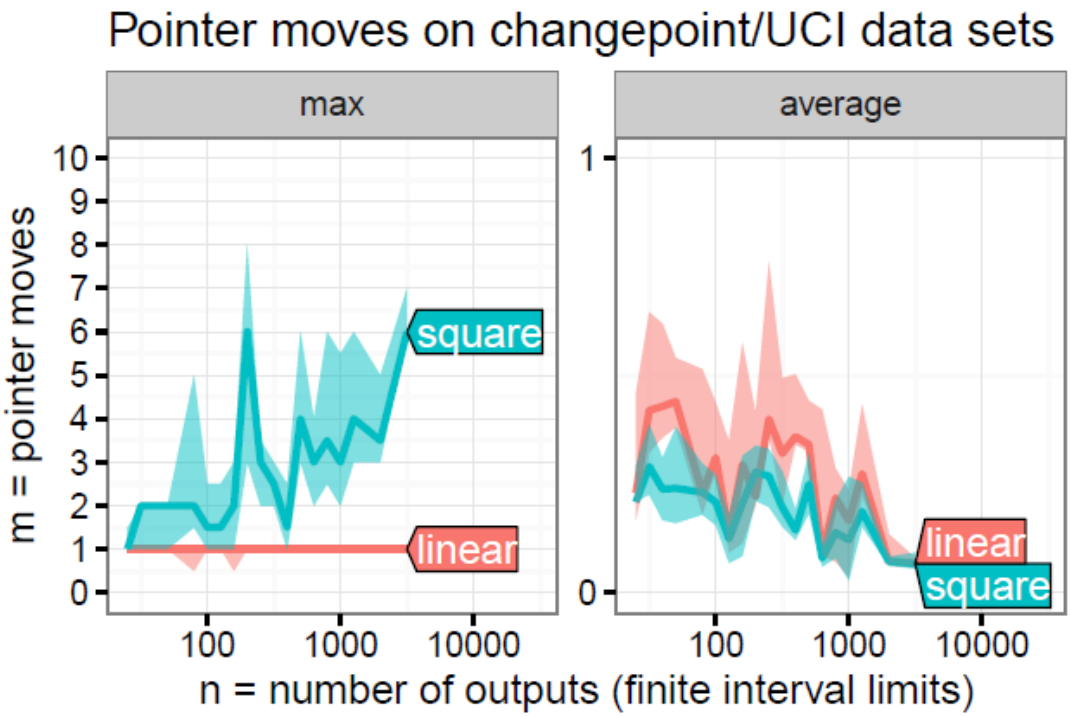
Figure 1



Drouin, et al. Maximum Margin Interval Trees, Figure 3

Inputs: n data points and m pointer moves per data point
Outputs: max and average number of pointer moves m over all real and simulated data sets
Function to learn: MMIT algorithm with squared and linear hinge loss solvers

This can be used in detection in DNA copy number. The paper uses changpoint and UCI data sets. The data sets can be found at https://github.com/aldro61/mmit-data. For the changpoint neuroblastoma data set, it has 3 files: features, folds, and targets. The features file has 252 columns and 324 rows, there are different statistical data. The folds file has 1 column and 324 rows. The targets file has 2 columns and 324 rows.

Features:

	n.identity	n.sqrt	n.log	n.square	data orig s	data orig s	data orig r	data orig r	data orig s	data orig s	data orig s	data orig s	data orig c	data orig c	data orig c	data orig c	data orig c	data orig c	data orig c	data orig c
2	474	21.77154	6.161207	224676	148.5757	22074.74	0.313451	0.098251	0.183431	0.428288	-1.69592	0.033647	-0.6529	0.259423	0.336855	0.412239	0.701327	0.42628	0.0673	0.113471
3	155	12.4499	5.043425	24025	14.34727	205.8442	0.092563	0.008568	0.263228	0.513057	-1.33474	0.069289	-0.3603	-0.17462	0.16221	0.319617	0.571434	0.12982	0.030493	0.026312
4	79	8.888194	4.369448	6241	-9.40281	88.41275	-0.11902	0.014166	0.186725	0.432117	-1.67812	0.034866	-0.41889	-0.25412	-0.13924	-0.03949	0.759582	0.175469	0.064576	0.019387
5	163	12.76715	5.09375	26569	-13.086	171.2434	-0.08028	0.006445	0.21294	0.461454	-1.54674	0.045343	-0.486	-0.25929	-0.10625	0.094906	0.480265	0.2362	0.067232	0.011289
6	118	10.86278	4.770685	13924	-14.2253	202.3584	-0.12055	0.014533	0.22976	0.479333	-1.47072	0.05279	-0.68966	-0.21301	-0.05365	0.018278	0.474047	0.475631	0.045372	0.002878
7	480	21.9089	6.173786	230400	-158.287	25054.91	-0.32977	0.108745	0.482915	0.694921	-0.72791	0.233207	-1.02915	-0.77658	-0.44526	0.049631	0.681674	1.059142	0.603073	0.198258
8	199	14.10674	5.293305	39601	-8.4505	71.41095	-0.04246	0.001803	0.349949	0.591565	-1.04997	0.122464	-0.712	-0.3555	-0.023	0.268	0.704	0.506944	0.12638	0.000529
9	188	13.71131	5.236442	35344	47.249	2232.468	0.251324	0.063164	0.336059	0.579706	-1.09047	0.112935	-0.164	0.011	0.068	0.5345	1.044	0.026896	0.000121	0.004624
10	256	16	5.545177	65536	-10.9722	120.3884	-0.04286	0.001837	0.111055	0.33325	-2.19773	0.012333	-0.507	-0.08975	-0.026	0.006	0.699	0.257049	0.008055	0.000676
11	223	14.93318	5.407172	49729	-1.4055	1.97543	-0.0063	3.97E-05	0.116412	0.341192	-2.15062	0.013552	-0.439	-0.094	0.01	0.0855	0.345	0.192721	0.008836	1.00E-04

Folds:

	fold
2	2
3	1
4	5
5	2

Targets:

	min.log.penalty	max.log.penalty
2	4.42830209165049	6.37383232266832
3	1.09399546115361	Inf
4	-0.0624500726264742	3.4031455381391
5	1.28791666958988	5.19372513867243
6	1.57044822729742	5.1660964615837

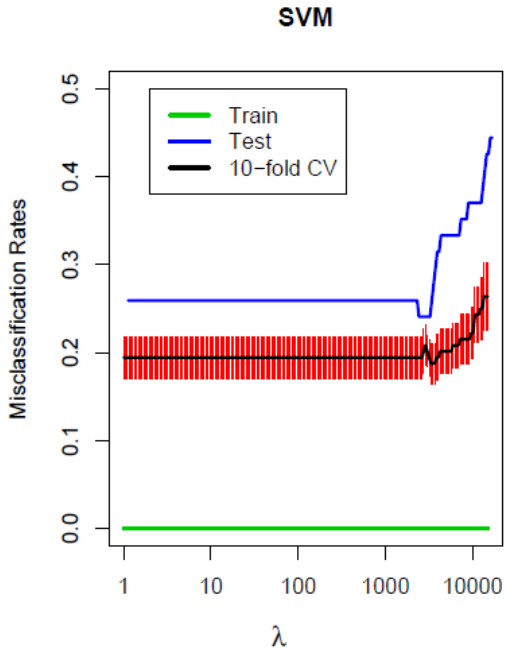
Pseudo-code:

Algorithm 1 Dynamic programming algorithm for computing minimum total hinge loss.

Input: limits $\mathbf{y} \in \mathbb{R}^n$, signs $s \in \{-1, 1\}$, margin $\epsilon \in \mathbb{R}$.
Initialize: $B \leftarrow \text{map}\{\}$, $J \leftarrow B.\text{end}()$, $M \leftarrow \text{Coefs}(0)$
for data points t from 1 to n:
  $f \leftarrow \text{Coefs}[s_t \ell(s_t(\mu - y_t) + \epsilon)]$
  $b \leftarrow y_t - s_t \epsilon$
  $B.\text{insert}(b, f)$
  if $0 < s_t(B[J].\text{breakpoint} - y_t) + \epsilon$:
    $M \leftarrow M + \text{Coefs}[\ell(s_t(\mu - y_t) + \epsilon)]$
  while !MinInInterval(M, B, J):
    if Increasing(M): $J \leftarrow J - 1$; $M \leftarrow M - B[J].\text{function}$
    else: $M \leftarrow M + B[J].\text{function}$; $J \leftarrow J + 1$
  $\mu_t^*, P_t^* \leftarrow \text{Minimize}(M, B, J)$
Output: $\mu^* \in \mathbb{R}^n, P^* \in \mathbb{R}^n$

I will implement the MMIT algorithm and process the dataset. Then collect the output data and generate the figure. In this coding project I will learn about the interval regression tree and hinge loss.

Figure 2



Hastie, et al. The Entire Regularization Path for the Support Vector Machine, Figure 11 Left

Inputs: The cancer dataset
Outputs: The misclassification rates for different methods
Function to learn: To learn how to calculate entire path of SVM solutions for every value of the cost parameter

This can be used in diseases classification by different features. The paper uses the dataset Multi-Class Cancer Diagnosis Using Tumor Gene Expression Signatures Ramaswamy et al., 2001, https://software.broadinstitute.org/cancer/software/genepattern/datasets. However, the file format of the dataset is not very familiar to me. It has 90 features and thousands rows.

Dataset:

Description	Accession	Normal__Breast_BR_1	Normal__Breast_BR_2	Normal__Breast_BR_3	Normal__Breast_BR_4	Normal__Breast_93_I_184	Normal__Prostate_PR_2	Normal__Prostate_PR_3
16063								
AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	12 A	61 A	-5 A	-13.3 A	-48 A	-46 A	-56.3 A	94 A	-32 A	-87 A	-62 A	87 A	-34 A	-25 A	-71 A	133 A	-98 A	-74.
AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	-231 A	-580 A	-608 A	-344.5 A	-86 A	-456 A	-343.8 A	-745 A	-143 A	-71 A	-210 A	-254 A	-558 A	
AFFX-BioB-3_at (endogenous control)	AFFX-BioB-3_at	-287 A	-980 A	-810 A	-489.7 A	-51 A	-642 A	-986.7 A	-782 A	-206 A	-42 A	-272 A	-790 A	-334 A	
AFFX-BioC-5_at (endogenous control)	AFFX-BioC-5_at	12 A	76 A	163 A	23.8 A	-21 A	-107 A	-46.1 A	-130 A	77 A	136 A	164 A	278 A	-47 A	36 A	-65 A	149 A	-197
AFFX-BioC-3_at (endogenous control)	AFFX-BioC-3_at	-257 A	-444 A	-429 A	-274.8 A	-60 A	-349 A	-506.7 A	-489 A	-107 A	-114 A	-159 A	-593 A	-406	
AFFX-BioDn-5_at (endogenous control)	AFFX-BioDn-5_at	-131 A	-209 A	-254 A	-228.6 A	-89 A	-222 A	-250.0 A	-311 A	-201 A	-128 A	-189 A	-107 A	-340
AFFX-BioDn-3_at (endogenous control)	AFFX-BioDn-3_at	-403 A	144 A	445 A	-173.4 A	5 A	-306 A	-623.5 A	-557 A	-68 A	279 A	212 A	-330 A	-400 A	-201 A	
AFFX-CreX-5_at (endogenous control)	AFFX-CreX-5_at	-108 A	-235 A	-194 A	-163.1 A	-36 A	-158 A	-303.0 A	-251 A	-147 A	-65 A	-108 A	-347 A	-124 A	
AFFX-CreX-3_at (endogenous control)	AFFX-CreX-3_at	-32 A	-10 A	-75 A	-0.0 A	17 A	35 A	45.7 A	-147 A	-14 A	-23 A	-38 A	-208 A	24 A	-12 A	187 A	-35 A	-88
AFFX-BioB-5_st (endogenous control)	AFFX-BioB-5_st	437 A	319 A	338 A	228.2 A	60 A	91 A	131.3 A	427 A	-1 A	0 A	16 A	197 A	53 A	-48 A	304 A	343 A	402 A	-38.
AFFX-BioB-M_st (endogenous control)	AFFX-BioB-M_st	-216 A	-615 A	-317 A	-342.0 A	16 A	-338 A	-305.4 A	-525 A	-60 A	-66 A	-314 A	392 A	-250 A	-52 A	
AFFX-BioB-3_st (endogenous control)	AFFX-BioB-3_st	-689 A	-716 A	-692 A	-541.2 A	-132 A	-342 A	-260.8 A	-415 A	-323 A	-181 A	-426 A	-797 A	-455
AFFX-BioC-5_st (endogenous control)	AFFX-BioC-5_st	-141 A	-463 A	-202 A	-142.6 A	-81 A	-322 A	-595.5 A	-510 A	-42 A	-55 A	-87 A	-210 A	-236 A	-145	
AFFX-BioC-3_st (endogenous control)	AFFX-BioC-3_st	-112 A	-172 A	-186 A	-164.6 A	-23 A	-49 A	-308.0 A	-108 A	-125 A	-101 A	-207 A	-373 A	-194 A	
AFFX-BioDn-5_st (endogenous control)	AFFX-BioDn-5_st	327 A	460 A	572 A	-120.3 A	57 A	74 A	174.1 A	197 A	94 A	84 A	49 A	4 A	54 A	-85 A	381 A	83 A	476 A
AFFX-BioDn-3_st (endogenous control)	AFFX-BioDn-3_st	413 A	472 A	571 A	198.3 A	19 A	97 A	452.2 A	593 A	81 A	155 A	163 A	956 A	202 A	9 A	428 A	359 A	447 A
AFFX-CreX-5_st (endogenous control)	AFFX-CreX-5_st	-19 A	-180 A	-241 A	-36.4 A	14 A	-110 A	-81.0 A	-183 A	0 A	-87 A	-52 A	-19 A	41 A	45 A	-156 A	
AFFX-CreX-3_st (endogenous control)	AFFX-CreX-3_st	-415 A	-487 A	-564 A	-411.9 A	-52 A	-515 A	-662.6 A	-437 A	-265 A	-92 A	-250 A	-932 A	-244 A	
hum_alu_at (miscellaneous control)	hum_alu_at	13937 P	3862 A	7770 A	14431.1 P	15825 P	9768 P	8731.8 A	7768 A	15189 P	13836 P	11446 P	12733 P	15873
AFFX-DapX-5_at (endogenous control)	AFFX-DapX-5_at	-378 A	-625 A	-1517 A	-389.5 A	22 A	-453 A	-259.3 A	-1147 A	-39 A	-36 A	-64 A	180 A	-478 A	-23 A	147
AFFX-DapX-M_at (endogenous control)	AFFX-DapX-M_at	-326 A	-938 A	-1427 A	-297.0 A	50 A	-213 A	-699.6 A	-712 A	102 A	-28 A	148 A	-57 A	-132 A	5 A	86

Algorithm:
Start with λ large and decrease it toward zero, keeping track of all the events that occur along the way. As λ decreases, $||\beta||$ increases, and hence the width of the margin decreases. As this width decreases, points move from being inside to outside the margin. Their corresponding ai change from $\alpha_i = 1$ when they are inside the margin $(y_i f(x_i) < 1)$ to $\alpha_i = 0$ when they are outside the margin $(y_i f(x_i) > 1)$. By continuity, points must linger on the margin $(y_i f(x_i) = 1)$ while their α_i decrease from 1 to 0. We will see that the $\alpha_i(\lambda)$ trajectories are piecewise-linear in λ, which affords a great computational savings: as long as we can establish the break points, all values in between can be found by simple linear interpolation. Note that points can return to the margin, after having passed through it.

I will implement the algorithm, and use the data set to output the training, test, and cross-validation error. Then change the value of λ and output a set of values, use these values to generate the figure. I will learn how to generate entire regularization path for SVM.