Incorporating known risk factors into models

Jiaxuan Wang

<2017-02-14 Tue>

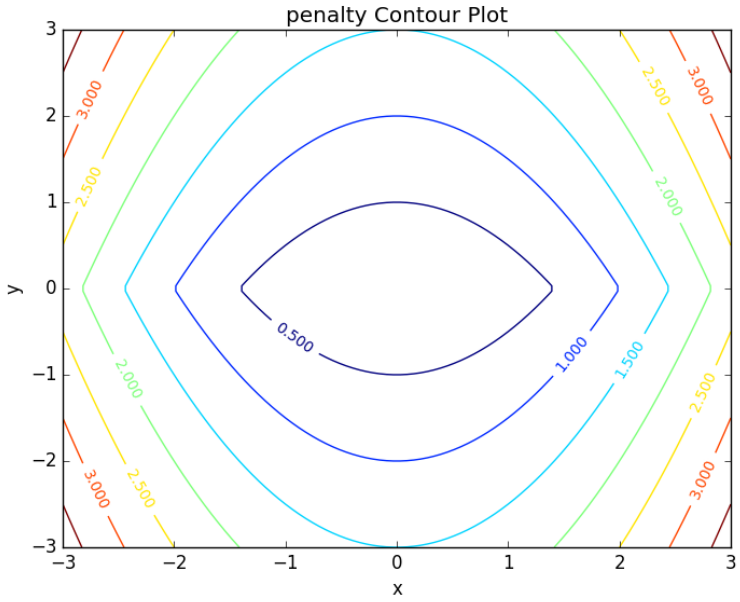
1 objective

Incorporating known risk factors with unknown risk factors in predicting outcome. In the case of choosing between correlated variables, the model should favor known risk factors.

2 approaches taken

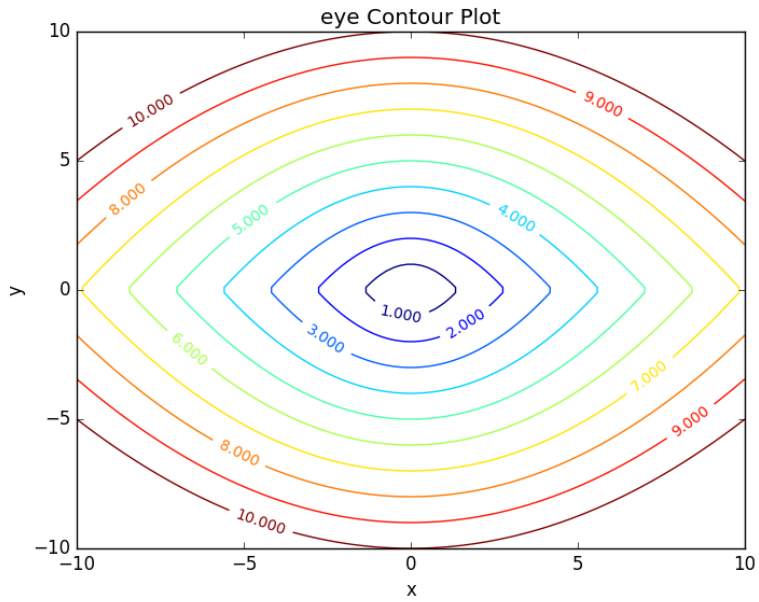
2.1 old approach

0.5 * λ_2 $||\mathrm{r} * \theta||_2^2$ + λ_1 $||(1\text{-}\mathrm{r}) * \theta||_1$ where r $\in \{0,1\}^{\mathrm{d}}$, $\theta \in \mathrm{R}^{\mathrm{d}}$

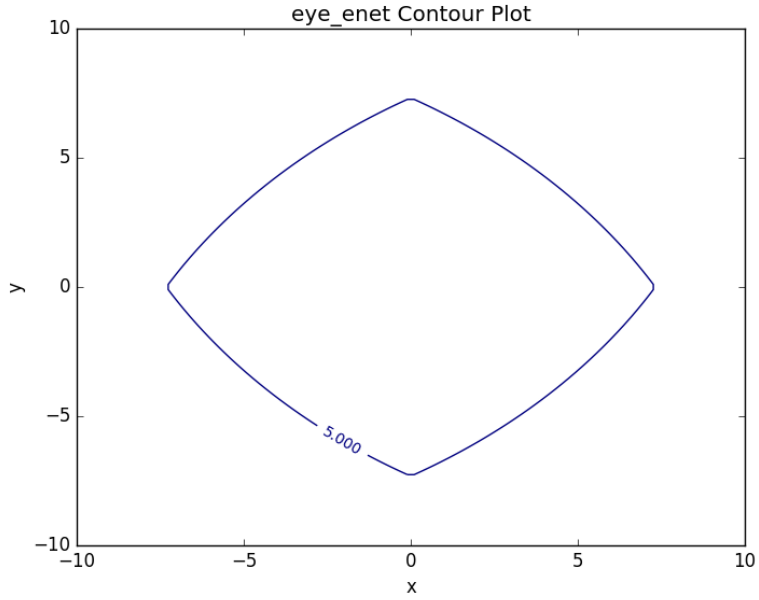


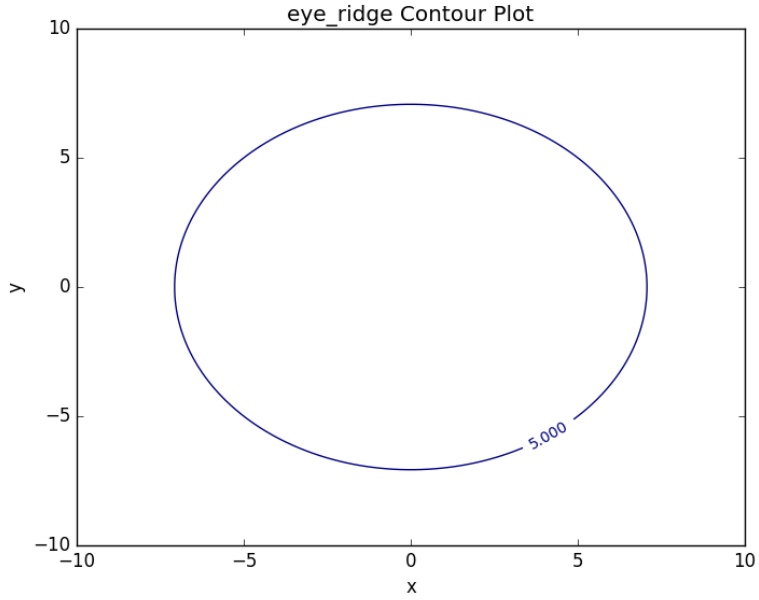
2.2 new approach

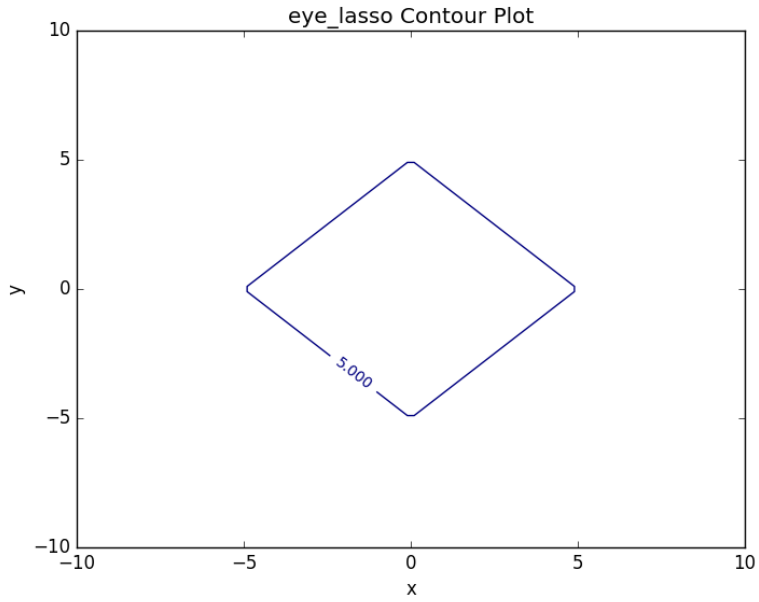
Fix a convex body that have property of the previous contour plot such that the angle at the end point is 45 degree. The following is the contour plot of its induced norm



In fact, by relaxing the constraint of r over binary to float, we can recover enet (setting r=0.5 * 1). Even without extending r, we can recover ridge (r=1) and lasso (r=0)


eye_enet Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000

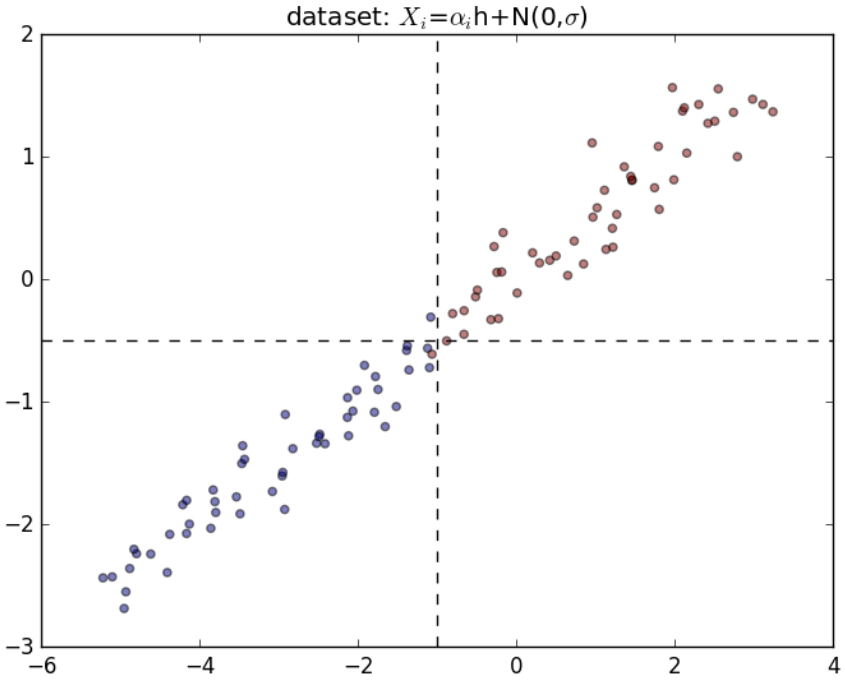

eye_ridge Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000



3 experiments

3.1 set up

Data n=100:



h = linspace(-2.5, 1, n)

x_0 ~ Uniform(1..4) h + N(0, 0.2)

x_1 ~ Uniform(1..4) h + N(0, 0.2)

Loss function is the negative loss likelihood of the logistic regression model.

Optimizer: AdaDelta

Number of Epoch: 1000

Regulizers: elastic net, lasso, ridge, OWL, weighted lasso, weighted ridge, penalty, eye penalty

3.1.1 eye penalty

pena(θ) = α * (0.5 * (1-c) * $||r * \theta||_2^2$ + c * $||(1-r) * \theta||_1$)

where r $\in [0,1]^d$, $\theta \in R^d$, $\alpha \in R$, c $\in R$

unit ball defined as:

pena(θ) = k

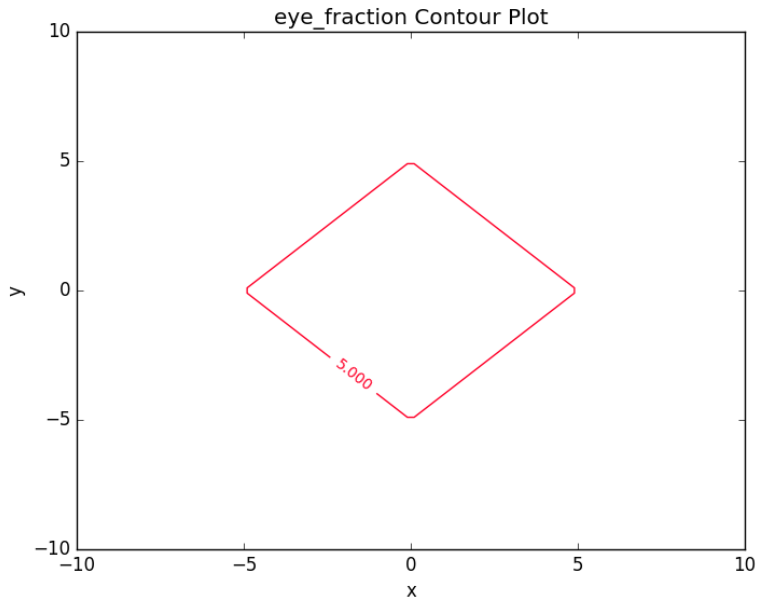
where k is chosen so that slope in the first quadrant between known risk factor x and unknown risk factor is -1

$eye_{penalty}(\theta)$ = |t| s.t pena(θ/t) = k if $\theta \neq 0$ else 0

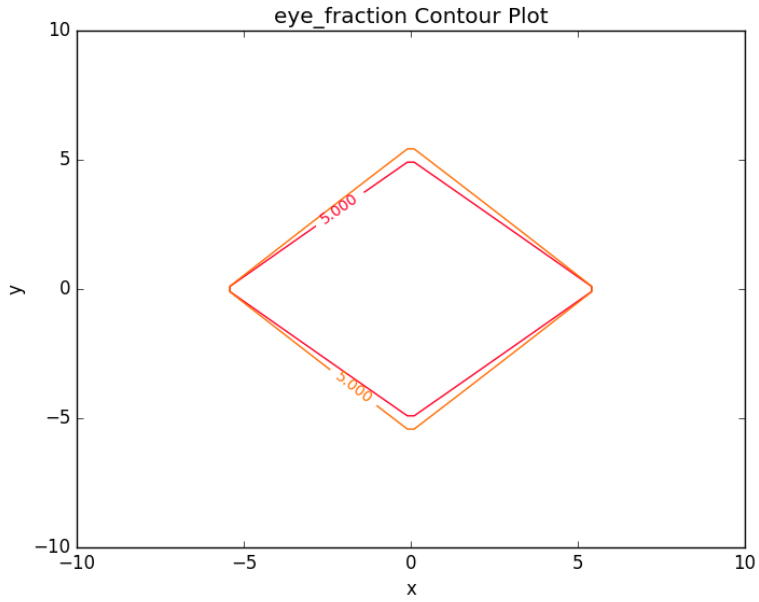
This is indeed a norm

1. $\text{eye}_{\text{penalty}}(\text{t } \theta) = |\text{t}| \text{ eye}_{\text{penalty}}(\theta)$
2. $\text{eye}_{\text{penalty}}(\theta + \beta) \leq \text{eye}_{\text{penalty}}(\theta) + \text{eye}_{\text{penalty}}(\beta)$
3. $\text{eye}_{\text{penalty}}(\theta) = 0$ iff $\theta = 0$

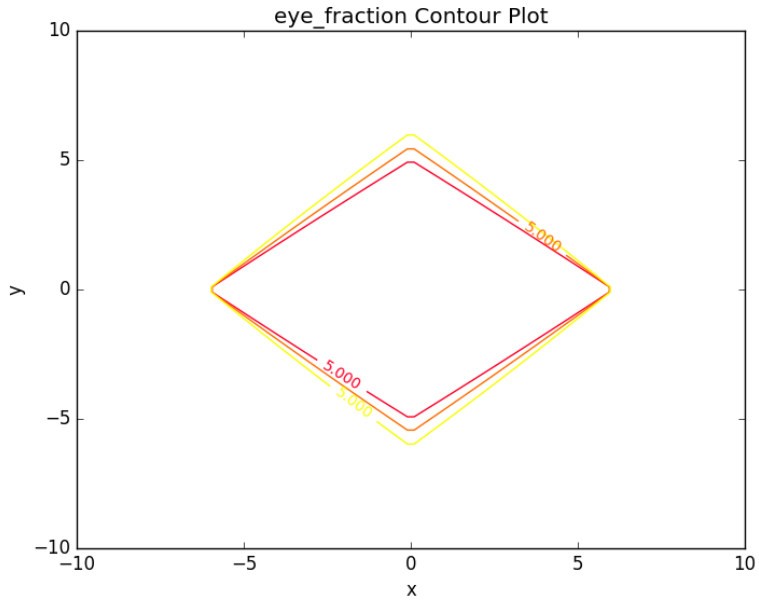
varying r_1 and r_2
$r_1 = 0.0$



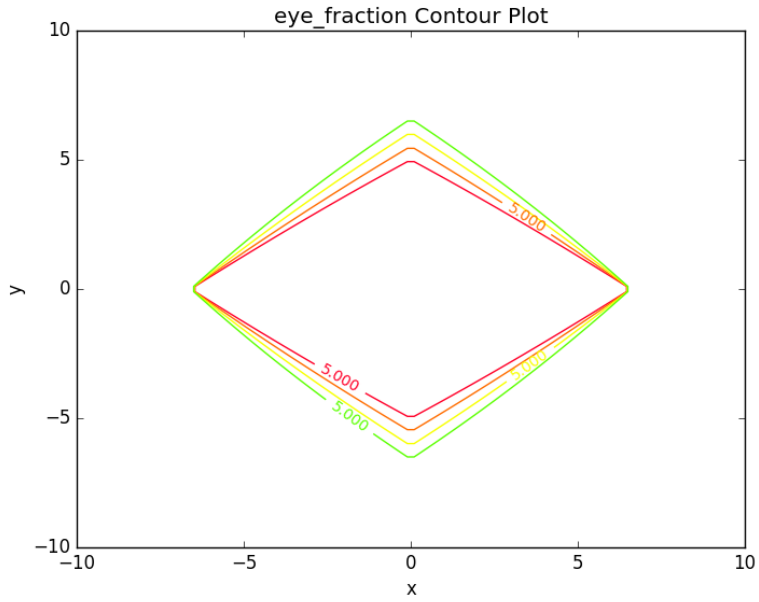
$r_1 = 0.1$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000

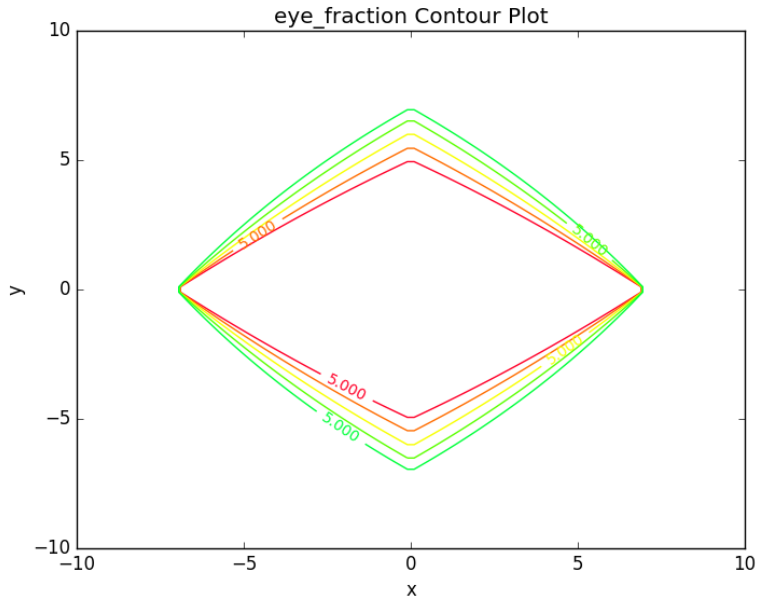
$r_1 = 0.2$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000

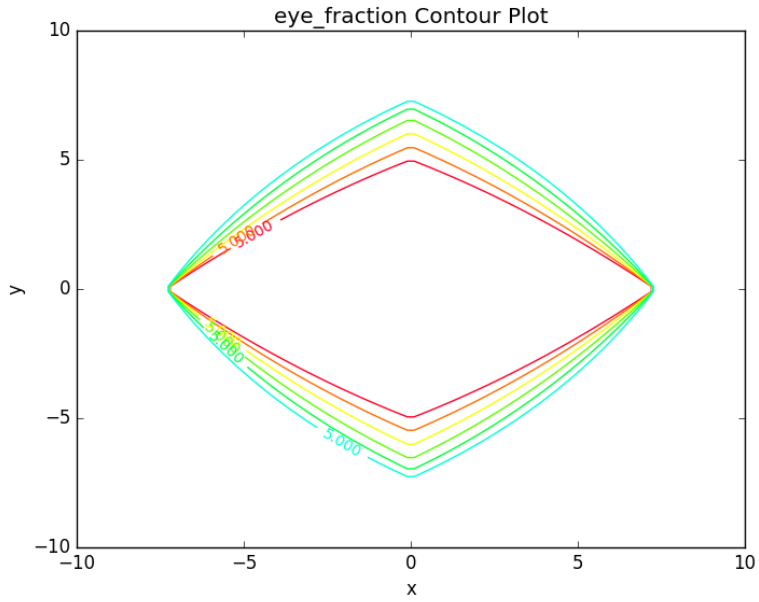
$r_1 = 0.3$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000

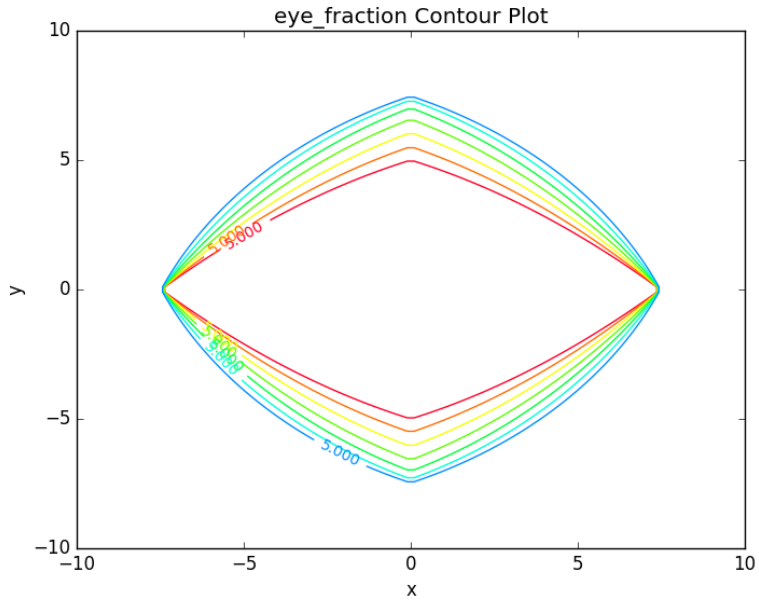
$r_1 = 0.4$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000

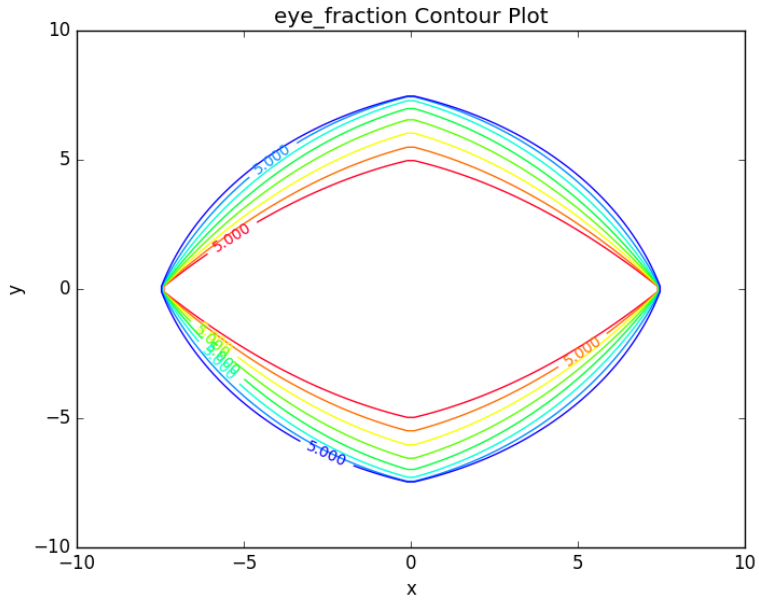
$r_1 = 0.5$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000

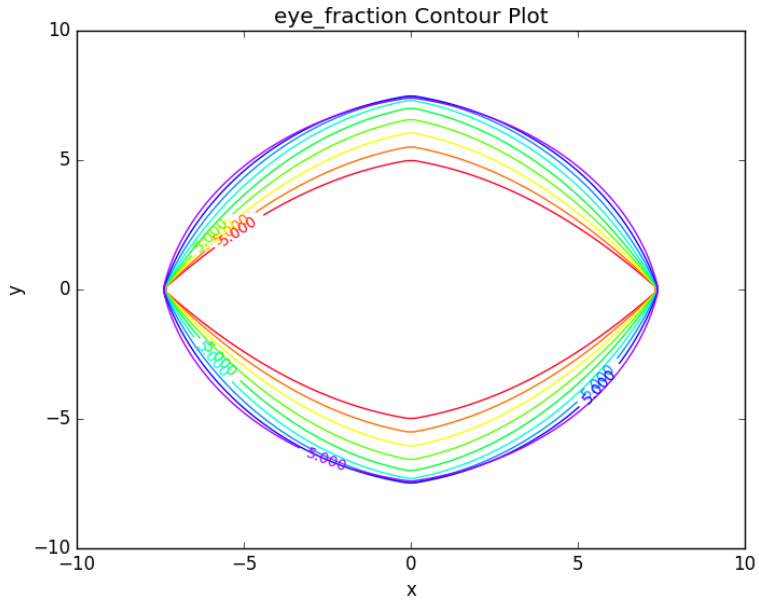
$r_1 = 0.6$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000

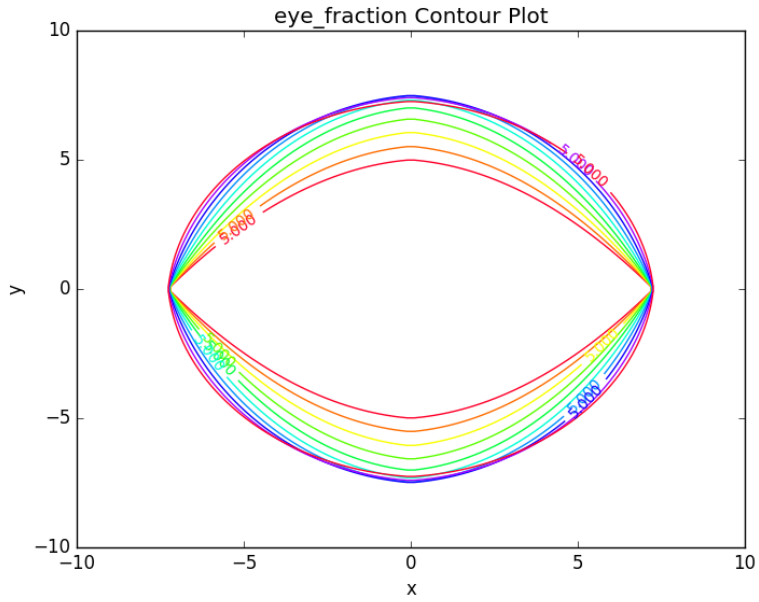
$r_1 = 0.7$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000
5.000

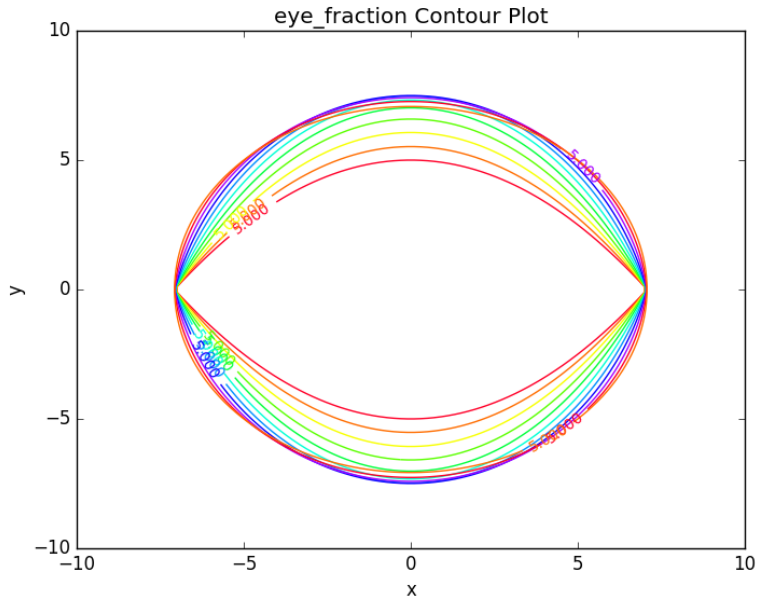
$r_1 = 0.8$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000
5.000

$r_1 = 0.9$

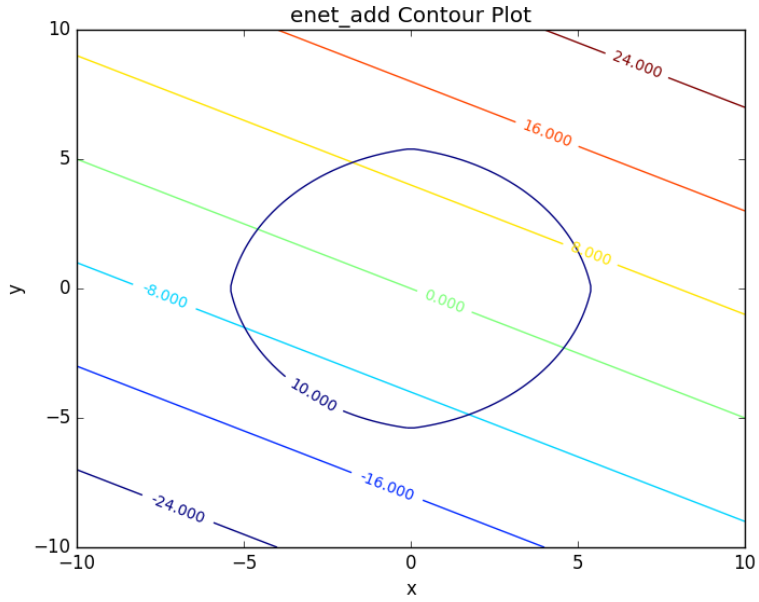

eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000
5.000

$r_1 = 1.0$


eye_fraction Contour Plot
10
5
0
−5
−10
y
−10
−5
0
5
10
x
5.000
5.000
5.000
5.000
5.000

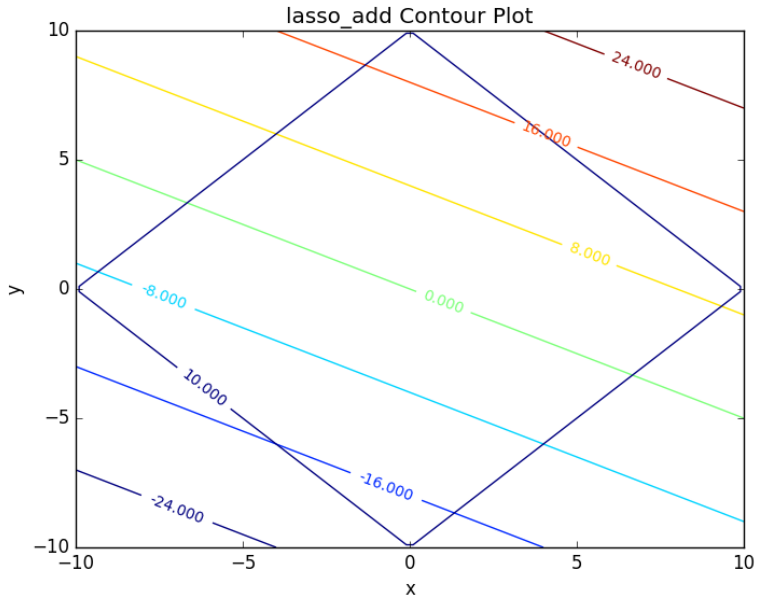
3.1.2 elastic net

α * (c * $||\theta||_1$ + 0.5 * (1 - c) * $||\theta||_2^2$) where c is a scaler



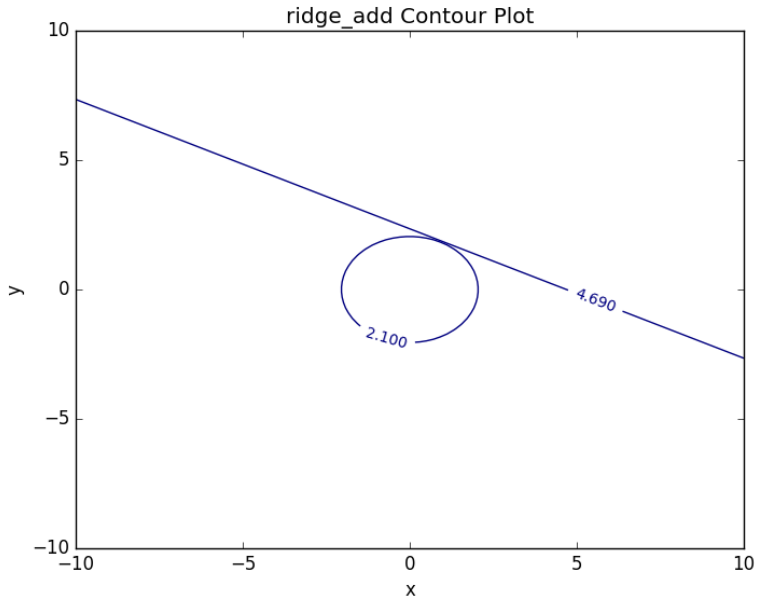
3.1.3 lasso

α * $||\theta||_1$



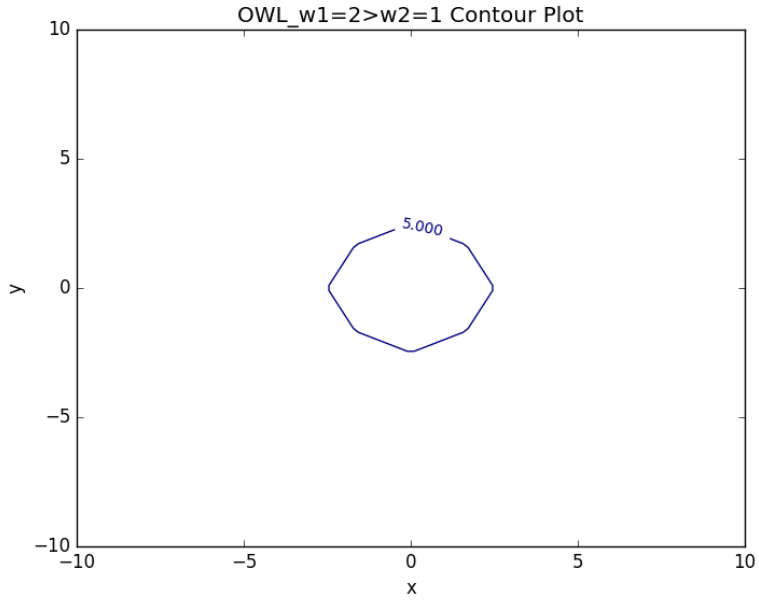
3.1.4 ridge

$0.5 * \alpha * ||\theta||_2^2$

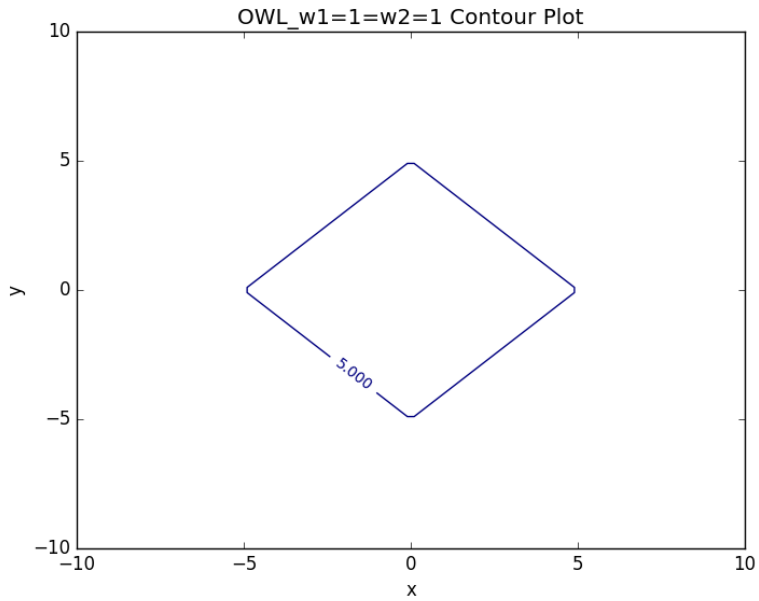


3.1.5 OWL

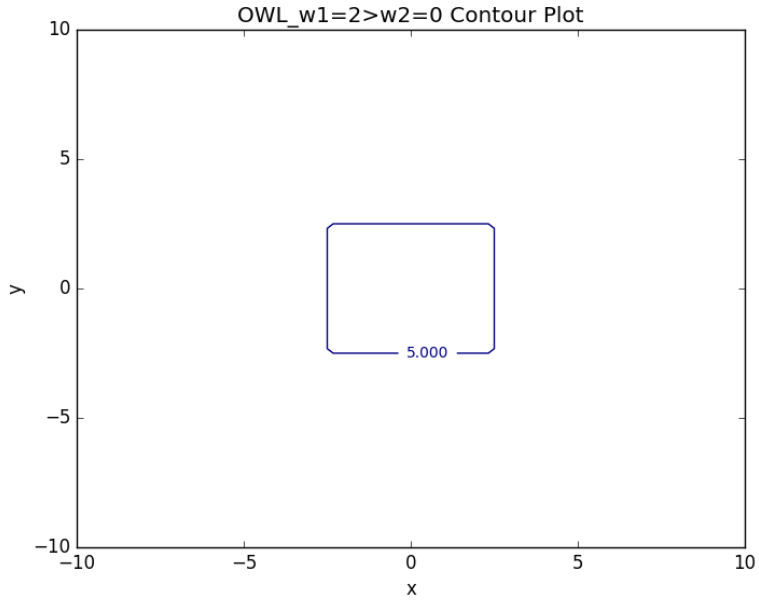
$\alpha * \sum_{i=1}^{n} w_i \ |x|_{[i]}$ where $w \in K_{m+}$ (monotone nonnegative cone)



degenerated case: back to lasso



degenerated case: back to l_{inf}



some properties:

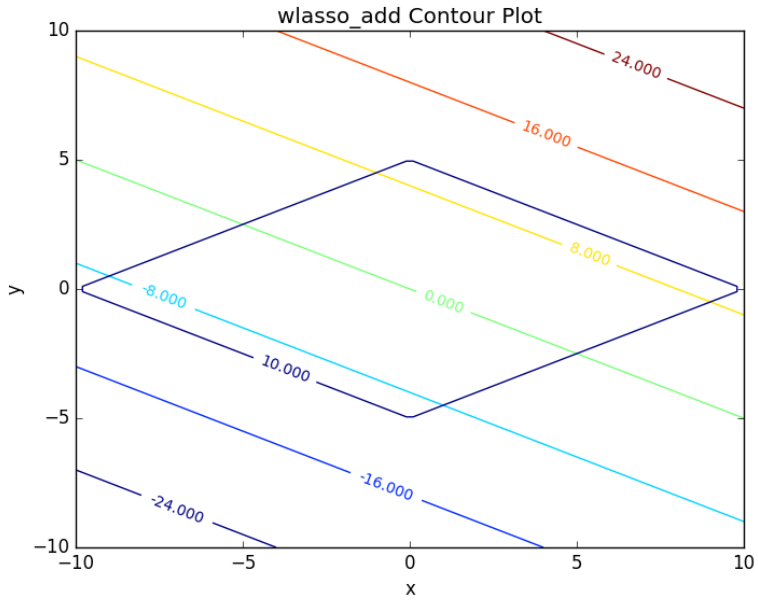
generalization of OSCAR norm

symmetry with respect to signed permutations

in the regular case, the minimal atomic set for this norm is known (the corners are easily calculated)

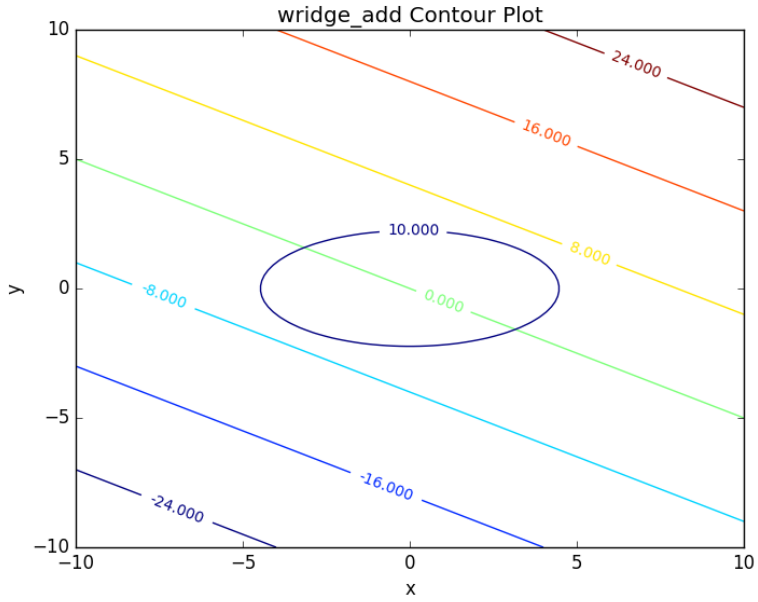
3.1.6 weighted lasso

α * ||w * θ||$_1$ where w $\in$ R_+$^{\mathrm{d}}$



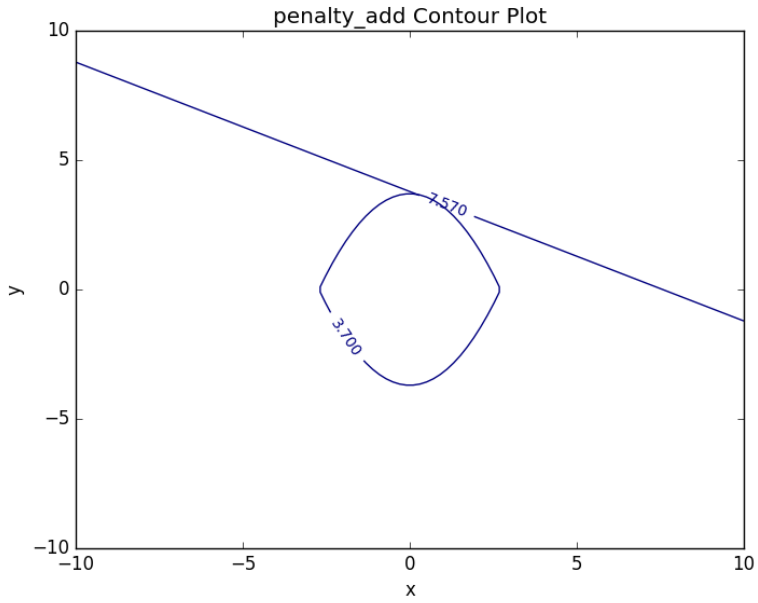
3.1.7 weighted ridge

$0.5 * \alpha * ||\mathrm{w} * \theta||_2^2$ where $\mathrm{w} \in \mathrm{R}_+^{\mathrm{d}}$



3.1.8 old penalty

α * (0.5 * (1-c) * $||r * \theta||_2^2$ + c * $||(1-r) * \theta||_1$) where r $\in \{0,1\}^d$, $\theta \in R^d$, $\alpha \in R$, c $\in R$



3.2 running procedure

3.2.1 first run (regularized b)

b regularized

fix hyperparmeters to predefined value

repeat the following 100 times:

generate data (x2 = 2x1), run the selected regularizers, record θ

3.2.2 second run (unregularized b, validation)

b unregularized

generate two datasets (x2 = 2x1), one for training, one for validation

parameter search over the different hyperparams of the regularizers

for each regularizer, use the hyperparmeters that acheives the minimal loss

repeat the following 100 times:

generate data, run the selected regularizers, record θ

3.2.3 third run (data normalized, eye penalty)

b unregularized

generate two datasets (x2 = 2x1), one for training, one for validation

normalize the data to 2 mean and 2 variance (validaton data is normalized using mean and variance for the training data)

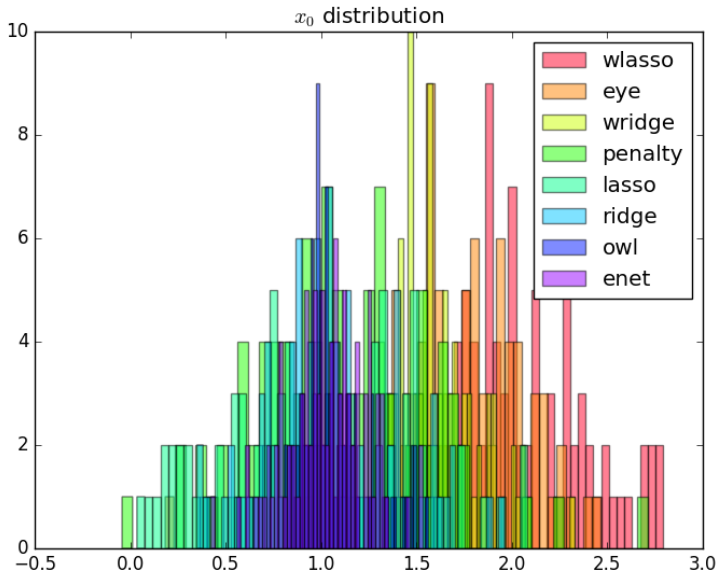
parameter search over the different hyperparams of the regularizers

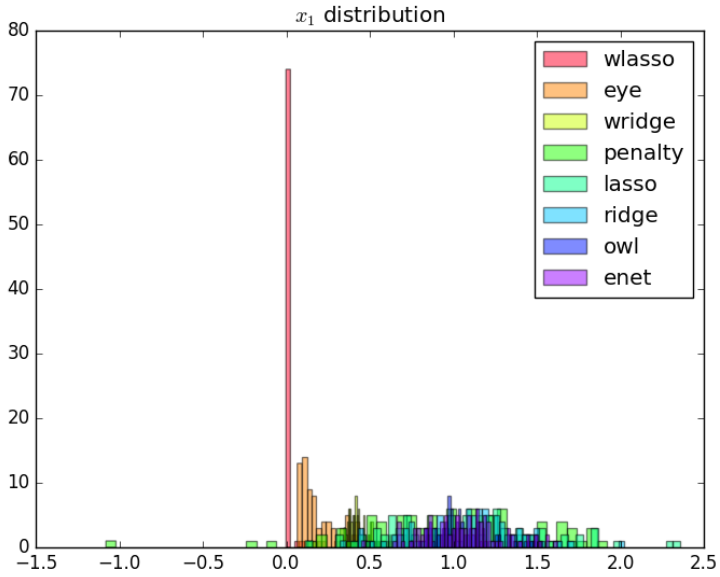
for each regularizer, use the hyperparmeters that acheives the minimal loss

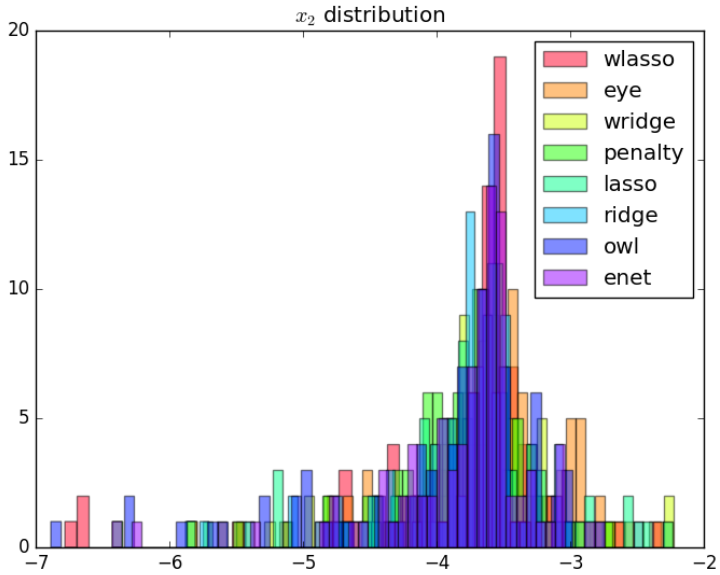
repeat the following 100 times:

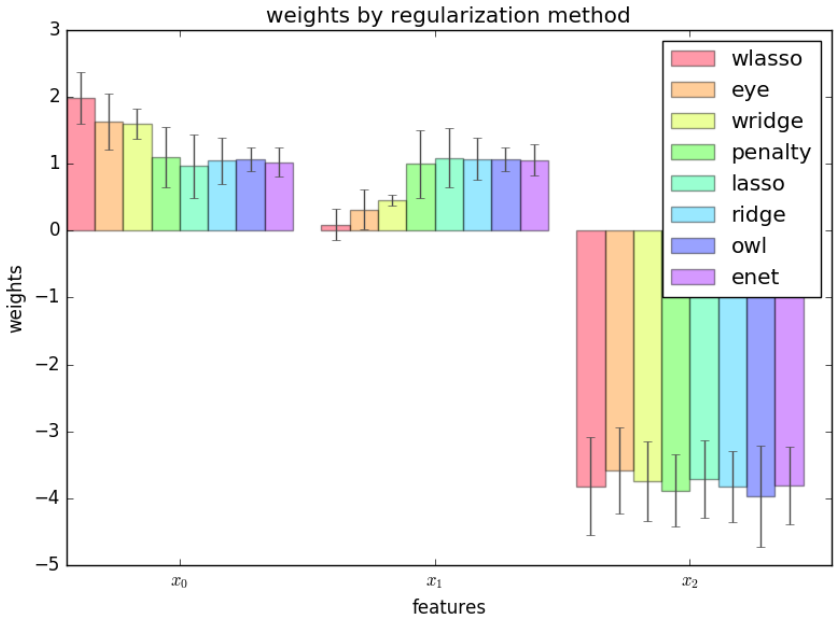
generate data, normalize data, run the selected regularizers, record θ

The choosing criteria is still loss b/c AUROC is always going to be 1 in the deterministic case:




x_1 distribution
wlasso
eye
wridge
penalty
lasso
ridge
owl
enet
80
70
60
50
40
30
20
10
0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


x_2 distribution
wlasso
eye
wridge
penalty
lasso
ridge
owl
enet
20
15
10
5
0
−7
−6
−5
−4
−3
−2



3.2.4 Fourth run (noise added)

b unregularized

generate two datasets, one for training, one for validation

normalize the data to 2 mean and 2 variance (validaton data is normalized using mean and variance for the training data)

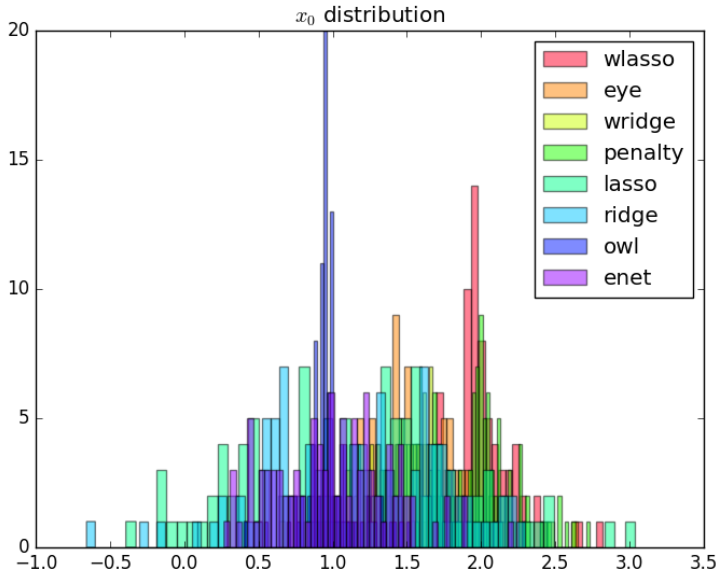
parameter search over the different hyperparams of the regularizers

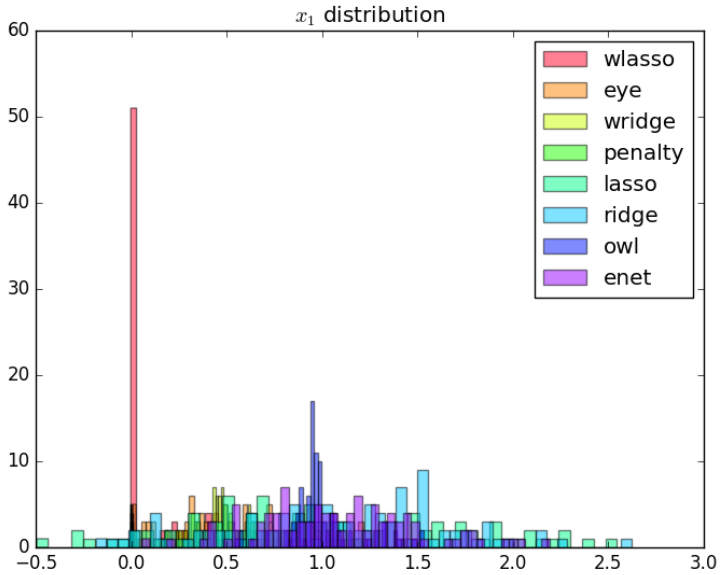
for each regularizer, use the hyperparmeters that acheives the minimal loss

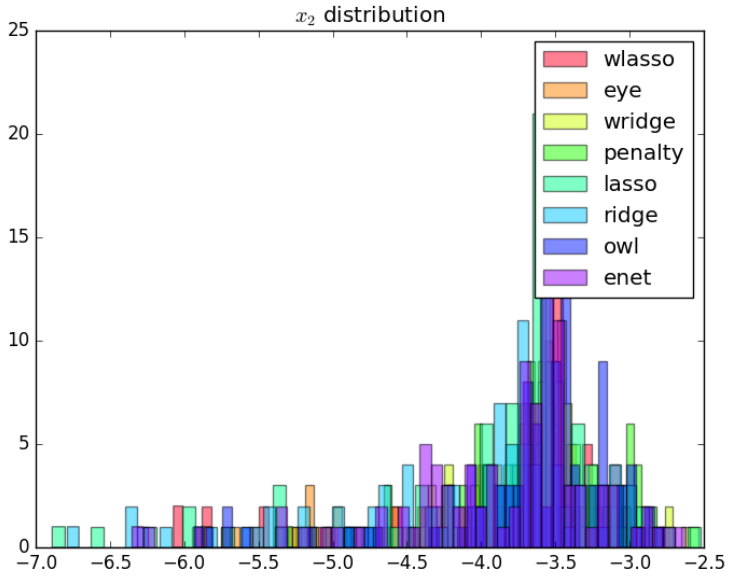
repeat the following 100 times:

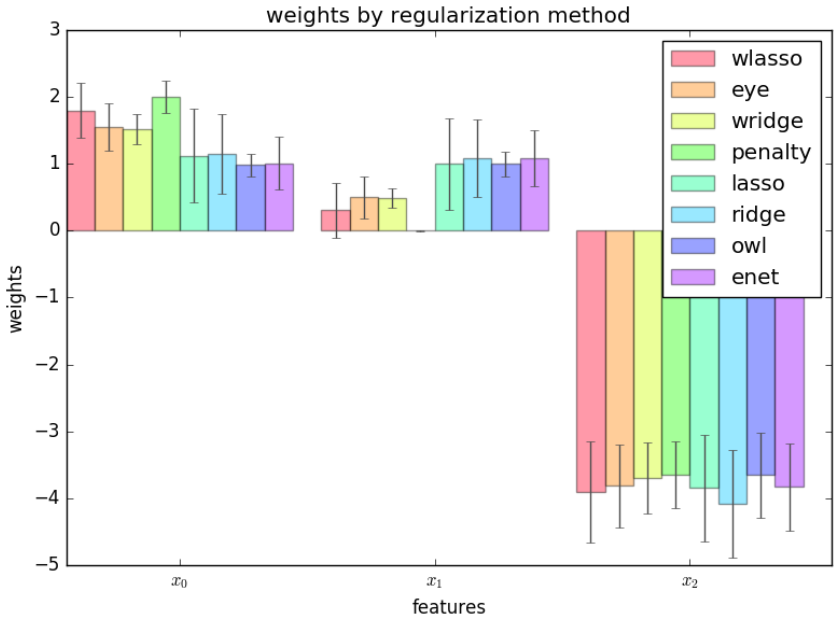
generate data (x_i = Uniform(0..4) h + N(0,0.2)), normalize data, run the selected regularizers, record θ

The choosing criteria is loss


x_0 distribution
wlasso
eye
wridge
penalty
lasso
ridge
owl
enet
20
15
10
5
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


x_1 distribution
wlasso
eye
wridge
penalty
lasso
ridge
owl
enet
60
50
40
30
20
10
0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0





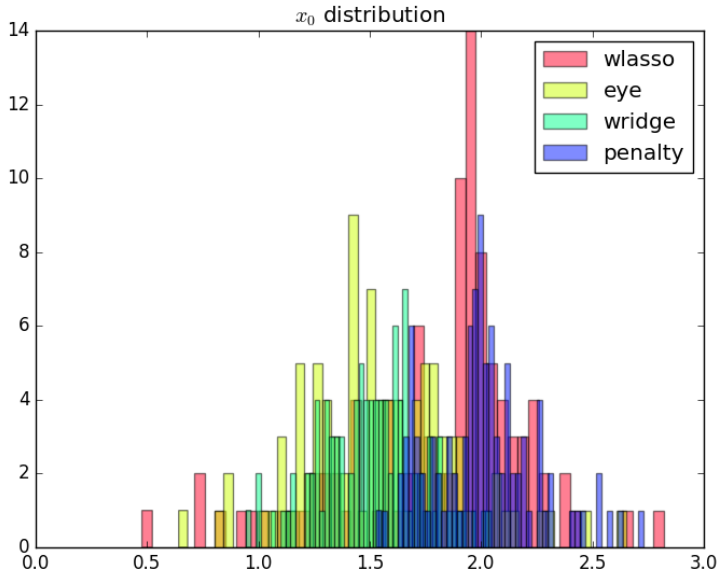
hyper parameter used:

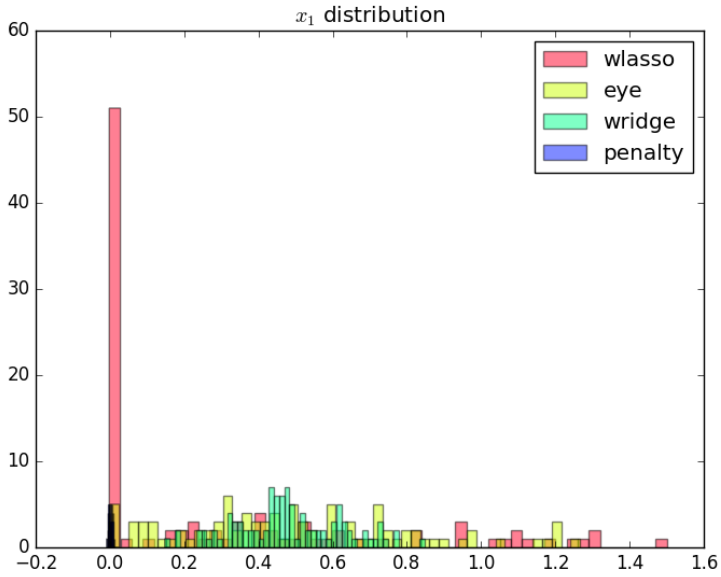
- enet(0.01, 0.2)
- eye(array([1., 0.]), 0.01, 0.4)

- lasso(0.0001)
- OWL([2, 1], 0.01)
- penalty(array([1., 0.]), 0.1, 1.0)
- ridge(0.001)
- weightedLasso(array([1., 2.]), 0.01)
- weightedRidge(array([1., 2.]), 0.01)

The sparsity in penalty can be explained as I placed no constraint on known risk factor (l1 ratio is 1), so it only regularizes x_1 not x_0

Also note that $l1_{ratio}$ controls the second derivative of eye penalty (in fact of magnitude (1-$l1_{ratio}$)/ $l1_{ratio}$). This means that smaller $l1_{ratio}$ makes the solution less sparse.




x_1 distribution
wlasso
eye
wridge
penalty
60
50
40
30
20
10
0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6