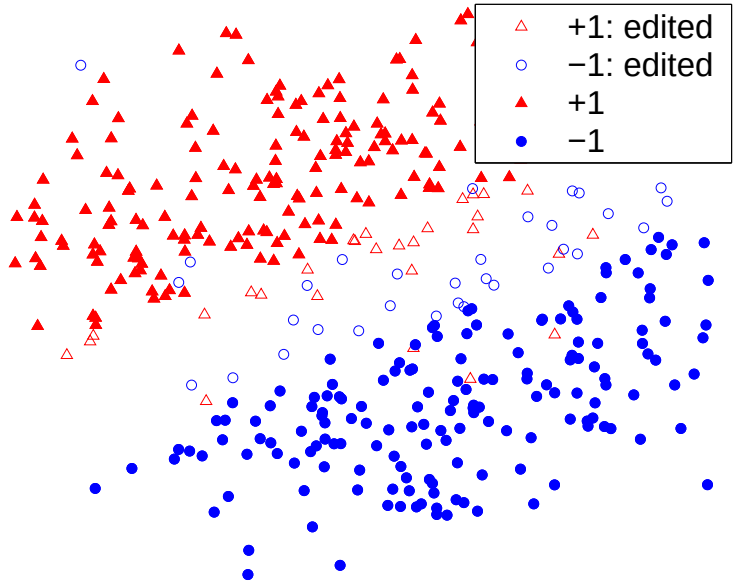
Idea: Carefully select subset of data!

Condensation and Editing

A trick from nearest neighbors training:



Condensation



Editing

Cross-Training

Reconsider Condense and MultiEdit:

Condense can be understood as: If $\mathbb{E}_S[y_i f_S(x_i)] > 1$, then remove it. SVM does this already.

MultiEdit can be understood as: If $\mathbb{E}_S[y_i f_S(x_i)] < 0$, then remove it. This is what we want.

We need to estimate the margin location $\mathbb{E}_S[y_i f_S(x_i)]$ of our point $\rightarrow$ Cross-validation.

Algorithm 1 (CROSSTRAINING).

1 Create s subsets of size N_2 by randomly drawing $N_2/2$ examples of each class.

2 Train s independent SVMs $f_1, \ldots, f_s$ using each of the subsets as the training set.

3 For each training example (x_i, y_i) estimate the margin average m_i and variance v_i:

$$m_i = \frac{1}{s}\sum_{r=1}^{s} y_i f_r(x_i) \qquad v_i = \frac{1}{s}\sum_{r=1}^{s} \left(m_i - y_i f_r(x_i)\right)^2$$

4 Discard all training examples for which $m_i + v_i < 0$.

5 Discard all training examples for which $m_i - v_i > 1$.

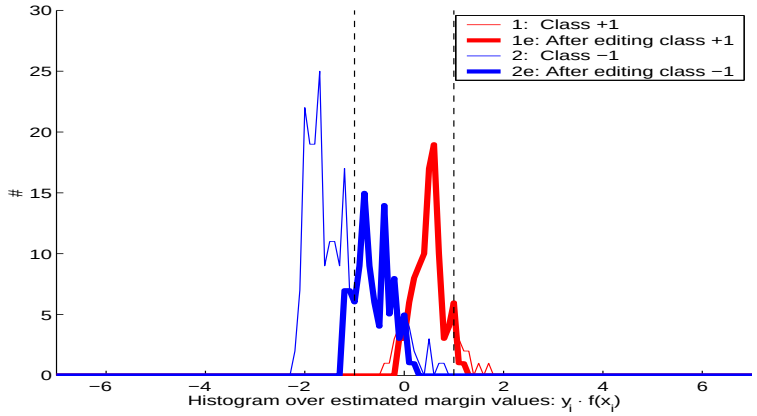
6 Train a final SVM on the remaining training examples.

Does it work?

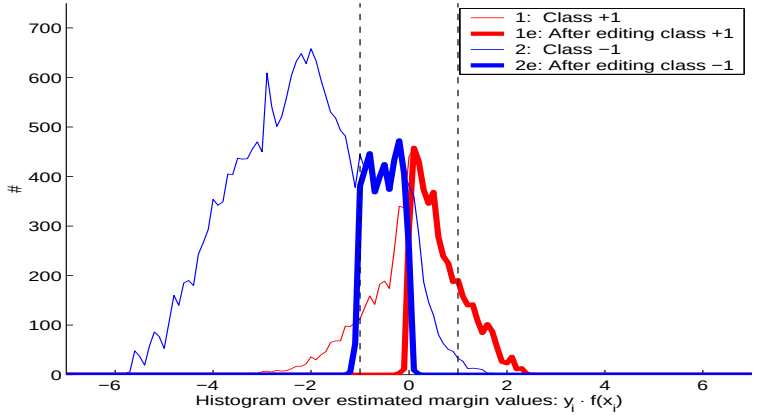
Dataset	Train Size	Test Size	SVM Perf.[%]	SVM #SV	XTrain Subsets	XTrain Perf.[%]	XTrain #SV
Banana	400	4900	**89.0**	111	5x 200	88.2	**51**
Waveform	400	4600	**90.2**	172	5x 200	88.7	**87**
Splice	1000	2175	90.0	601	5x 300	89.9	**522**
Adult	3185	16280	84.2	1207	5x 700	84.2	**606**
Adult	32560	16280	85.1	11325	5x 6000	84.8	**1194**
Forest	50000	58100	**90.3**	12476	5x 10000	89.2	**7967**
Forest	90000	58100	**91.6**	18983	5x 18000	90.7	**13023**
Forest	200000	58100	—	—	8x 30000	**92.1**	**19526**

But.... we lost control!

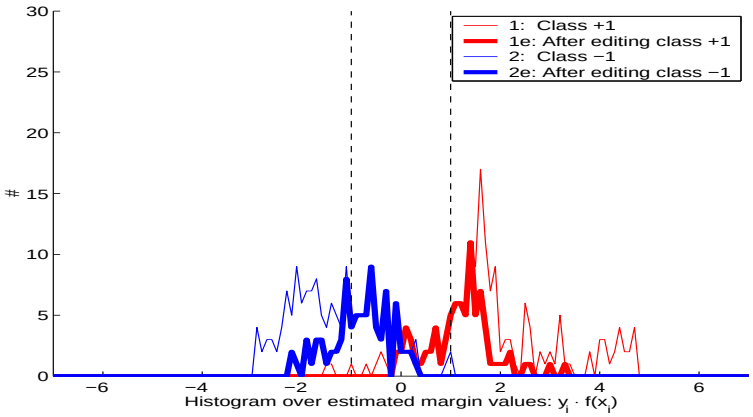
Dramatic Effect on the resulting class distribution $\Rightarrow$ Can lead to uncontrolled balance on the finite sample set.



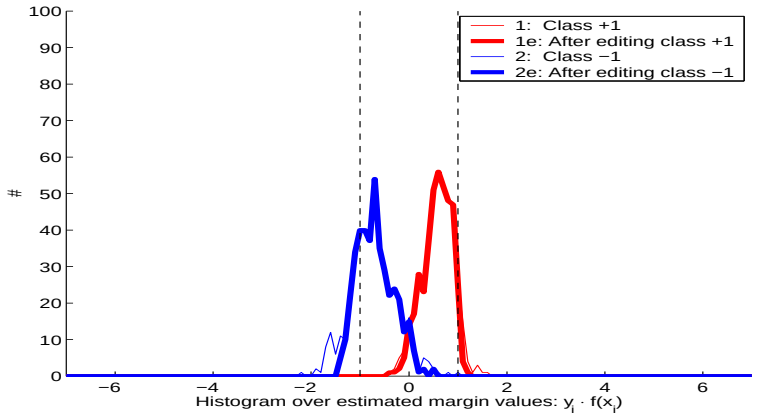
Waveform



Adult



Banana



Splice

Current Quests and Spinoff

Guessing:

Use $m_i = \frac{1}{s} \sum_{r=1}^{s} y_i f_r(x_i)$ to initialize Lagrangians of original problem and initialize cache.

We are looking for a training scheme such that:

- Controlled choice of subset should lead to better generalization error if size of subsets increase.

AND

- Does not destroy properties of original problem.

Overview

- Problem Statement
- Tricks in the primal
- Tricks in the dual
- Conjectures

Conjectures – Lynching the speaker is prohibited

1. Machine Learning is not equal Optimization $\Rightarrow$

Vicinity of Minima is sufficient.

2. If you demand less, you can do more $\Rightarrow$

Early stopping as additional regularizer might help handling large data sets.

3. No respect for the dual $\Rightarrow$

If you have to do approximations, do it in the primal!

If someone puts a gun on my head and asks me to do model selection by minimizing a cost function (bounds) or by cross validation:
I choose crossvalidation.

– Chih-Jen Lin + O. Chapelle –

SON

TEŞEKKÜRLER

References

Kimeldorf, G. S. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95.

Schölkopf, B., Herbrich, R., Smola, A. J., and Williamson, R. C. (2000). A Generalized Representer Theorem. Technical Report 2000-81, NeuroCOLT. Published in *Proceedings COLT'2001*, Springer Lecture Notes in Artificial Intelligence, 2001.

Steinwart, I. (2004). Sparseness of Support Vector Machines—Some Asymptotically Sharp Bounds. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.