Rekenen_met_planten

September 7, 2014

1 Calculating with plants

In this notebook we performed a simple exploratory data analysis of the tree data. We did the following simple experiments

- predicting temperature based on tree measurements in time
- preforming feature selection
- using ICA to find independent subsystems

```
In [1]: import numpy as np
        import pylab as pl
        import seaborn as sb
        from sklearn import linear_model
        import pandas as pd
```

```
In [2]: STEPS_FOR_TESTING = 24*60*1.5
        A_DAY = 60*24

        # data preparation
        data = np.genfromtxt('20140907_data_plants_trial.csv', delimiter=',')
        Y = data[1:,1]
        Y[np.isnan(Y)]=0 # replace nan's by zero, better use interpolation!!!
        X = data[1:,2:]
        X_norm=(X-np.mean(X,0))/np.std(X,0)
        Y_norm=(Y-np.mean(Y))/np.std(Y)

        features = [ u'LPS1', u'LPS5', u'LPS4', u'LPS6', u'TDP0cm', u'TDP14cm']
```

```
In [3]: # EXPERIMENT: memory versus prediction RIDGE REGRESSION
        # train and test data
        Y_norm_train = Y_norm[A_DAY:-A_DAY-STEPS_FOR_TESTING-1]
        Y_norm_test = Y_norm[-A_DAY-STEPS_FOR_TESTING:-A_DAY-1]
```

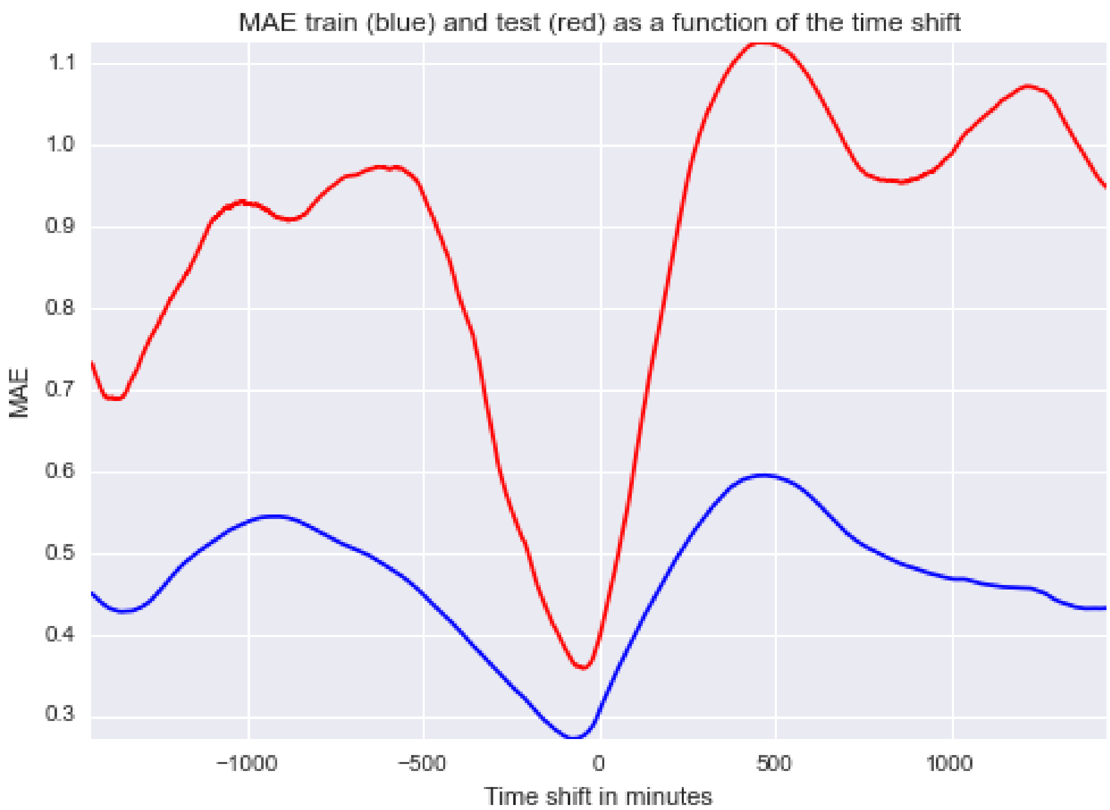
```
-c:3: DeprecationWarning: using a non-integer number instead of an integer will result in an error in th
-c:4: DeprecationWarning: using a non-integer number instead of an integer will result in an error in th
```

```
In [4]: clf = linear_model.Ridge(fit_intercept=False,normalize=False,alpha=0.1)
        MAE_train = []
        MAE_test = []
        predictions = []
        for time in range(-A_DAY,A_DAY):
            X_norm_train = X_norm[A_DAY-time:-A_DAY-STEPS_FOR_TESTING-1-time,:]   # -time: we go from f
            X_norm_test = X_norm[-A_DAY-STEPS_FOR_TESTING-time:-A_DAY-1-time,:]
            clf.fit(X_norm_train, Y_norm_train)
```

```
        # train error
        Y_pred = clf.predict(X_norm_train)
        MAE_train.append(np.mean(np.absolute(Y_pred-Y_norm_train)))
        # test error
        Y_pred = clf.predict(X_norm_test)
        predictions.append(Y_pred)
        MAE_test.append(np.mean(np.absolute(Y_pred-Y_norm_test)))
```

```
-c:6: DeprecationWarning: using a non-integer number instead of an integer will result in an error in th
-c:7: DeprecationWarning: using a non-integer number instead of an integer will result in an error in th
```

In [5]:
```
# plot MAE in function of time
ax = pl.gca()
ax.set_color_cycle(['b', 'r'])

ax.plot(range(-A_DAY,A_DAY), MAE_train)
ax.plot(range(-A_DAY,A_DAY), MAE_test)
pl.xlabel('Time shift in minutes')
pl.ylabel('MAE')
pl.title('MAE train (blue) and test (red) as a function of the time shift')
pl.axis('tight')
pl.show()
```

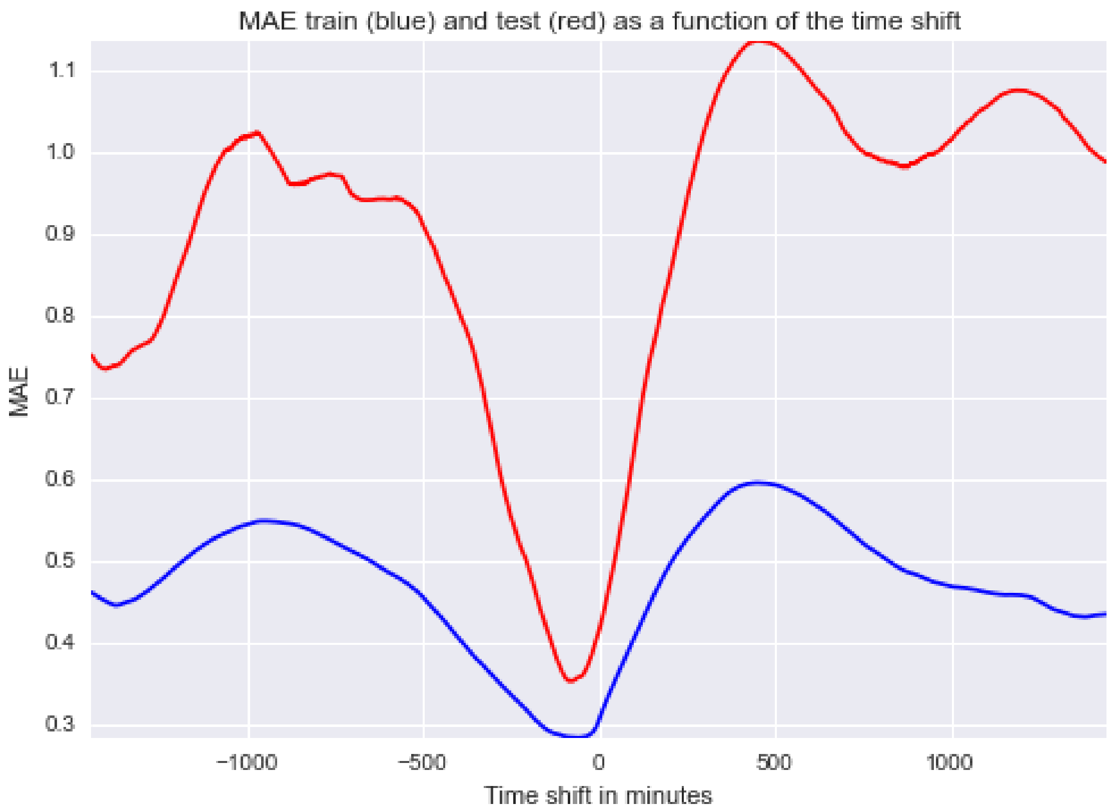


It is clear that we can predict temperature based on the tree measurements. The best prediction is predicting the temperature some moments in the past.

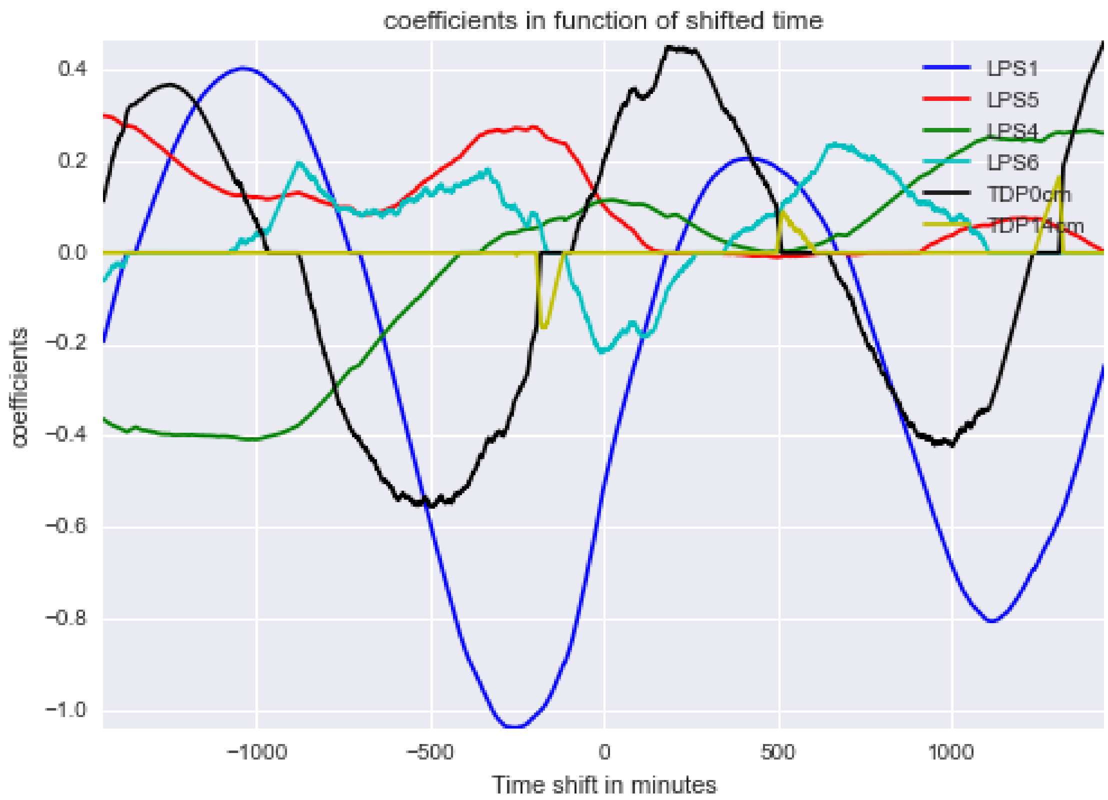
In [6]:
```
# EXPERIMENT: memory versus prediction with LASSO
# train and test data
Y_norm_train = Y_norm[A_DAY:-A_DAY-STEPS_FOR_TESTING-1]
Y_norm_test = Y_norm[-A_DAY-STEPS_FOR_TESTING:-A_DAY-1]

clf = linear_model.Lasso(fit_intercept=False,normalize=False,alpha=0.01)
MAE_train = []
MAE_test = []
predictions = []
coefs = np.zeros((len(range(-A_DAY,A_DAY)), 6))
nr = 0
for time in range(-A_DAY,A_DAY):
    X_norm_train = X_norm[A_DAY-time:-A_DAY-STEPS_FOR_TESTING-1-time,:]   # -time: we go from f
    X_norm_test = X_norm[-A_DAY-STEPS_FOR_TESTING-time:-A_DAY-1-time,:]
    clf.fit(X_norm_train, Y_norm_train)
    coefs[nr] = clf.coef_
    # train error
    Y_pred = clf.predict(X_norm_train)
    MAE_train.append(np.mean(np.absolute(Y_pred-Y_norm_train)))
    # test error
    Y_pred = clf.predict(X_norm_test)
    predictions.append(Y_pred)
    MAE_test.append(np.mean(np.absolute(Y_pred-Y_norm_test)))
    nr += 1
```

```
-c:13: DeprecationWarning: using a non-integer number instead of an integer will result in an error in
-c:14: DeprecationWarning: using a non-integer number instead of an integer will result in an error in
```

In [7]:
```
# plot MAE in function of time
ax = pl.gca()
ax.set_color_cycle(['b', 'r'])

ax.plot(range(-A_DAY,A_DAY), MAE_train)
ax.plot(range(-A_DAY,A_DAY), MAE_test)
pl.xlabel('Time shift in minutes')
pl.ylabel('MAE')
pl.title('MAE train (blue) and test (red) as a function of the time shift')
pl.axis('tight')
pl.show()
```



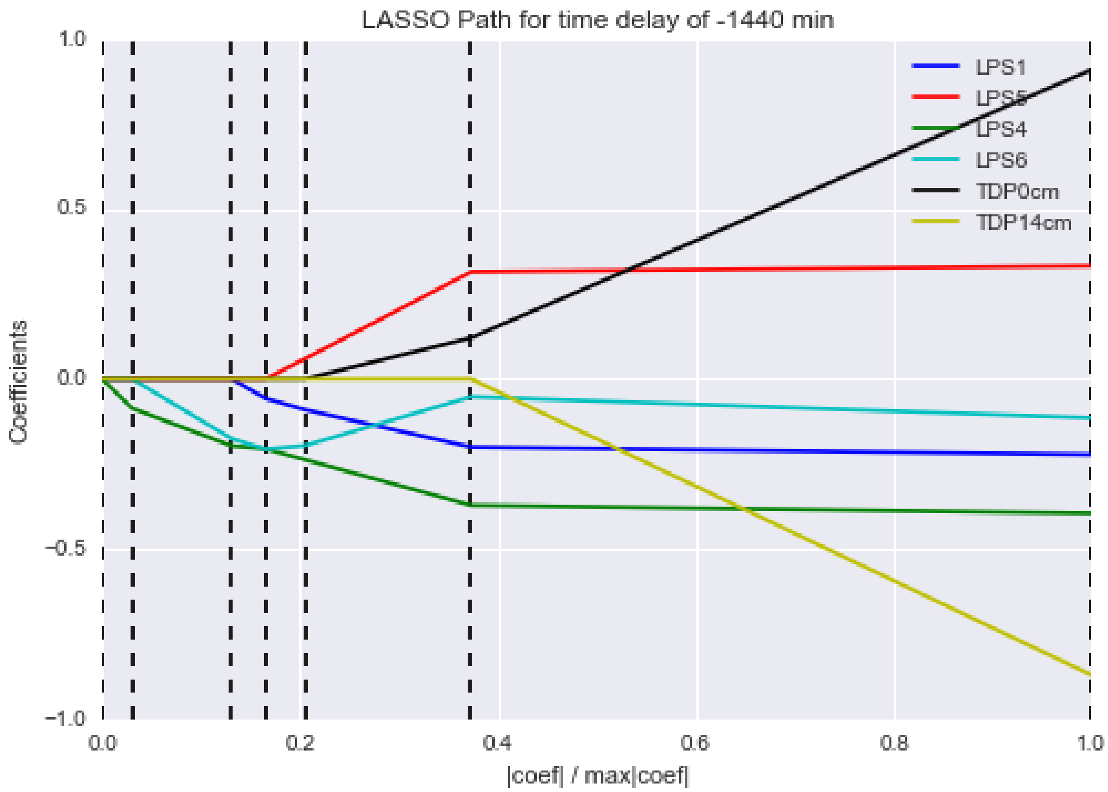
In [8]:
```
ax = pl.gca()
ax.set_color_cycle(['b', 'r', 'g', 'c', 'k', 'y', 'm'])

for i in range(6):
    ax.plot(range(-A_DAY,A_DAY), coefs[:,i], label = features[i])
pl.xlabel('Time shift in minutes')
pl.ylabel('coefficients')
pl.title('coefficients in function of shifted time')
pl.axis('tight')
pl.legend()
pl.show()
```



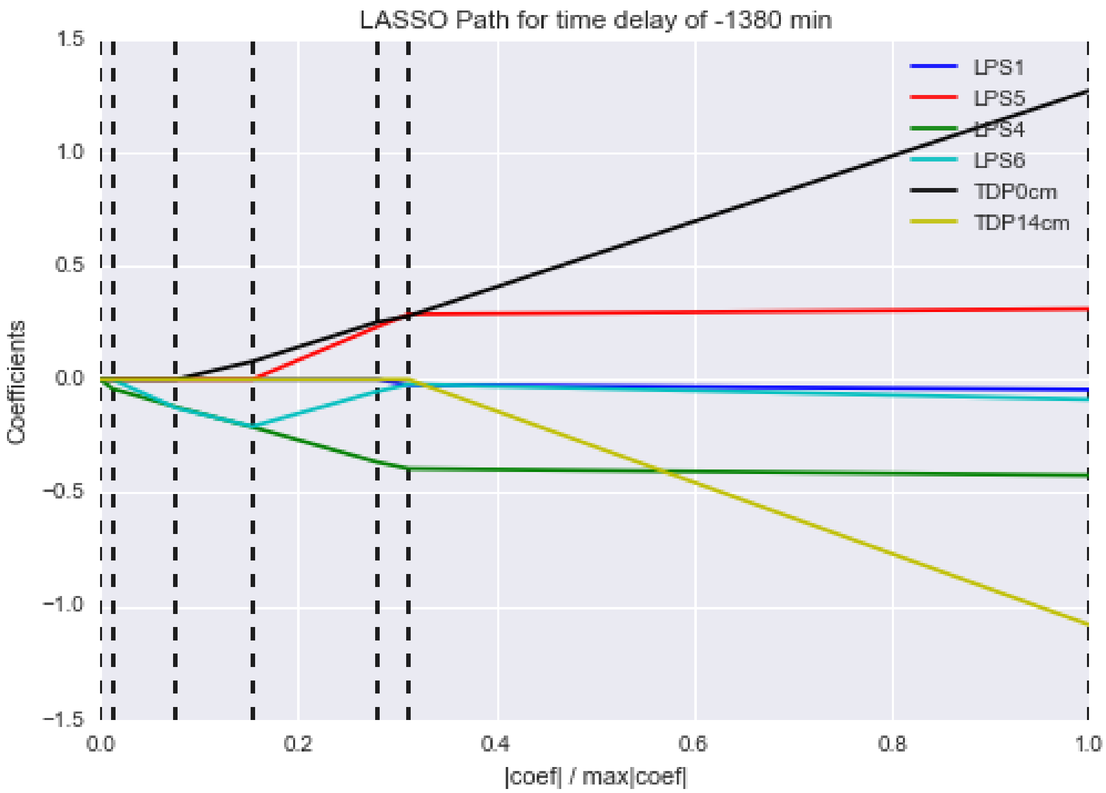

```
In [9]: # EXPERIMENT: memory versus prediction with LARS
        # train and test data
        Y_norm_train = Y_norm[A_DAY:-A_DAY-STEPS_FOR_TESTING-1]
        Y_norm_test = Y_norm[-A_DAY-STEPS_FOR_TESTING:-A_DAY-1]

        MAE_train = []
        MAE_test = []
        predictions = []
        coefs = []
        for time in range(-A_DAY,A_DAY,60):
            print time
            X_norm_train = X_norm[A_DAY-time:-A_DAY-STEPS_FOR_TESTING-1-time,:]    # -time: we go from f
            alphas, _, coefs = linear_model.lars_path(X_norm_train, Y_norm_train, method='lasso', verbos

            xx = np.sum(np.abs(coefs.T), axis=1)
            xx /= xx[-1]

            ax = pl.gca()
            ax.set_color_cycle(['b', 'r', 'g', 'c', 'k', 'y', 'm'])
            for i in range(6):
                pl.plot(xx, coefs[i].T, label = features[i])
            ymin, ymax = pl.ylim()
            pl.vlines(xx, ymin, ymax, linestyle='dashed')
            pl.xlabel('|coef| / max|coef|')
            pl.ylabel('Coefficients')
```

```
        pl.title('LASSO Path for time delay of %s min'%time)
        pl.axis('tight')
        pl.legend()
        pl.show()
```

```
-1440
.

-c:12: DeprecationWarning: using a non-integer number instead of an integer will result in an error in t
```

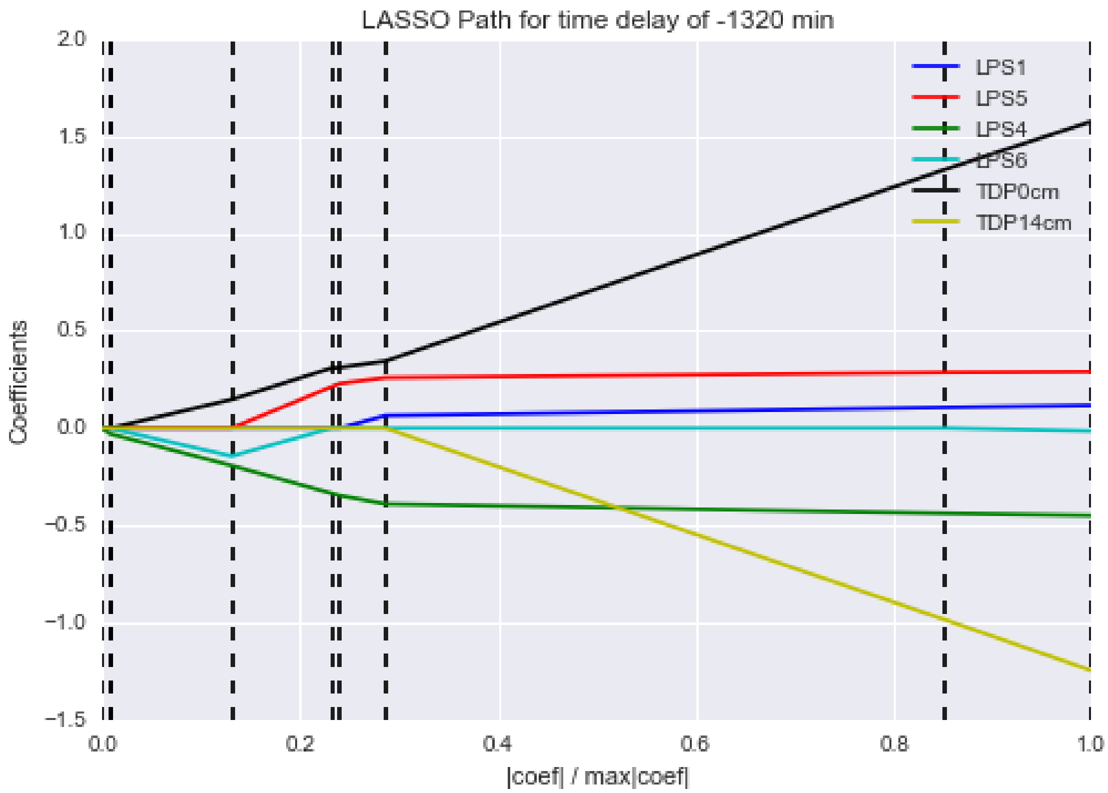



```
-1380
.
```


LASSO Path for time delay of -1380 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

−1320

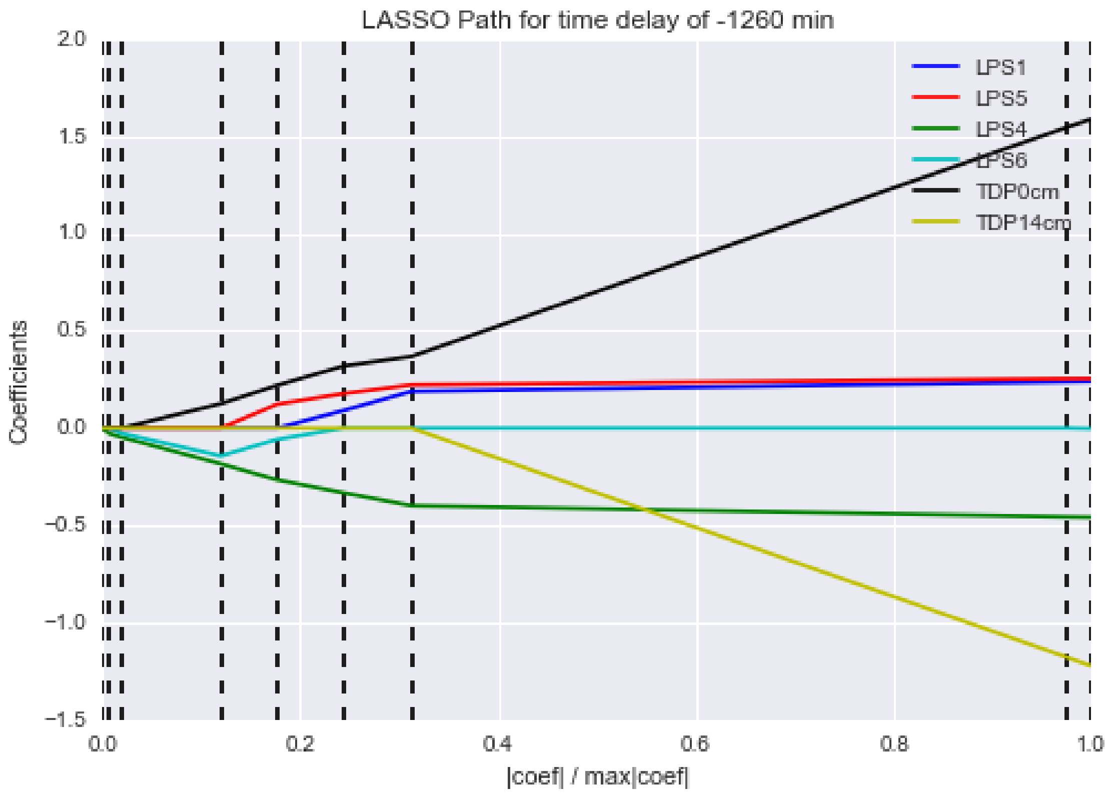
.



LASSO Path for time delay of -1320 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

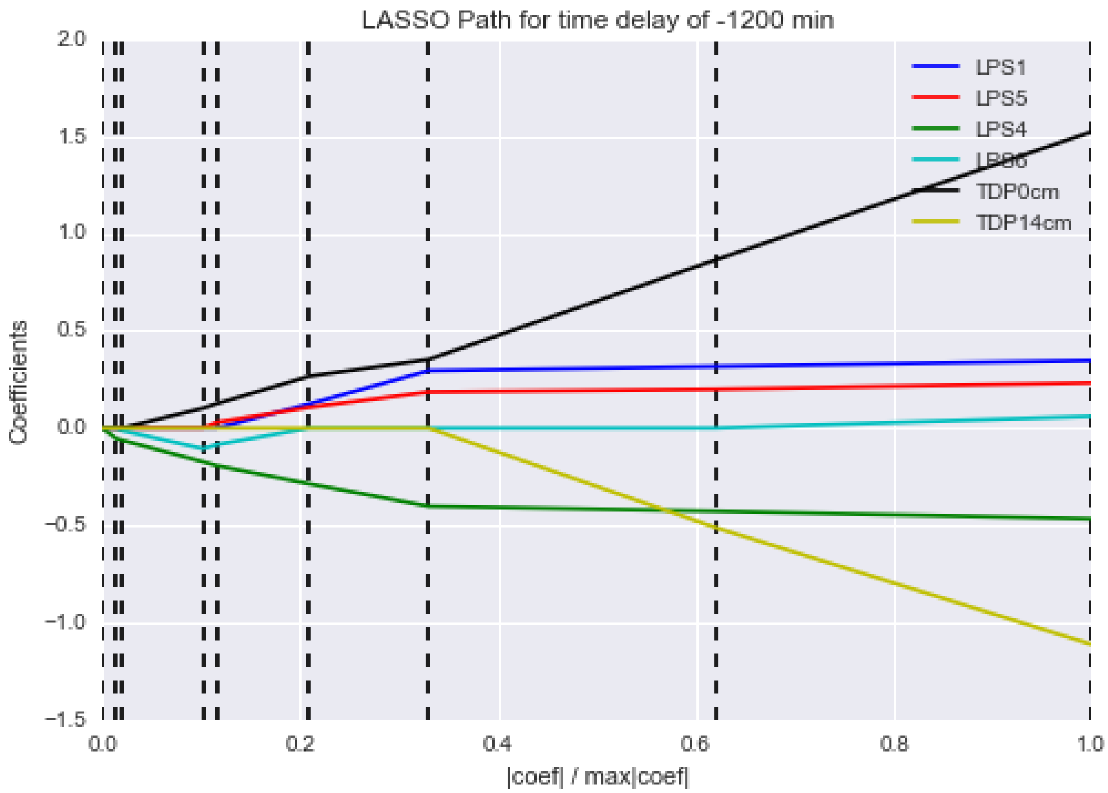
−1260

.


LASSO Path for time delay of -1260 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

−1200

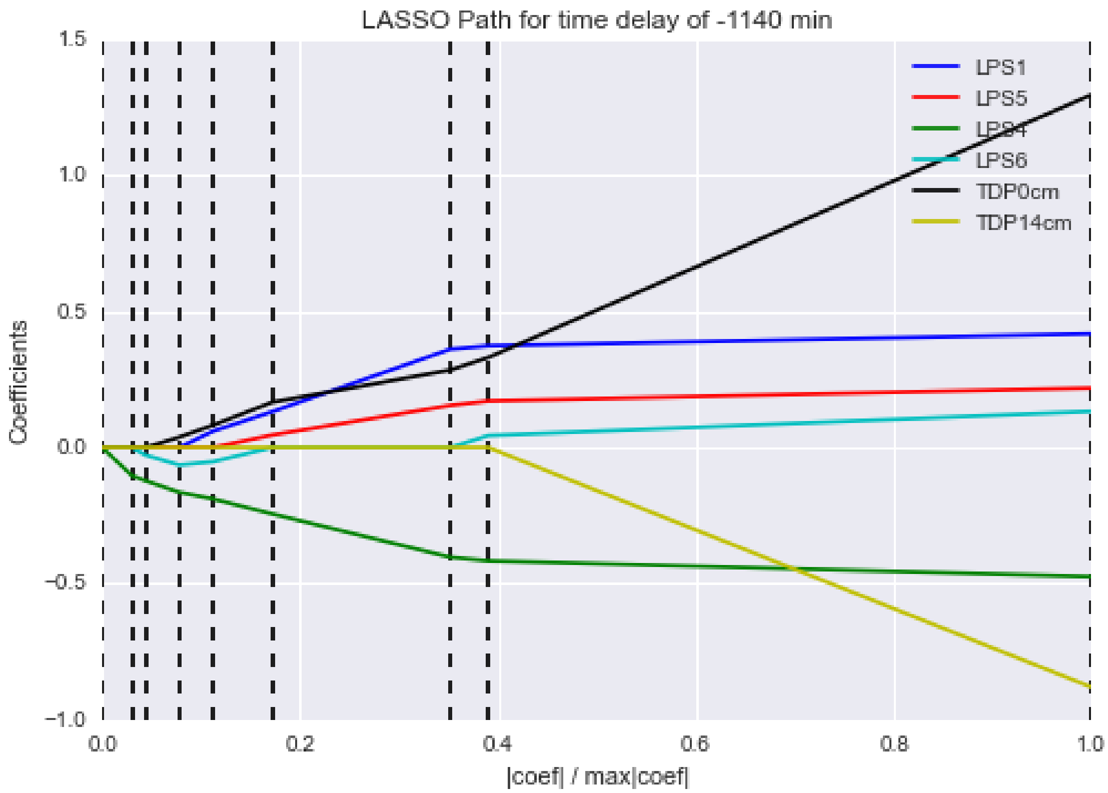
.



LASSO Path for time delay of -1200 min
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

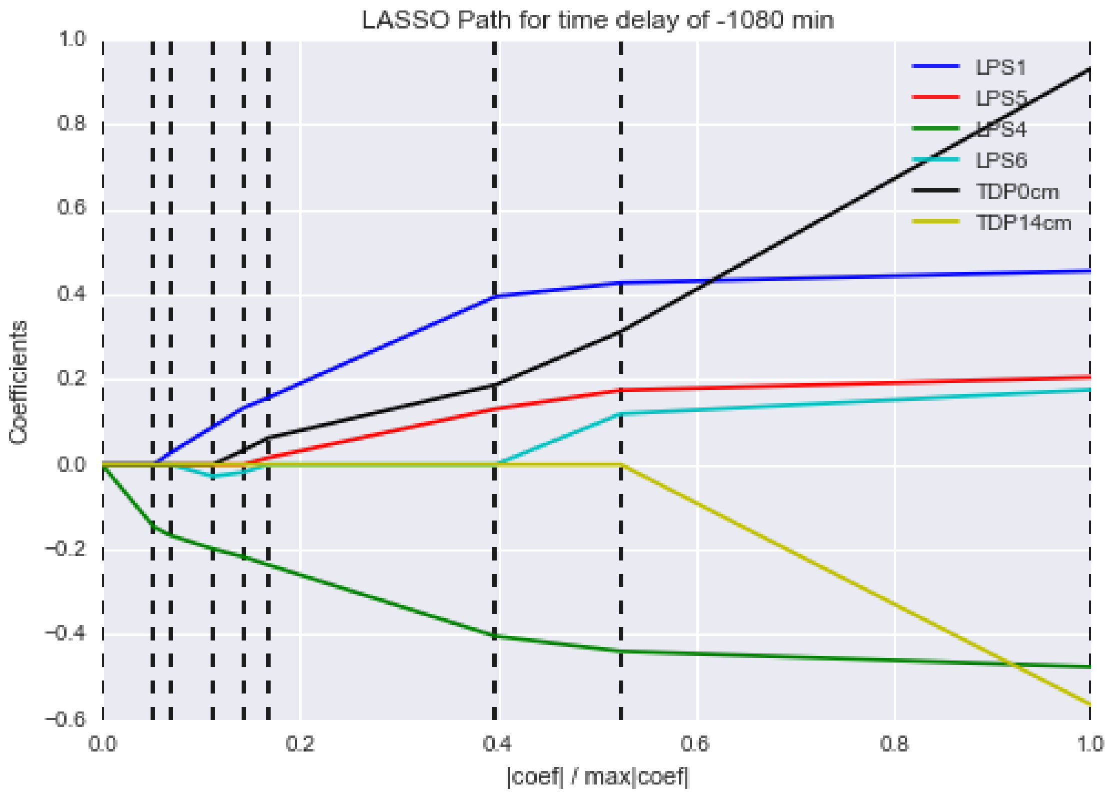
−1140

.



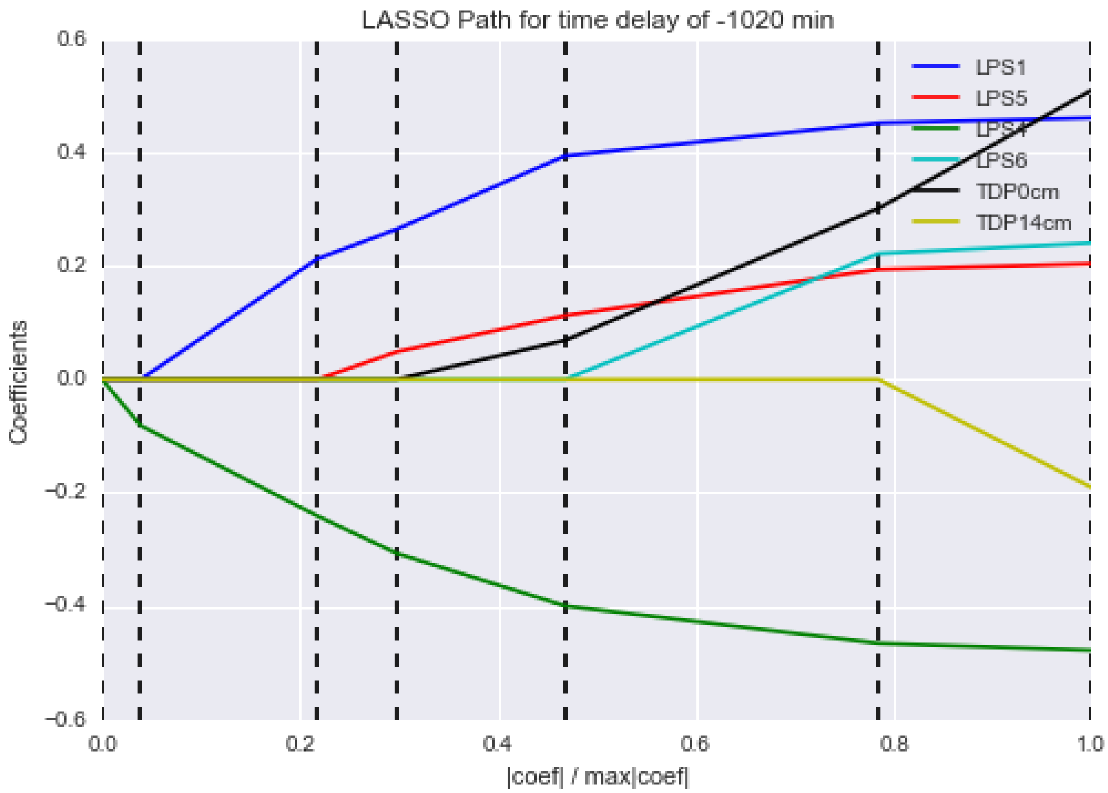
LASSO Path for time delay of -1140 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

−1080
.


LASSO Path for time delay of -1080 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0

−1020

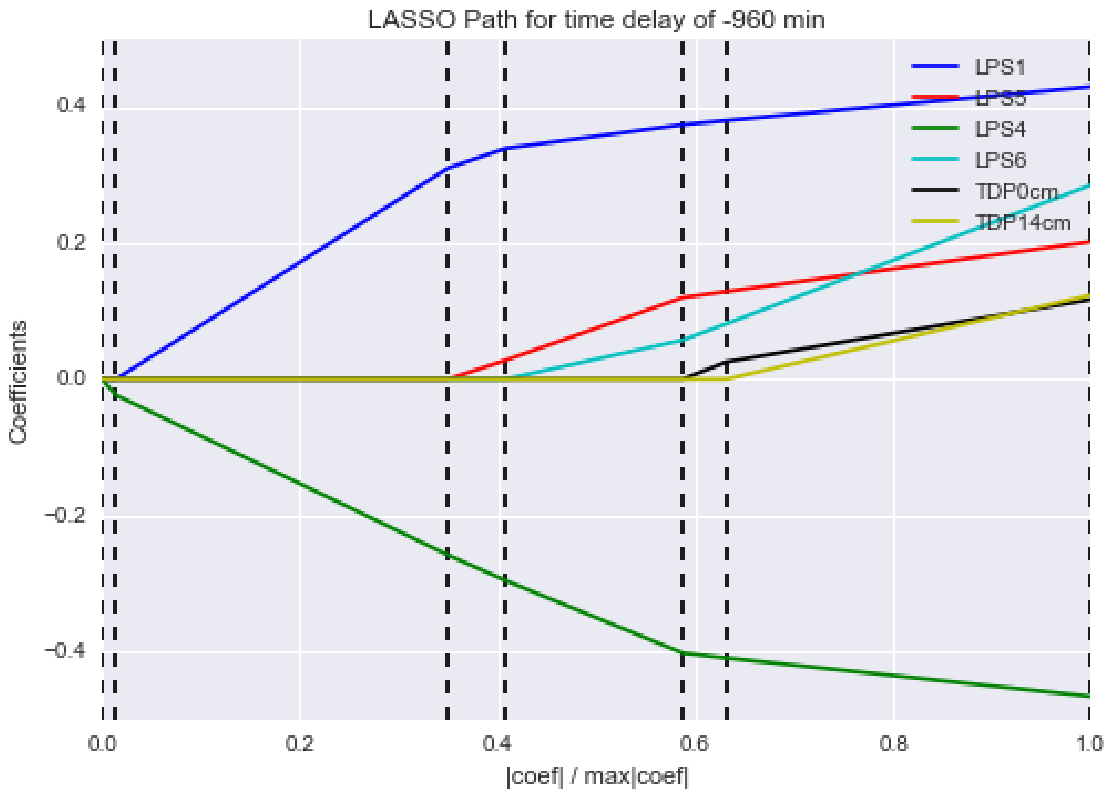
.



LASSO Path for time delay of -1020 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0

−960

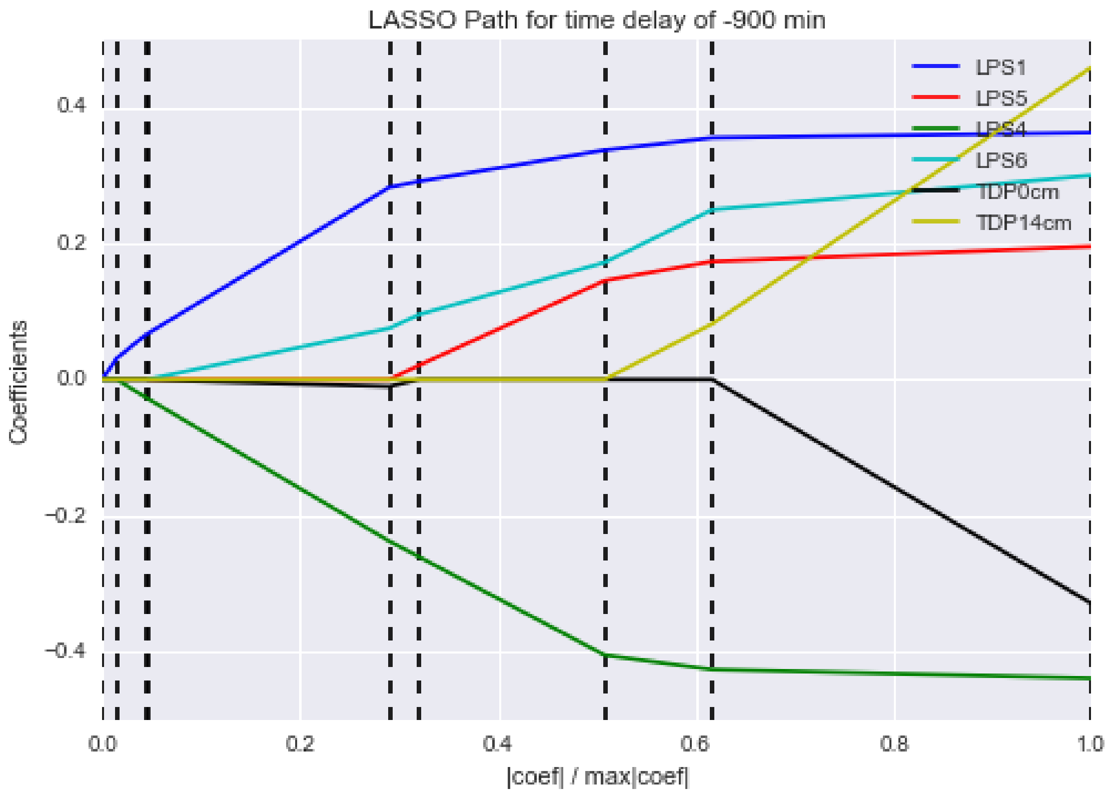
.



LASSO Path for time delay of -960 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
0.4
0.2
0.0
−0.2
−0.4
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|

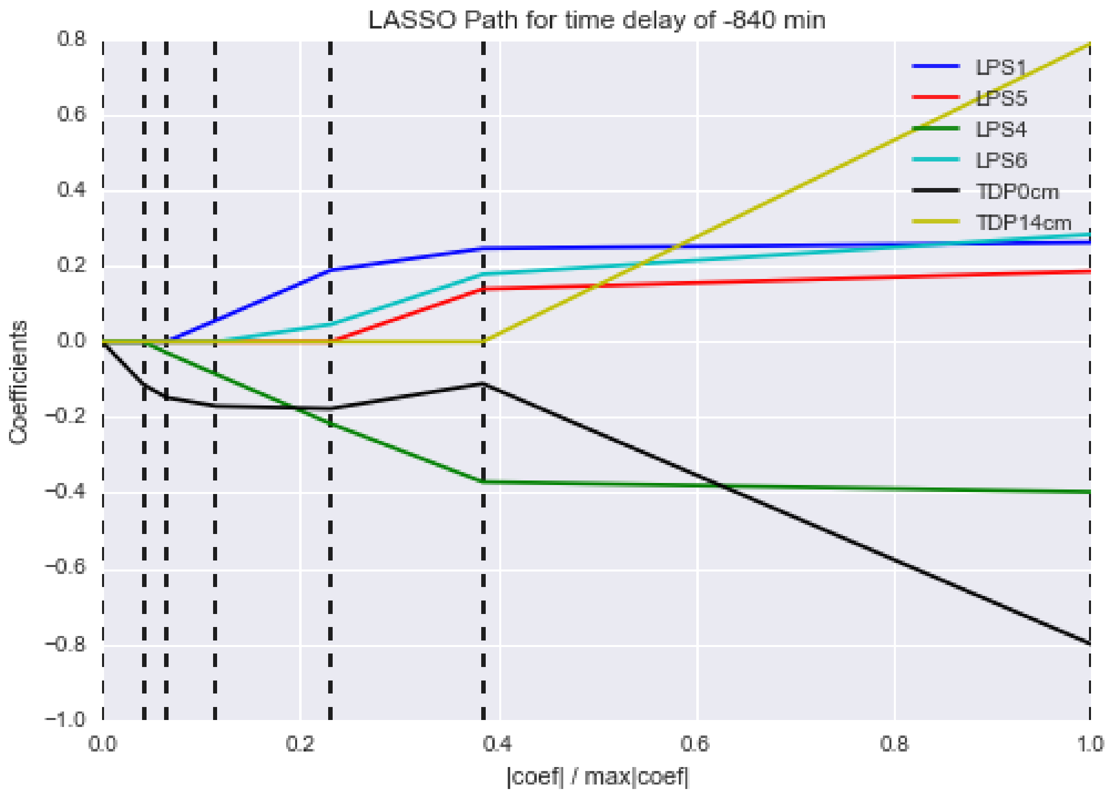
−900

.


LASSO Path for time delay of -900 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.4
0.2
0.0
−0.2
−0.4
0.0
0.2
0.4
0.6
0.8
1.0

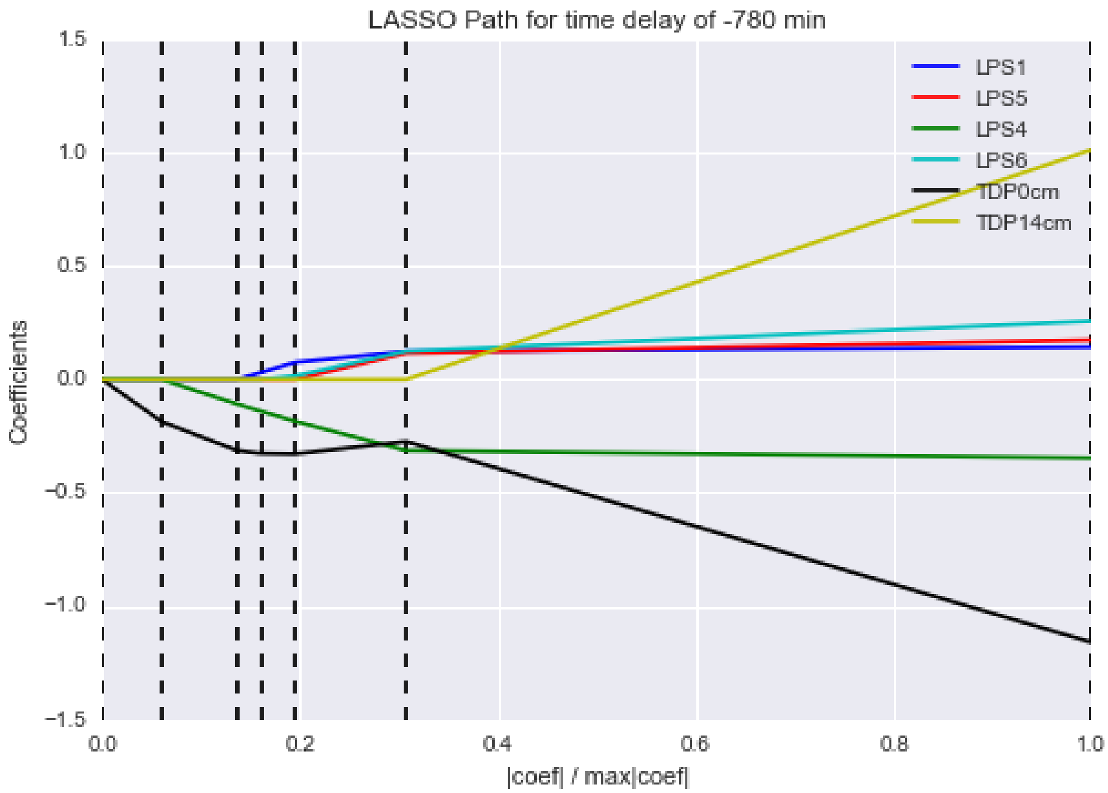
−840

.


LASSO Path for time delay of -840 min
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

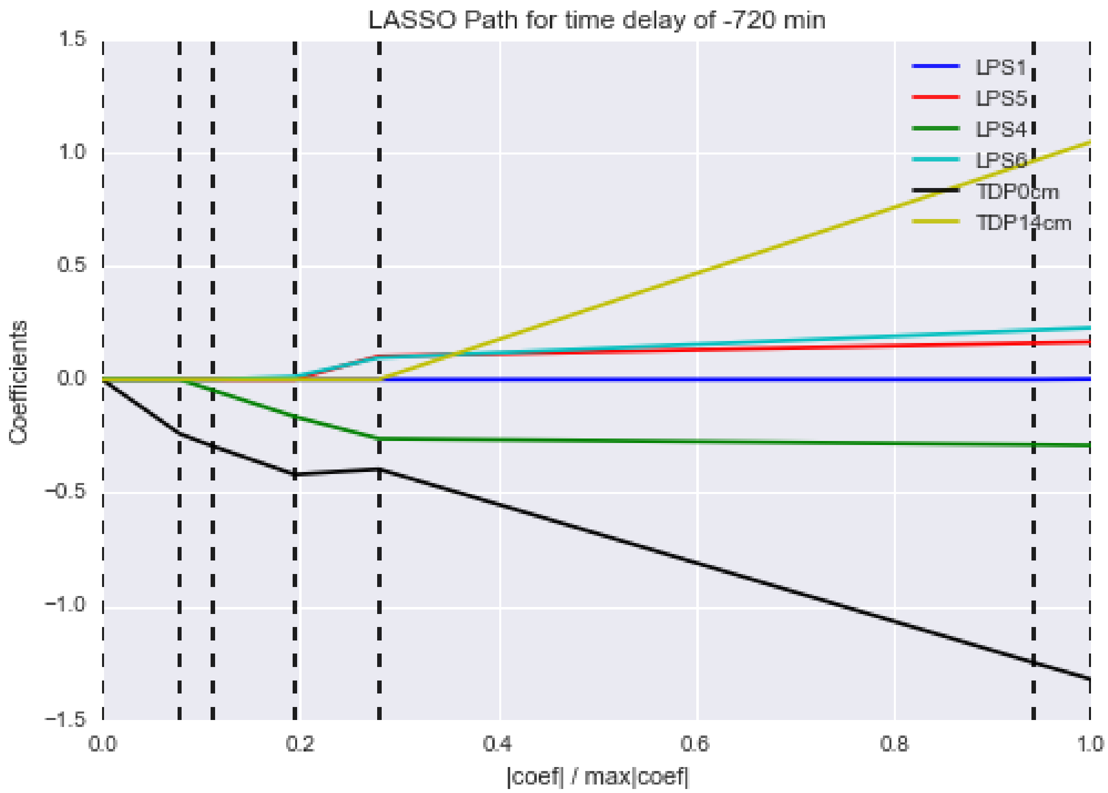
−780

.

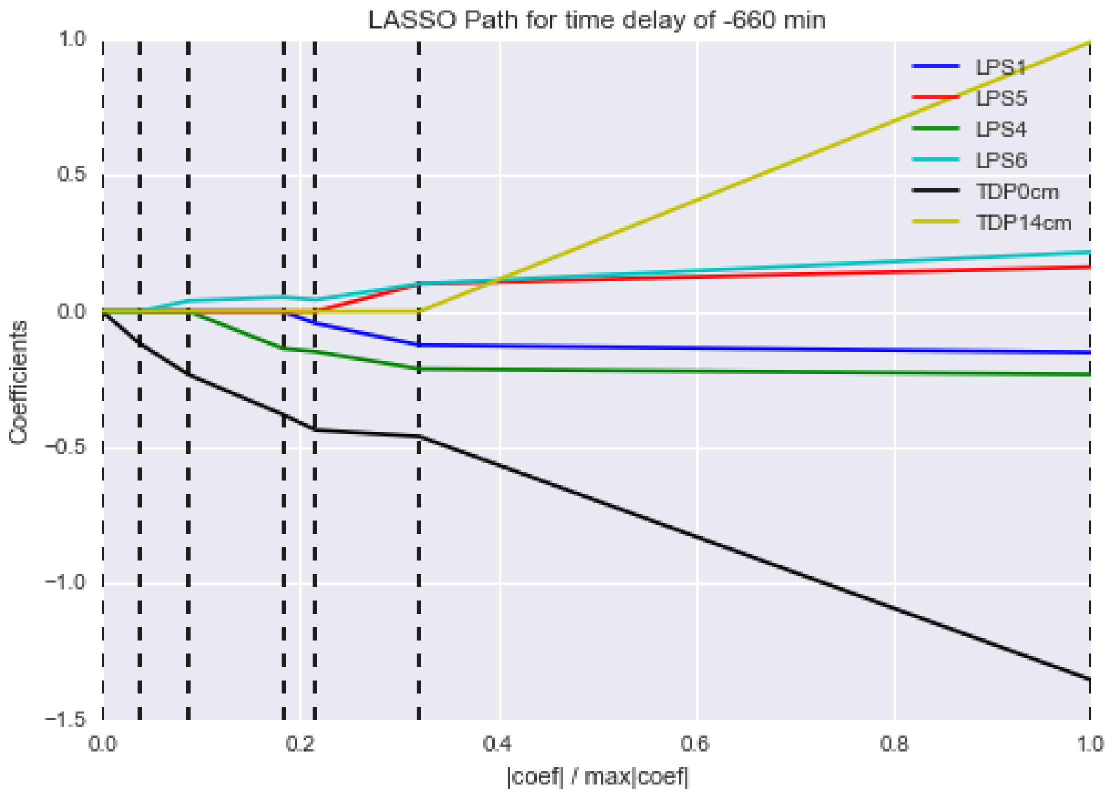

LASSO Path for time delay of -780 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

−720

.

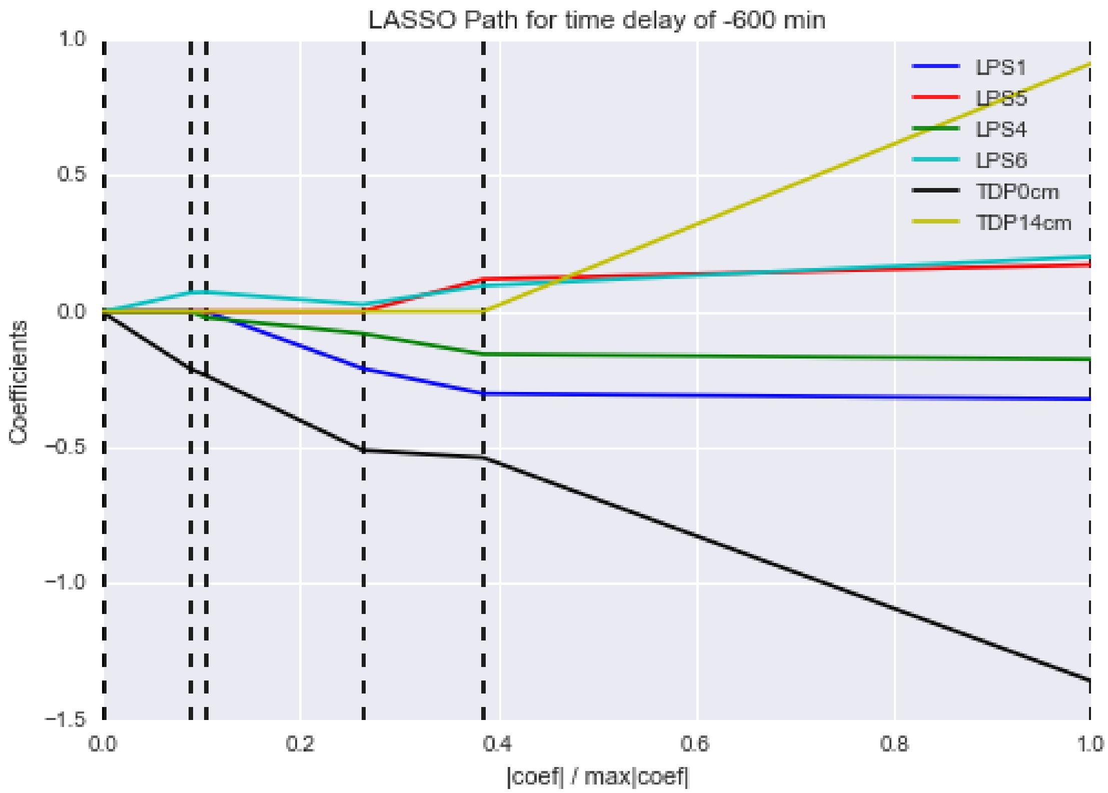

LASSO Path for time delay of -720 min
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

−660
.

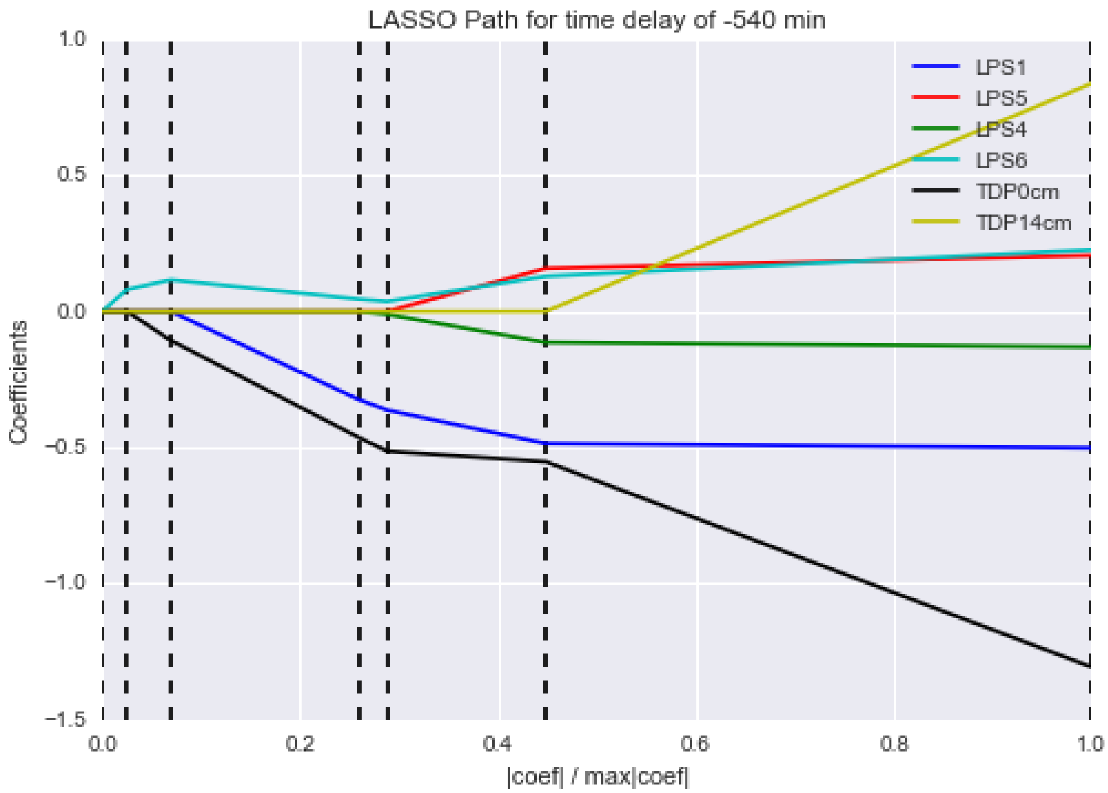

LASSO Path for time delay of -660 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

−600

.

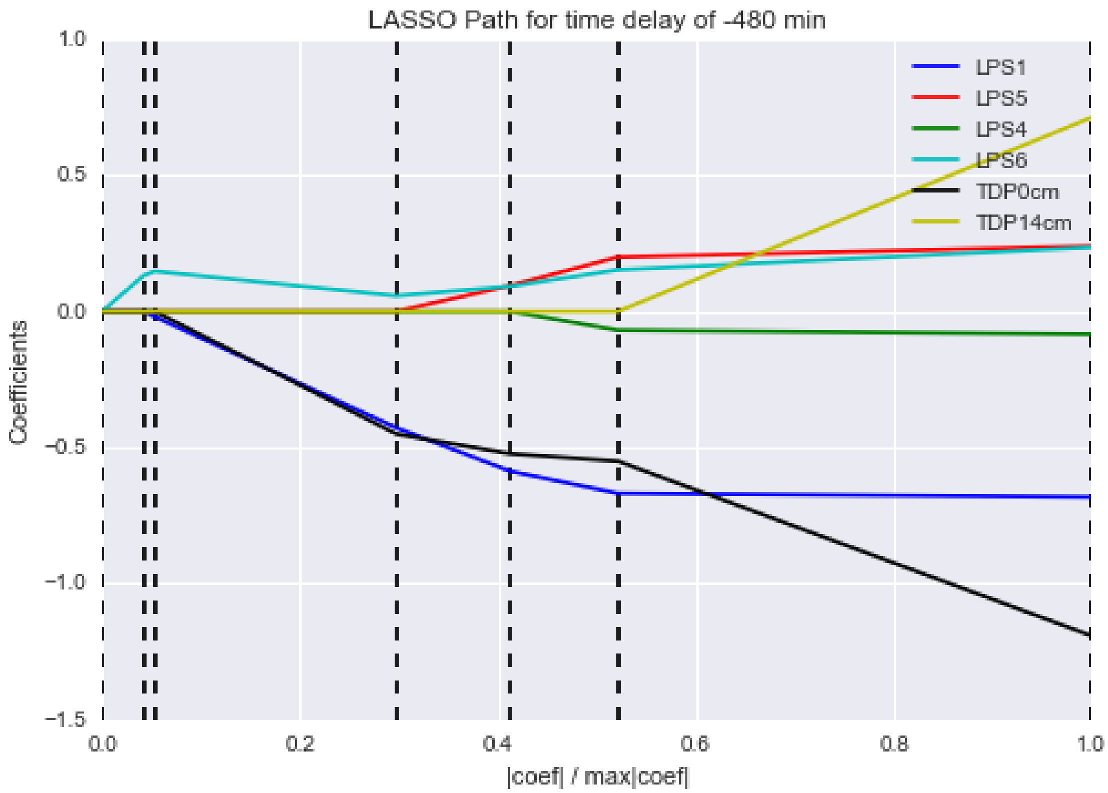

LASSO Path for time delay of -600 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

−540
.


LASSO Path for time delay of -540 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

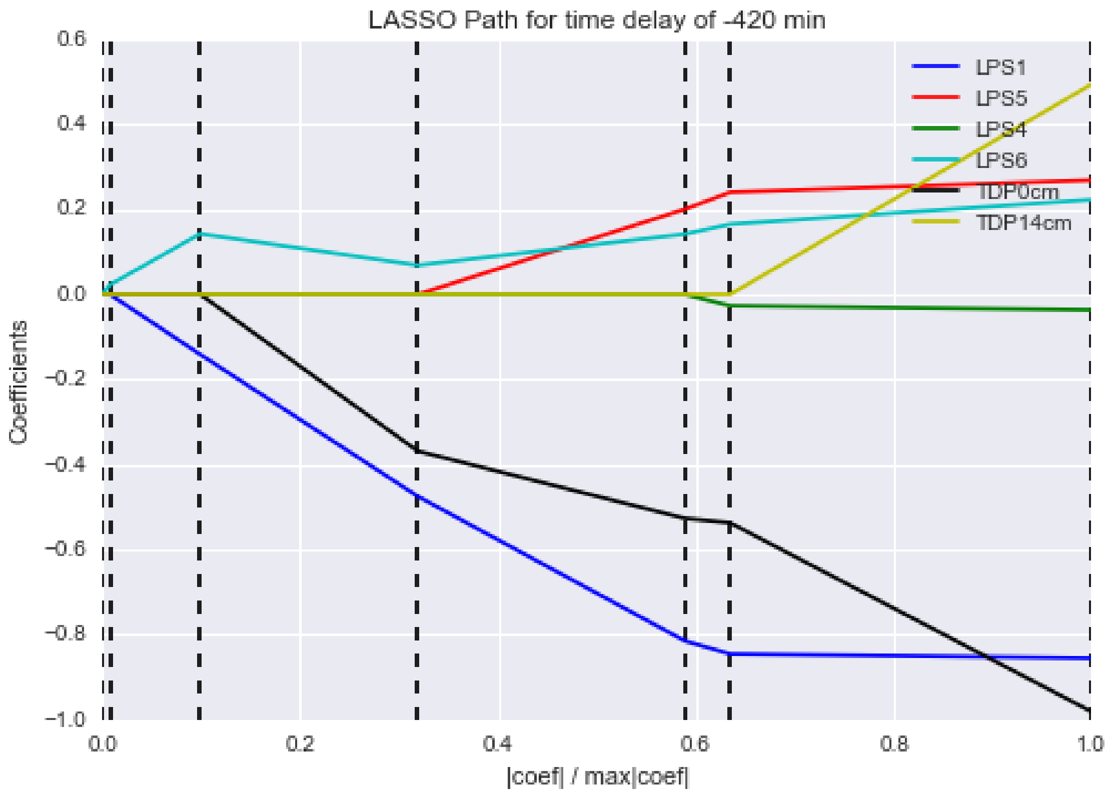
−480

.


LASSO Path for time delay of -480 min
1.0
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

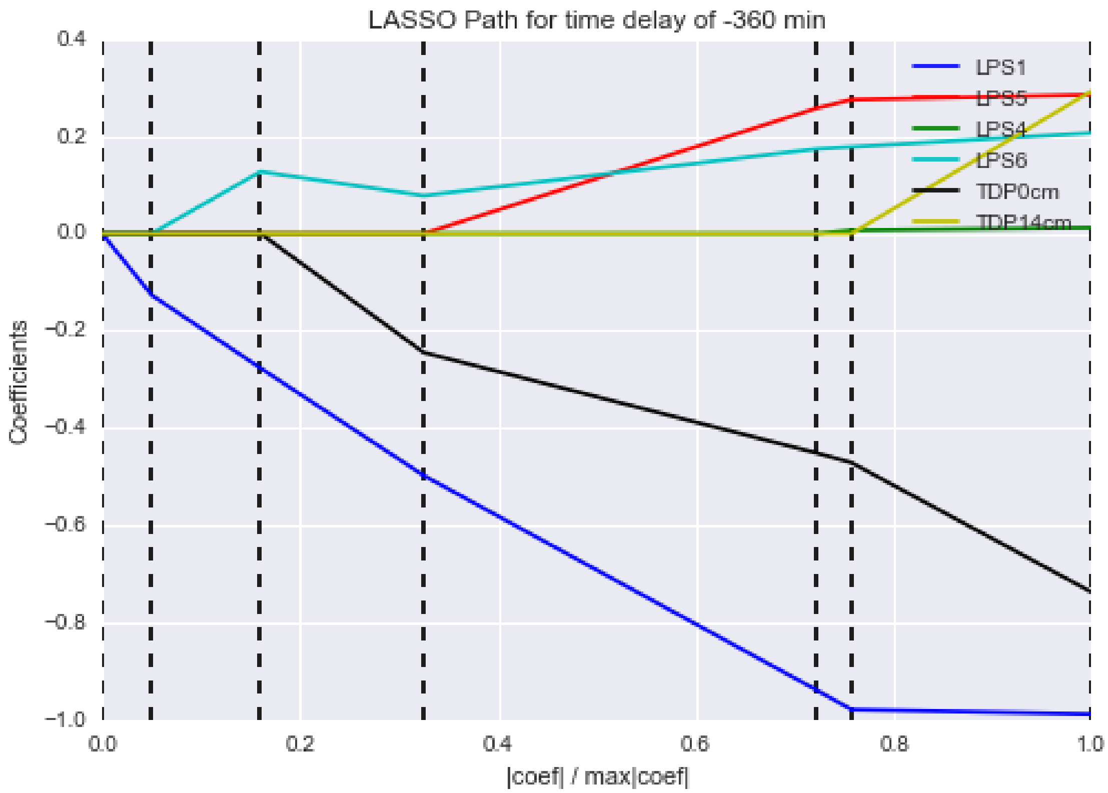
−420

.

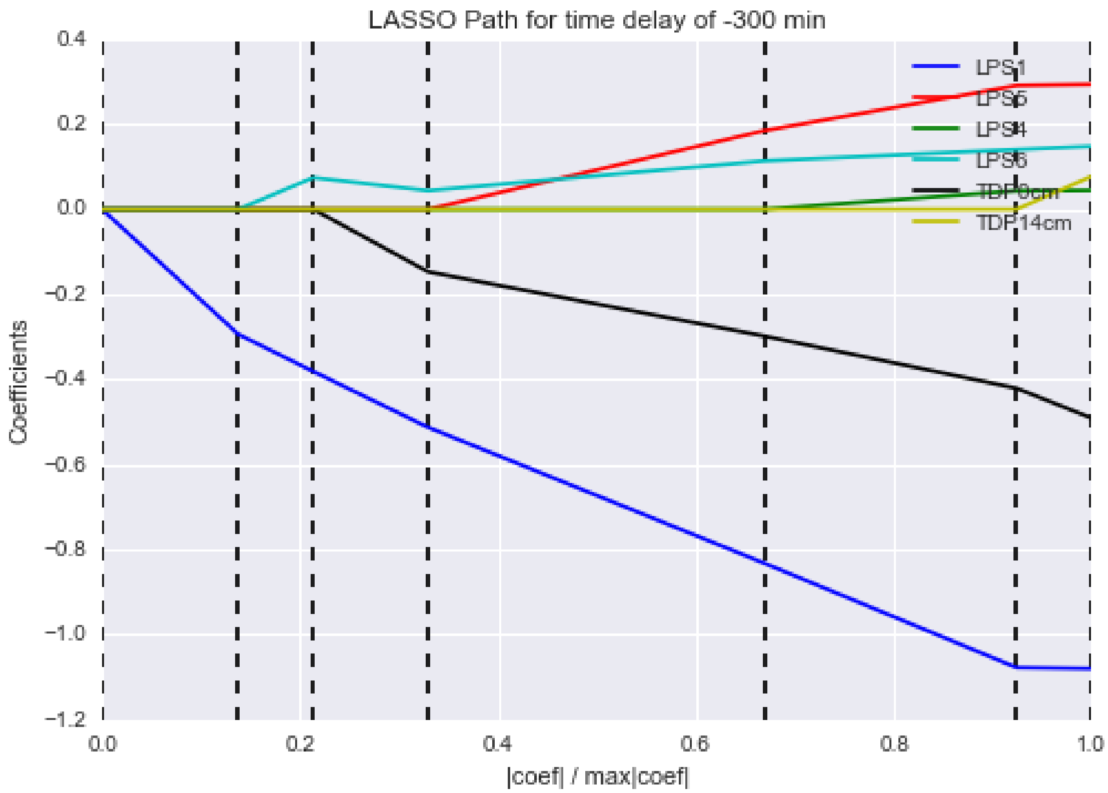

LASSO Path for time delay of -420 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

−360

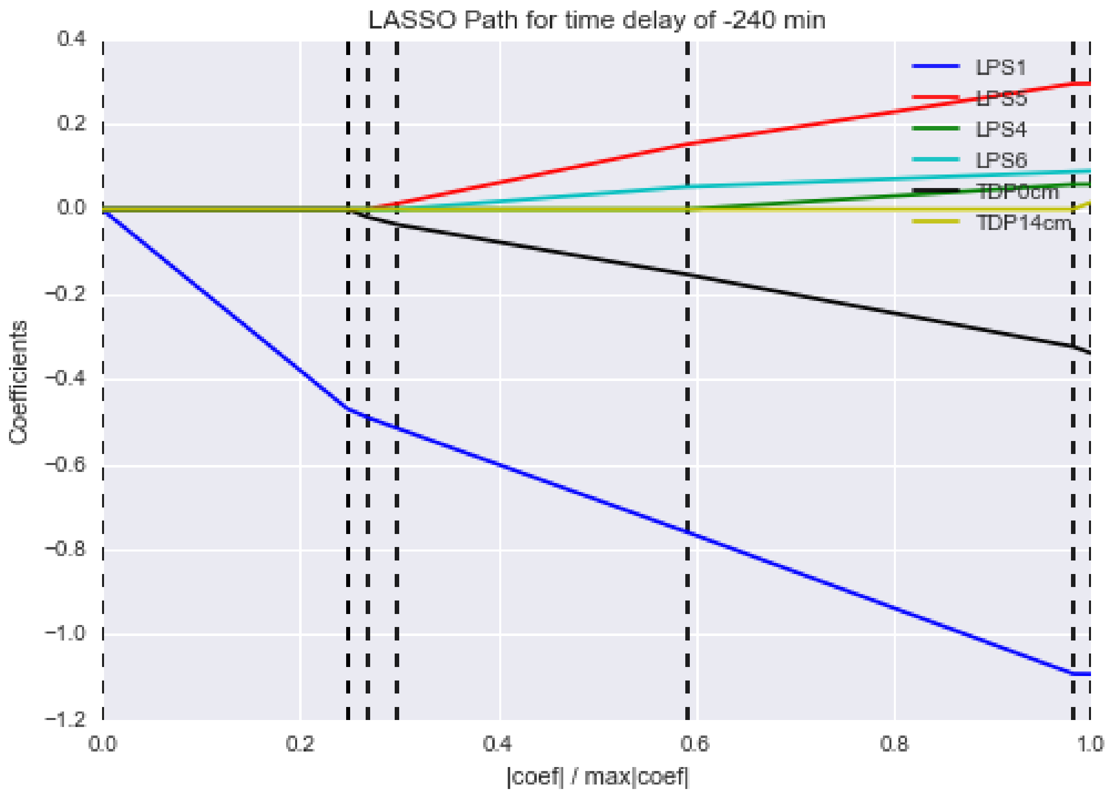
.


LASSO Path for time delay of -360 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

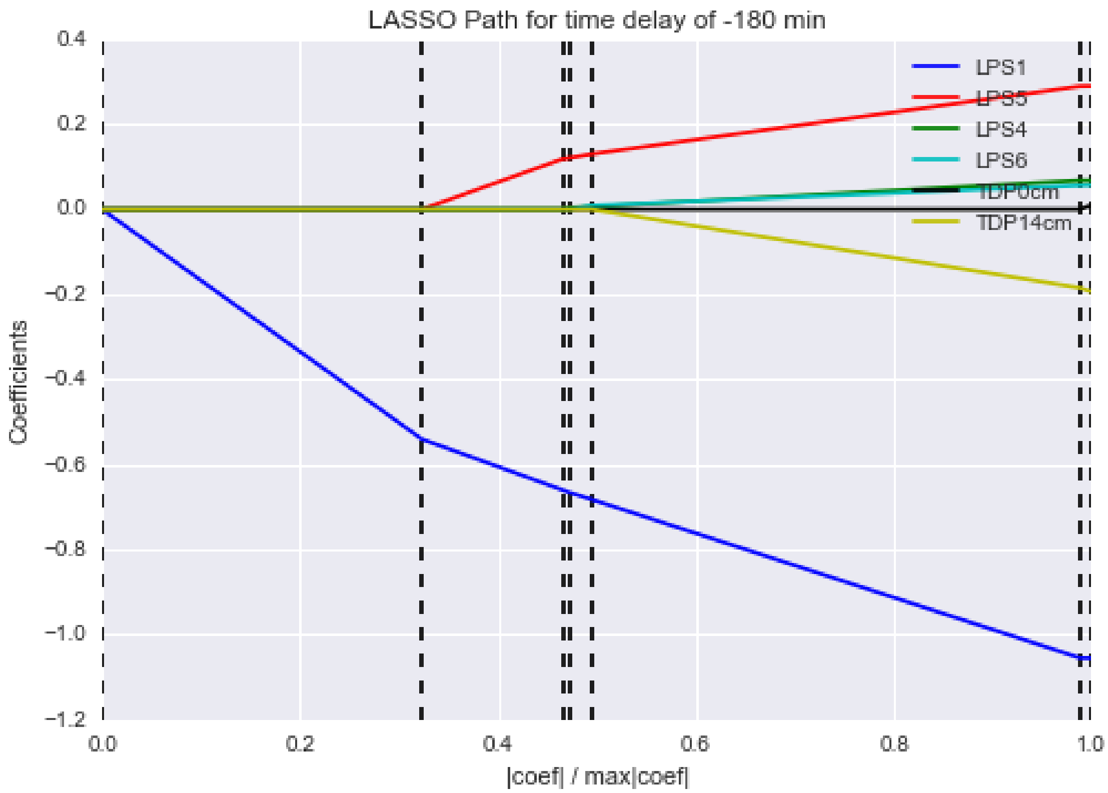
−300
.


LASSO Path for time delay of -300 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP8cm
TDP14cm
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
0.0
0.2
0.4
0.6
0.8
1.0

−240
.

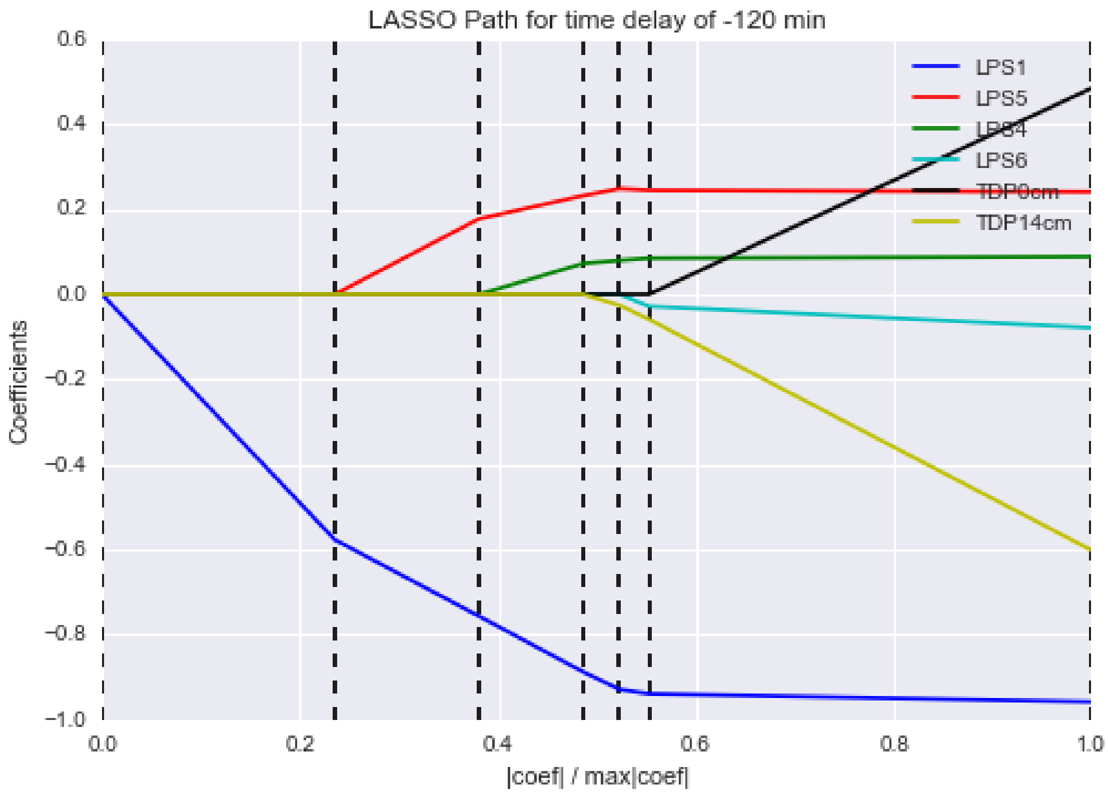

LASSO Path for time delay of -240 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
0.0
0.2
0.4
0.6
0.8
1.0

−180
.


LASSO Path for time delay of -180 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
0.0
0.2
0.4
0.6
0.8
1.0

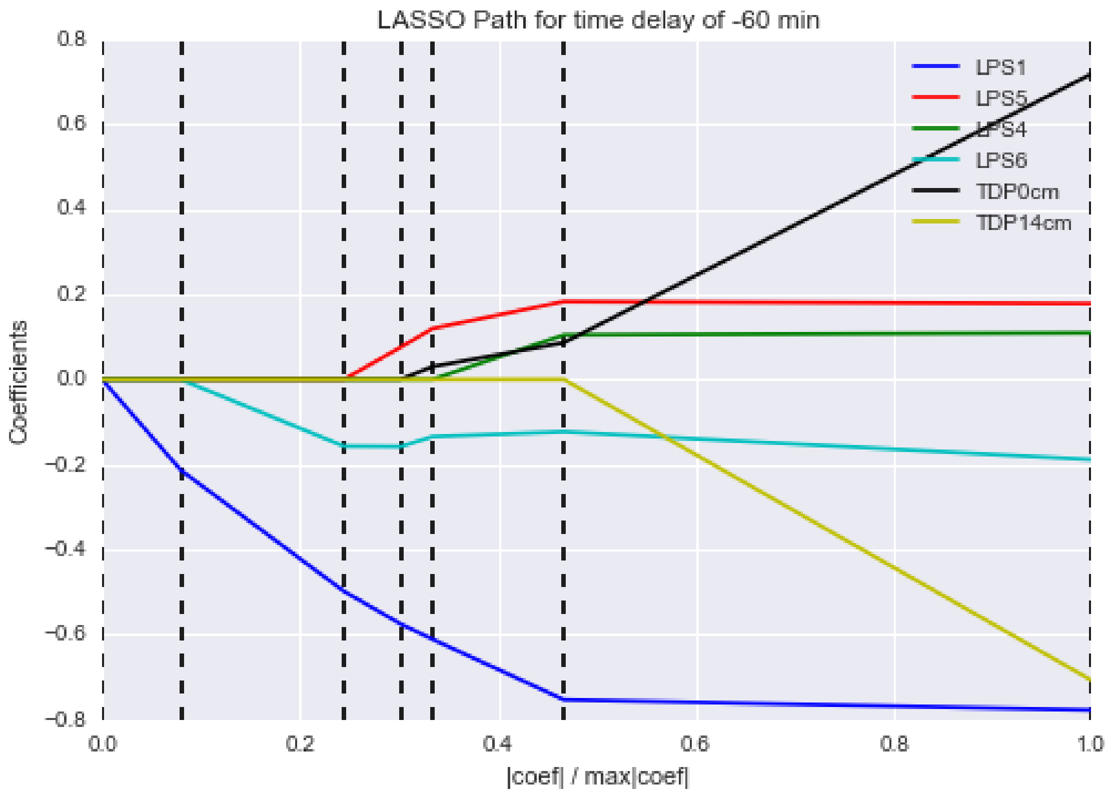
−120

.



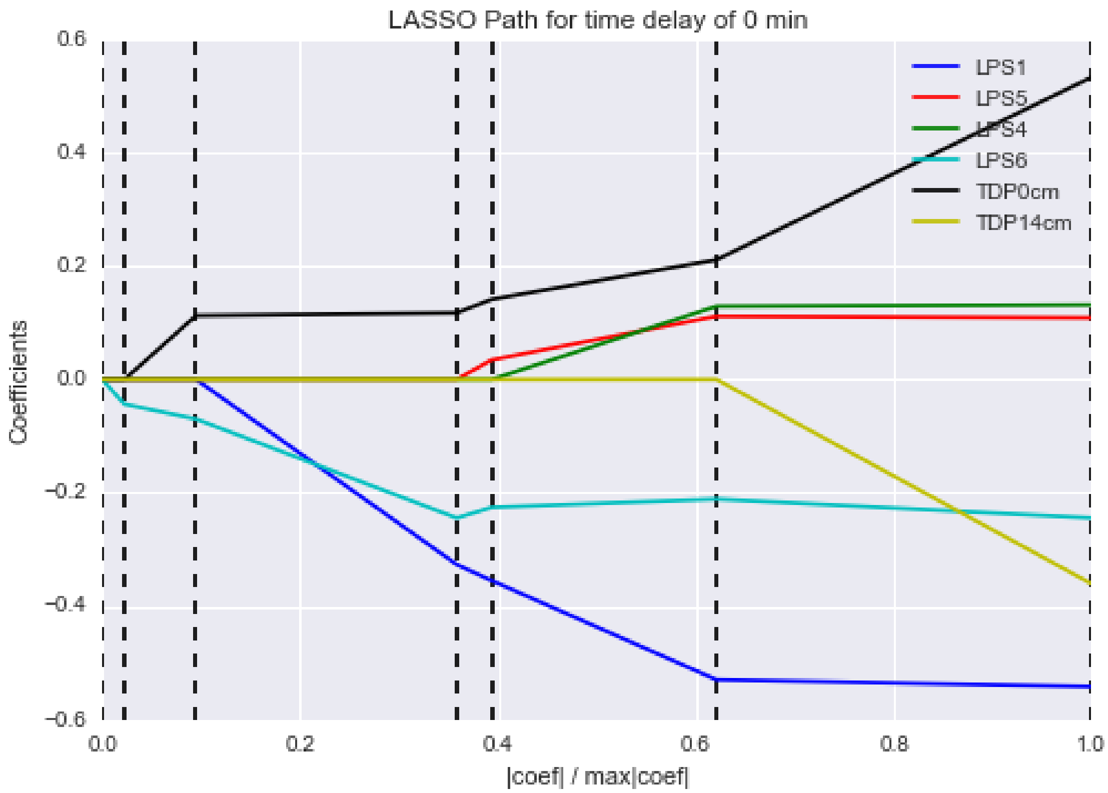
LASSO Path for time delay of -120 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

−60
.



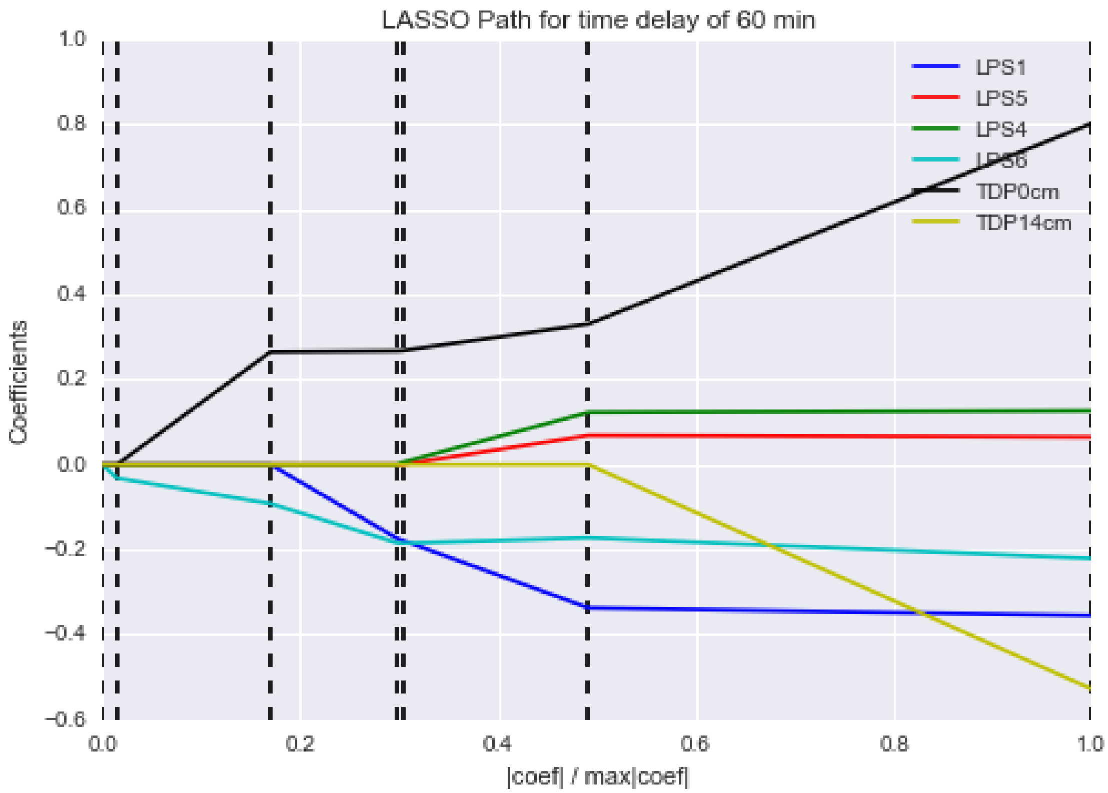
LASSO Path for time delay of -60 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.0
0.2
0.4
0.6
0.8
1.0

0
.

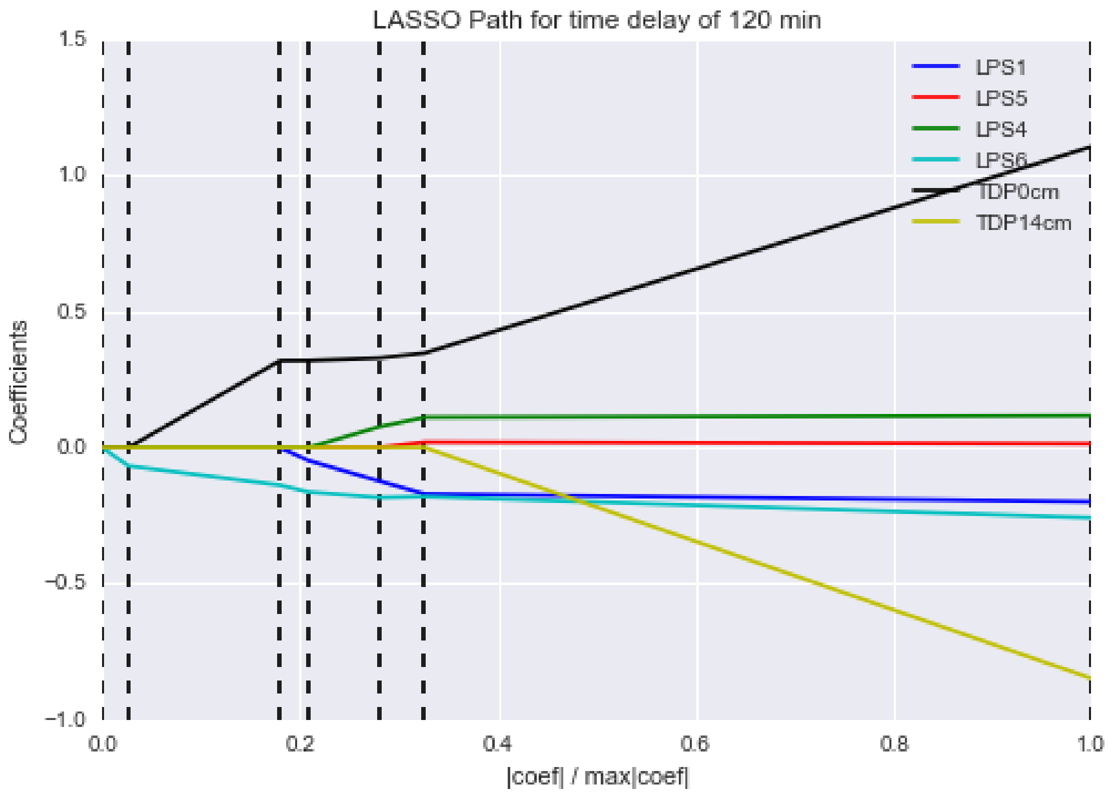


LASSO Path for time delay of 0 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0

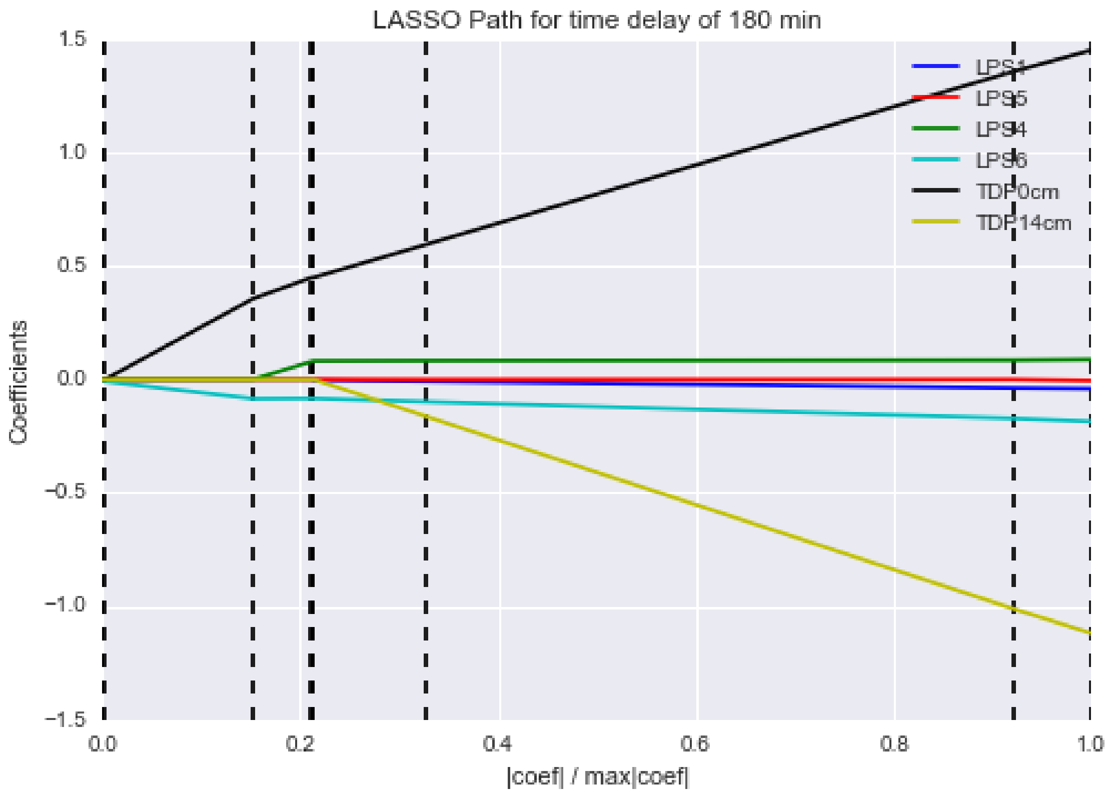
60
.


LASSO Path for time delay of 60 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|

120
.

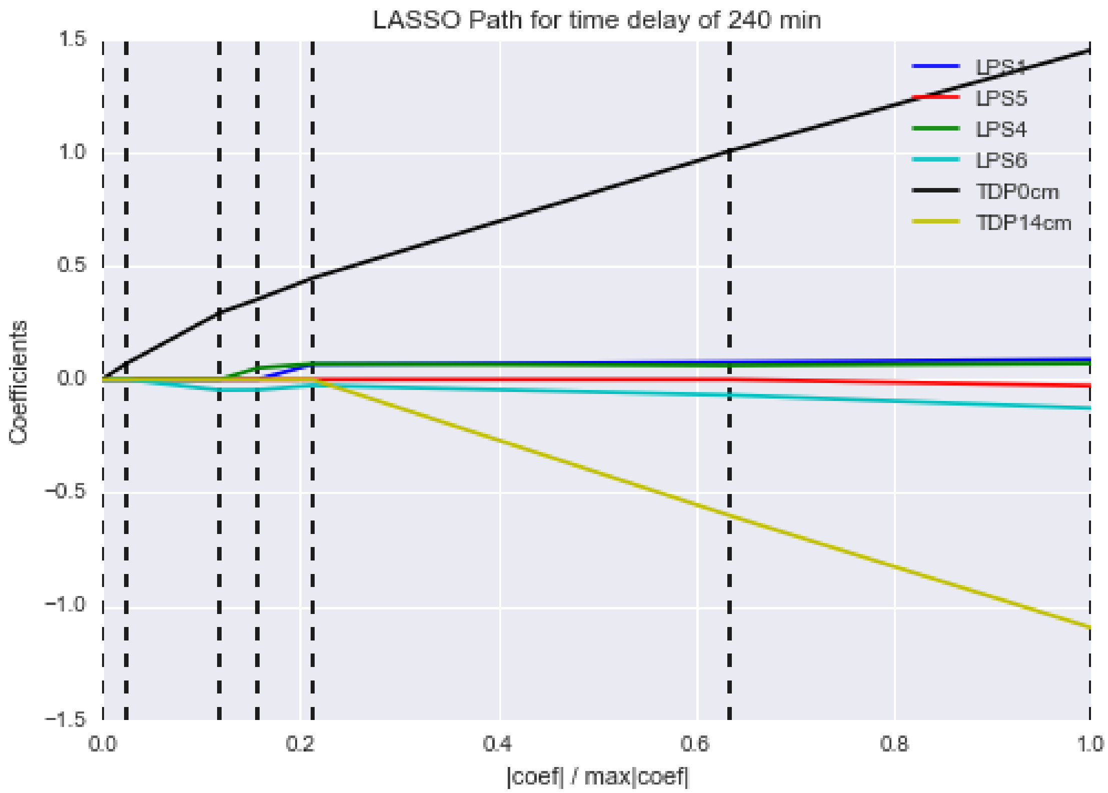

LASSO Path for time delay of 120 min
1.5
1.0
0.5
0.0
−0.5
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

180
.

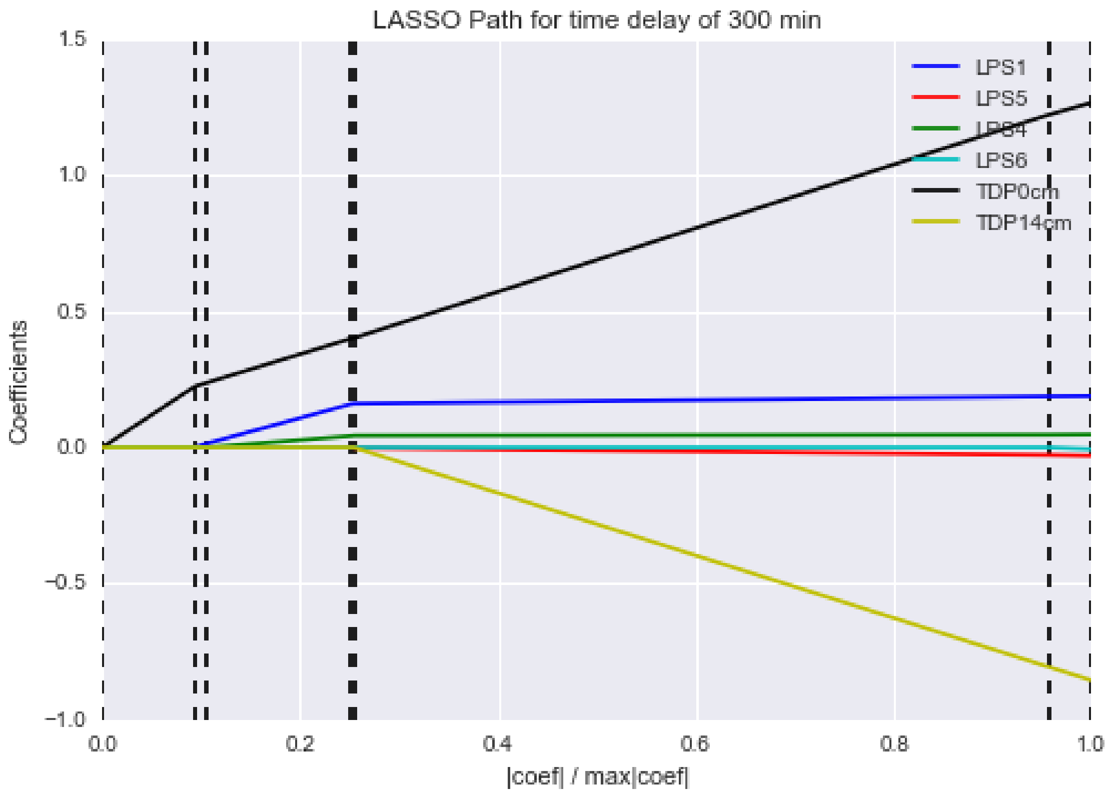

LASSO Path for time delay of 180 min
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

240

.

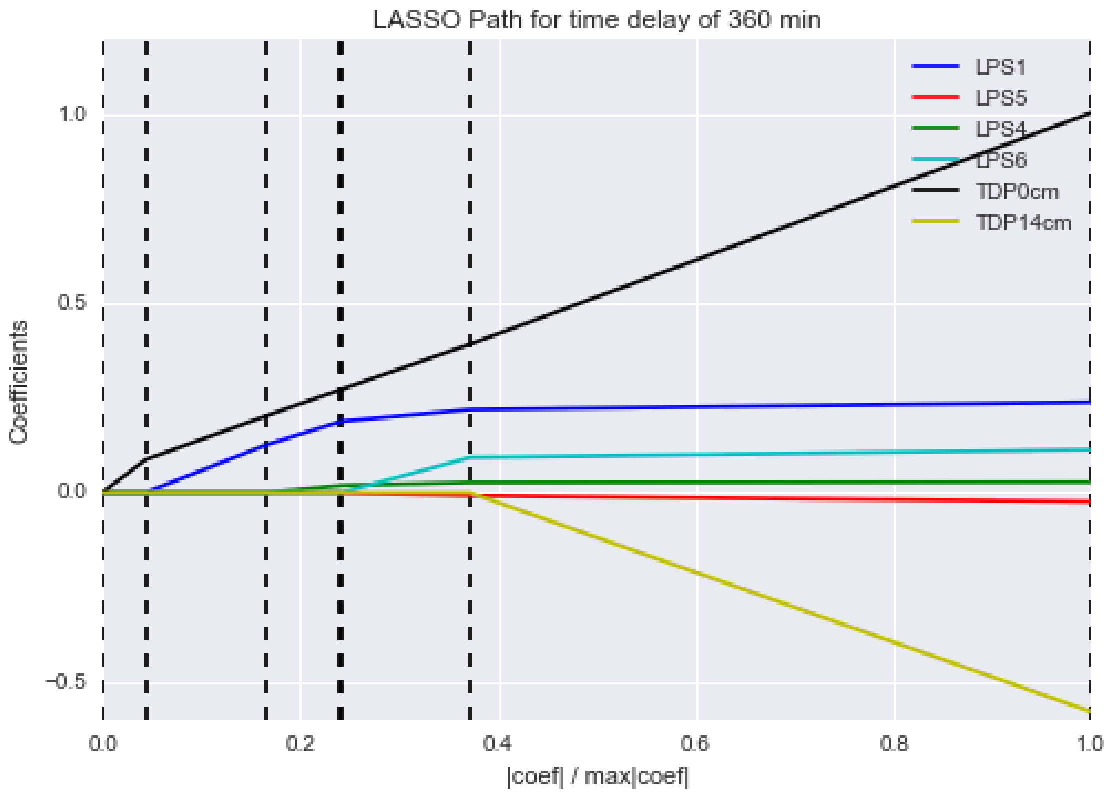

LASSO Path for time delay of 240 min
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

300
.


LASSO Path for time delay of 300 min
1.5
1.0
0.5
0.0
−0.5
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

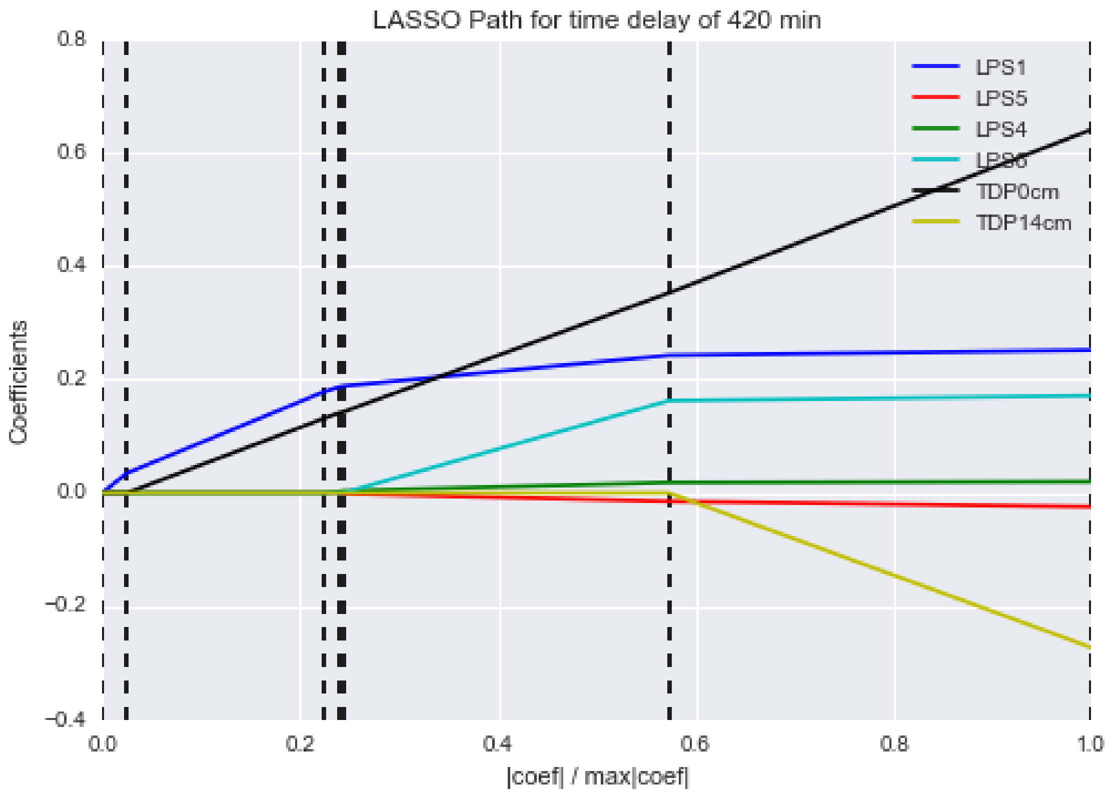
360

.

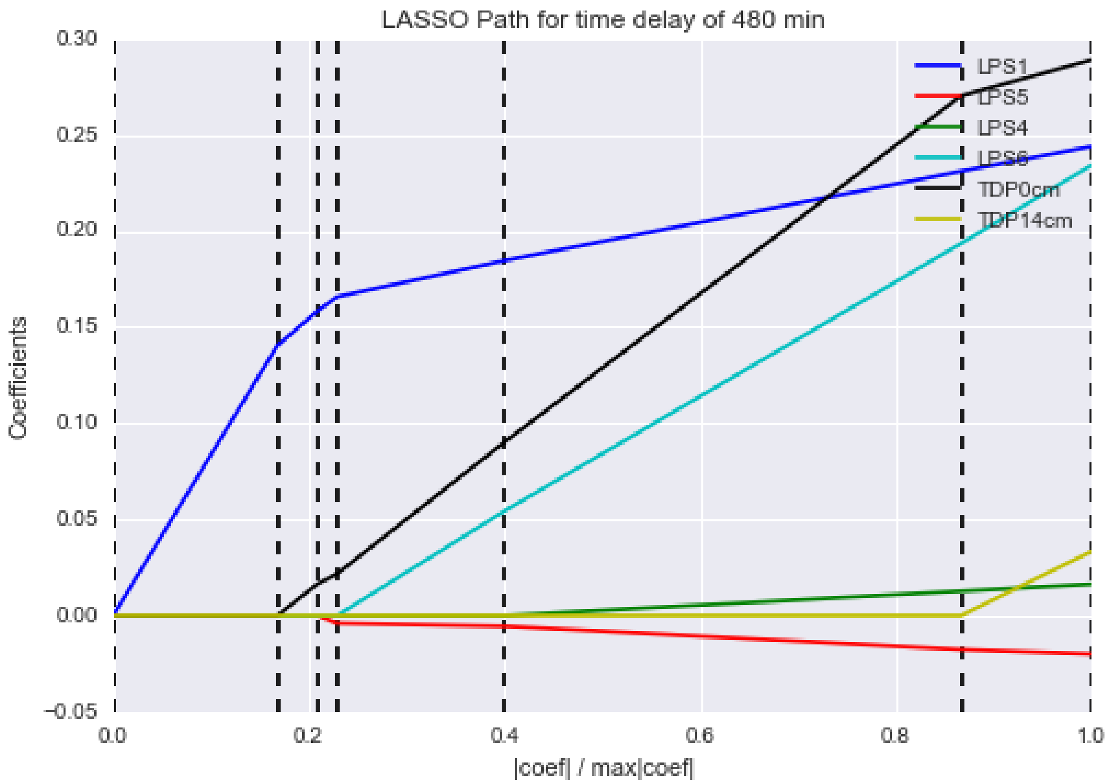


LASSO Path for time delay of 360 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.5
0.0
−0.5
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|

420
.

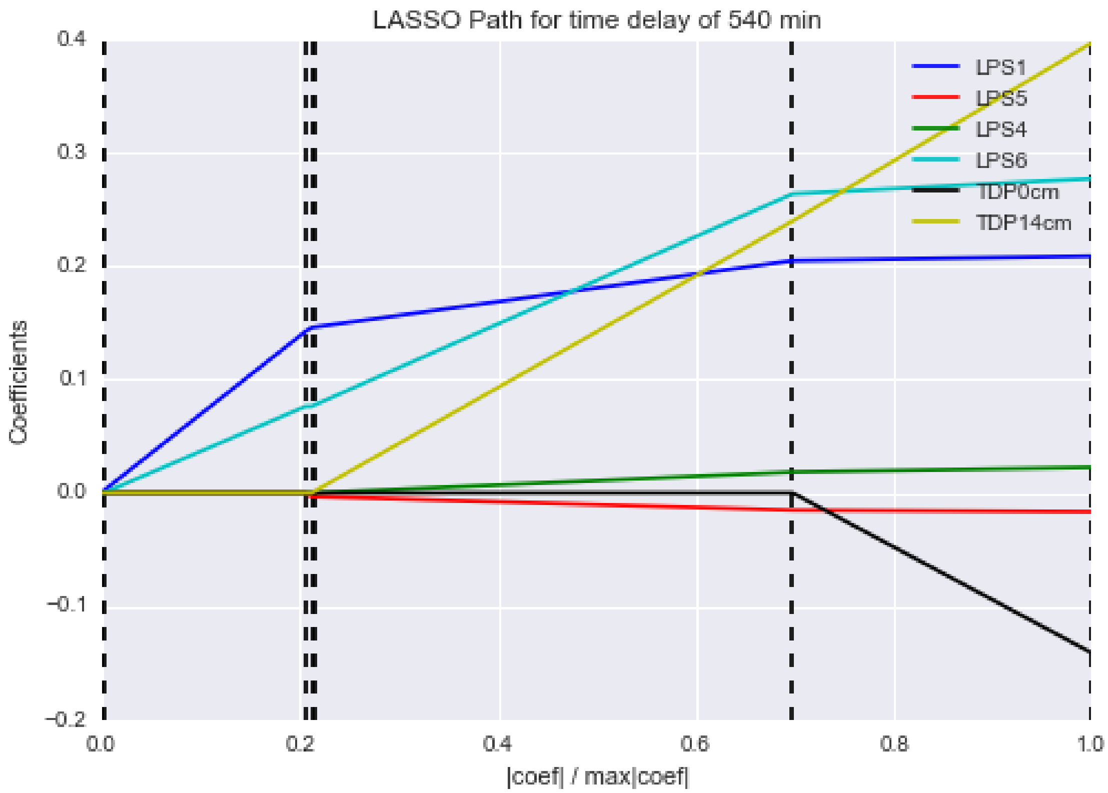

LASSO Path for time delay of 420 min
LPS1
LPS5
LPS4
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
0.2
0.4
0.6
0.8
1.0

480
.

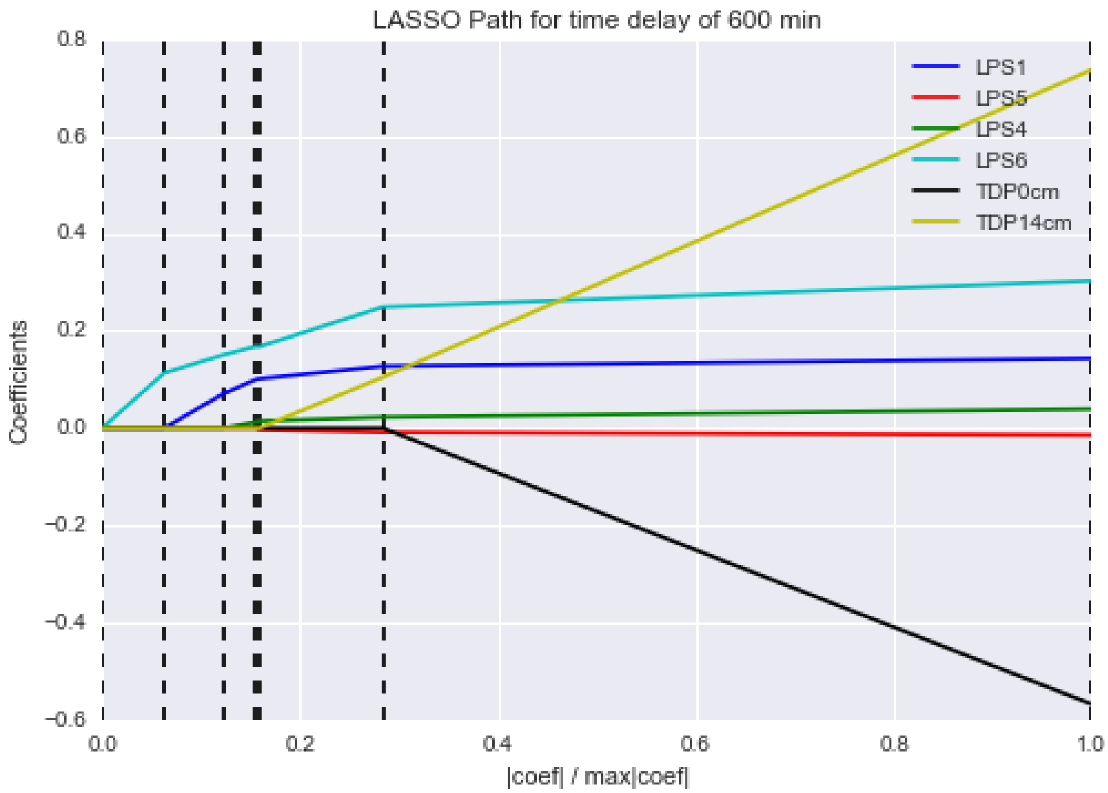

LASSO Path for time delay of 480 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0.0
0.2
0.4
0.6
0.8
1.0

540

.

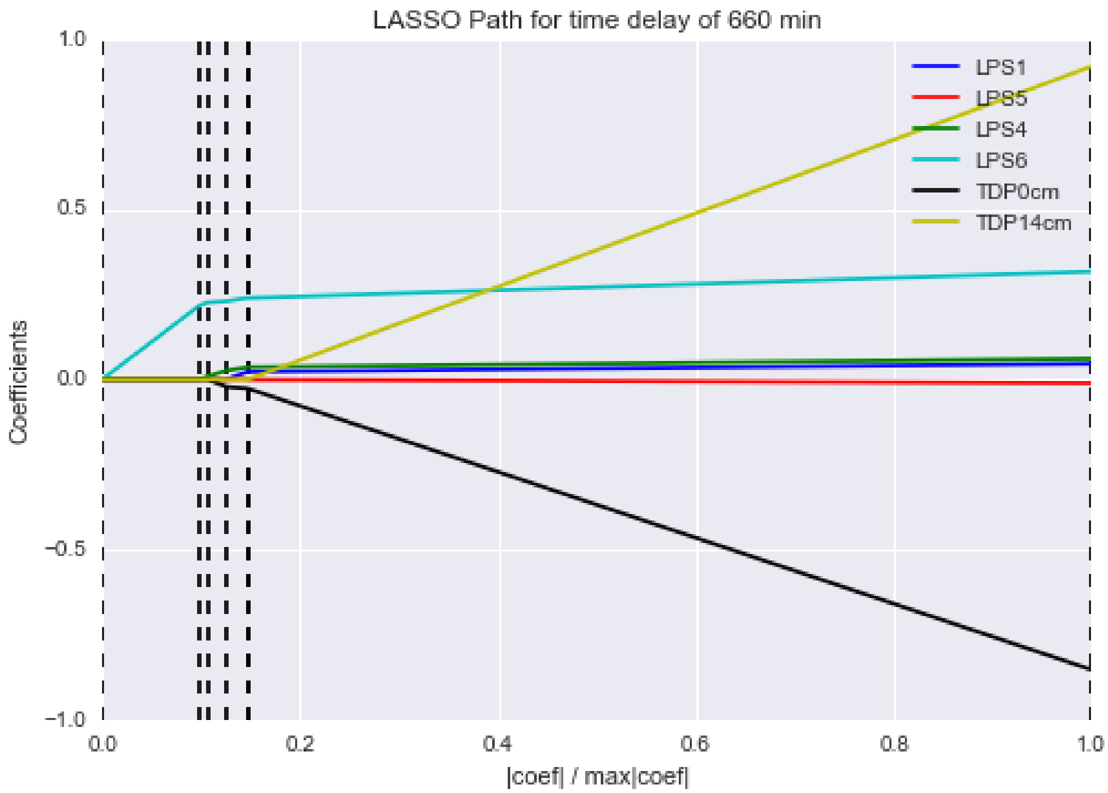

LASSO Path for time delay of 540 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

600
.

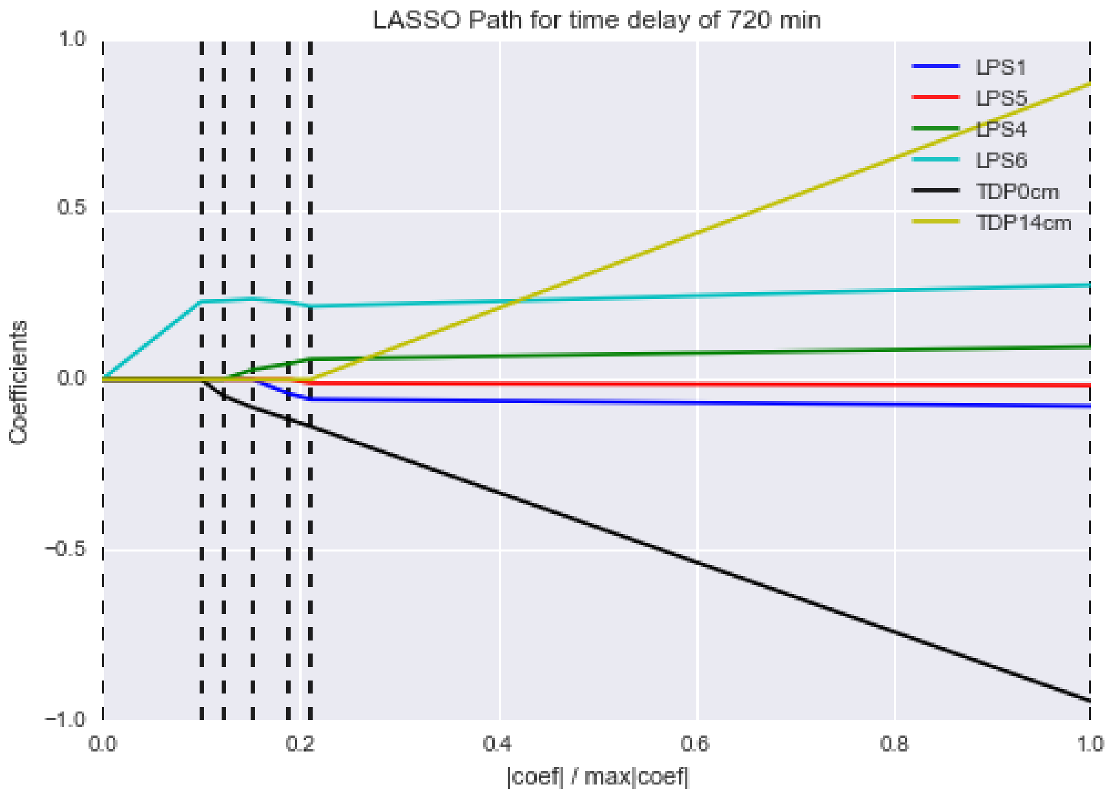

LASSO Path for time delay of 600 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|

660

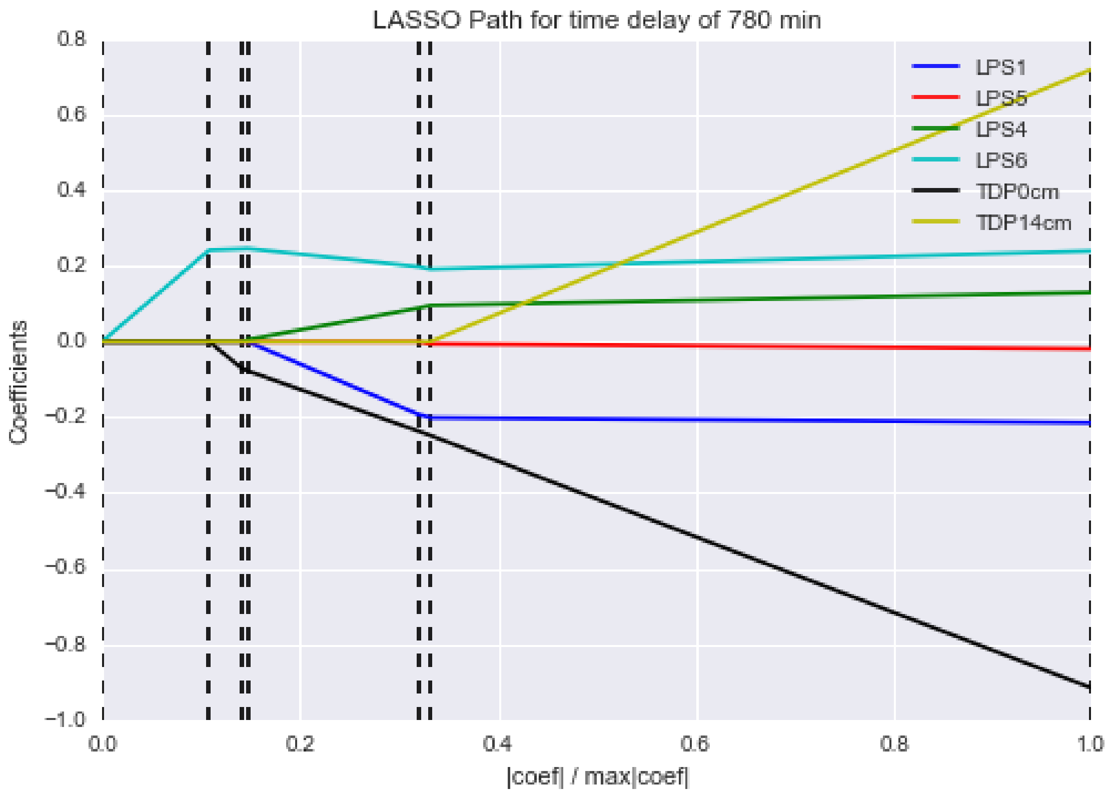
.


LASSO Path for time delay of 660 min
1.0
0.5
0.0
−0.5
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

720
.

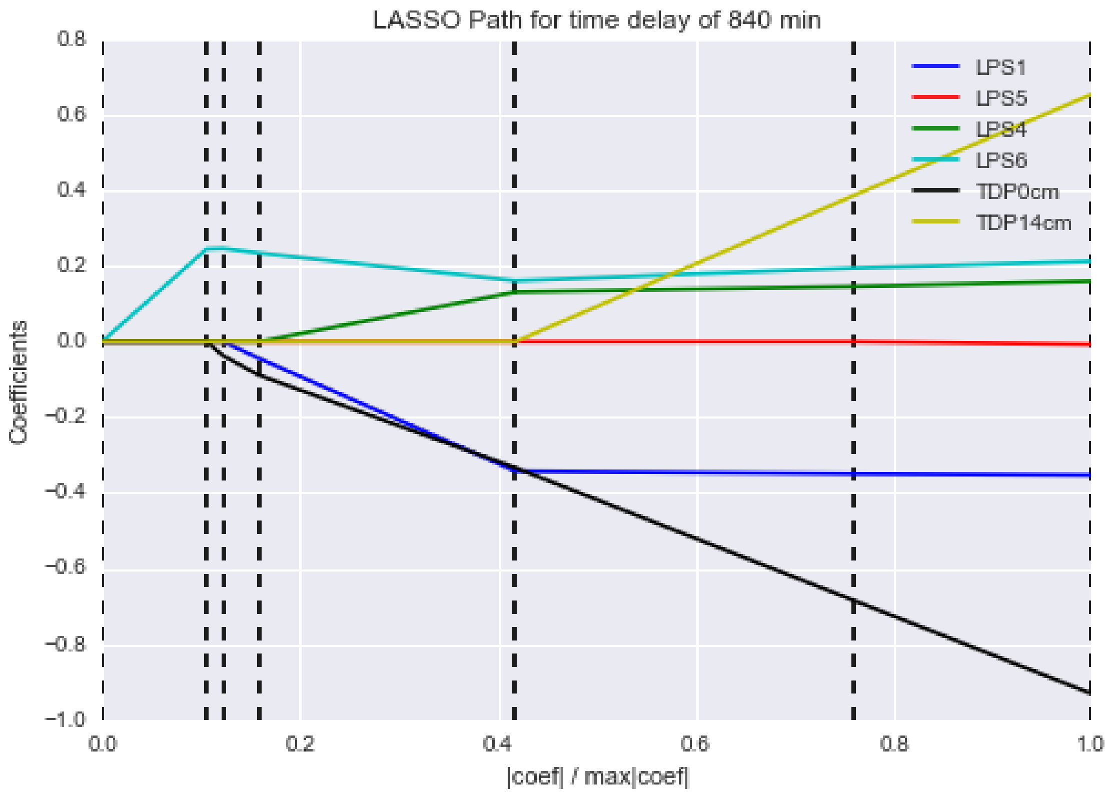

LASSO Path for time delay of 720 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

780
.

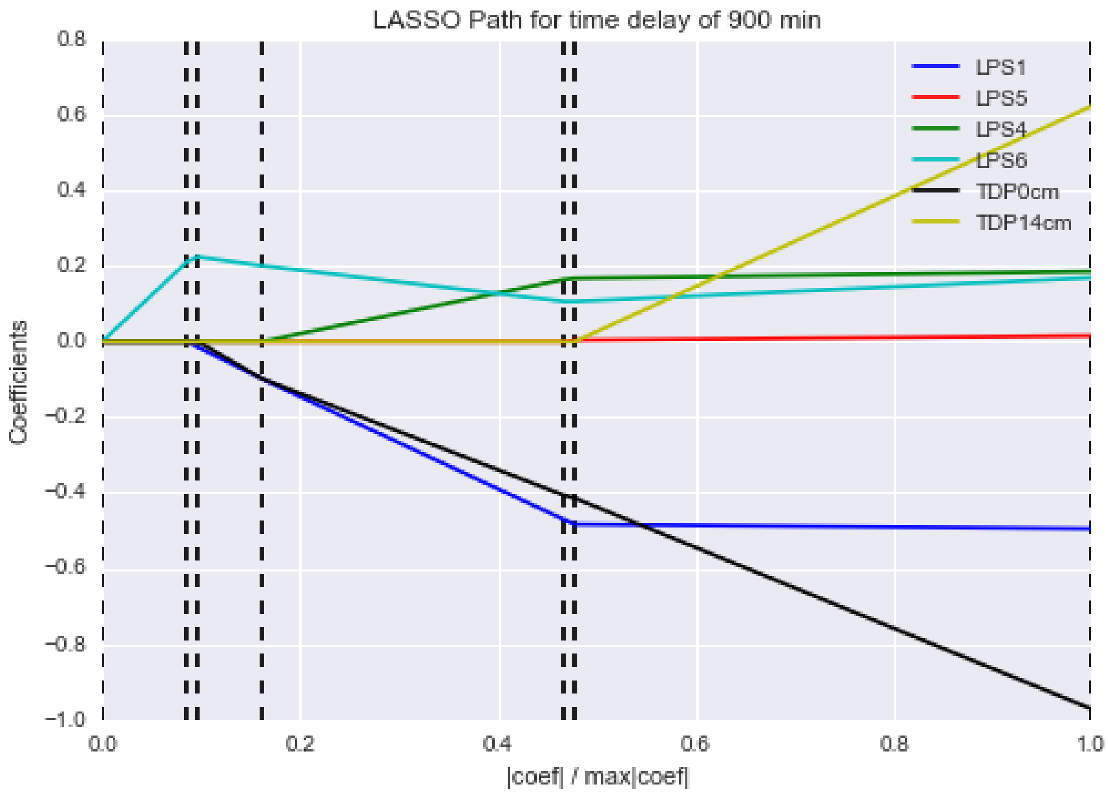


LASSO Path for time delay of 780 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

840
.

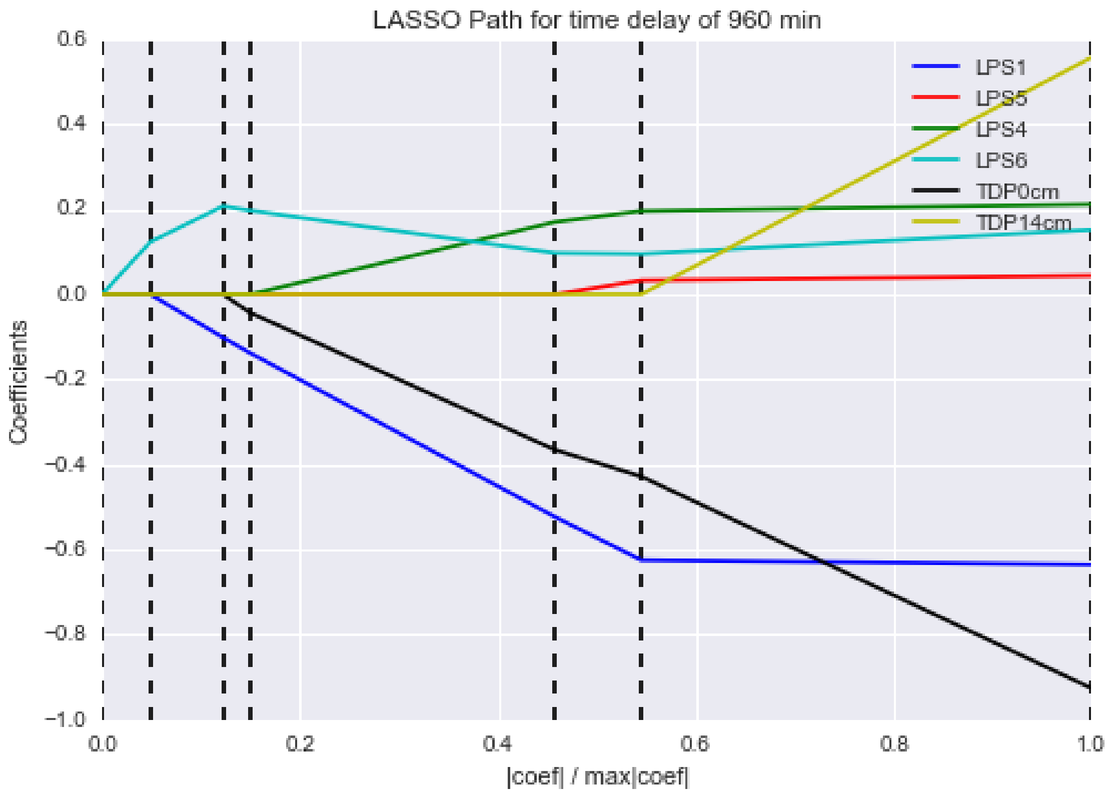

LASSO Path for time delay of 840 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

900
.


LASSO Path for time delay of 900 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

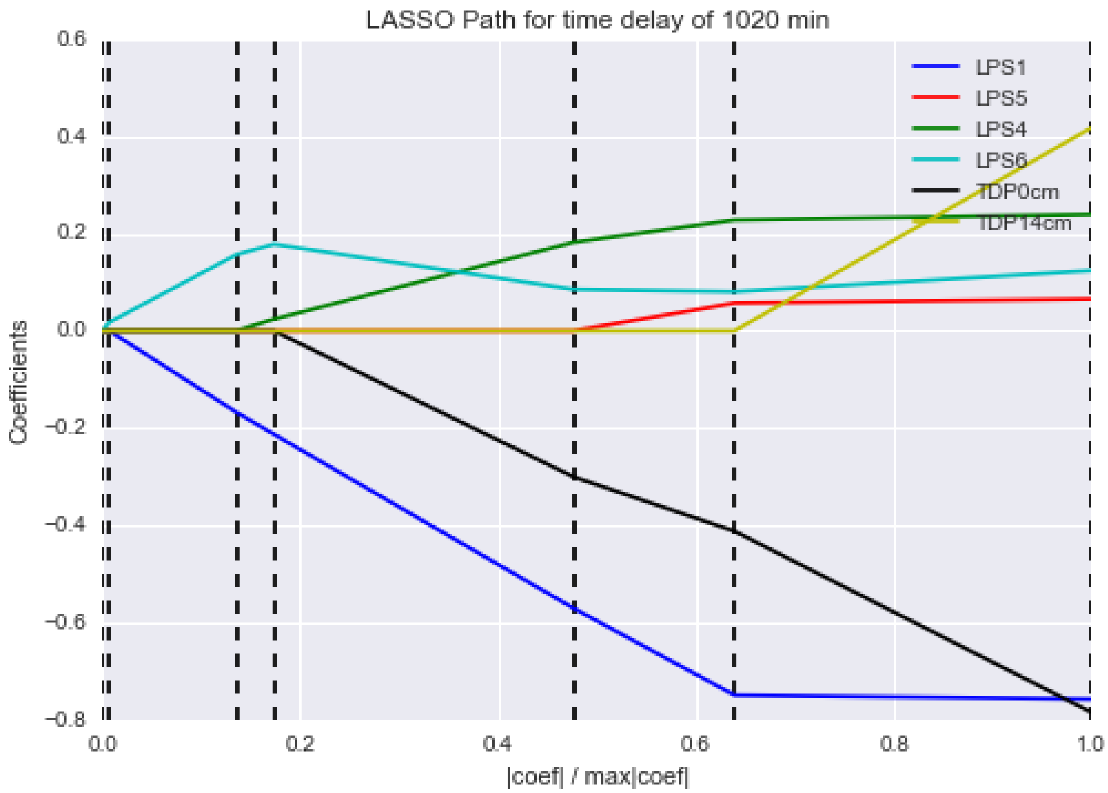
960

.

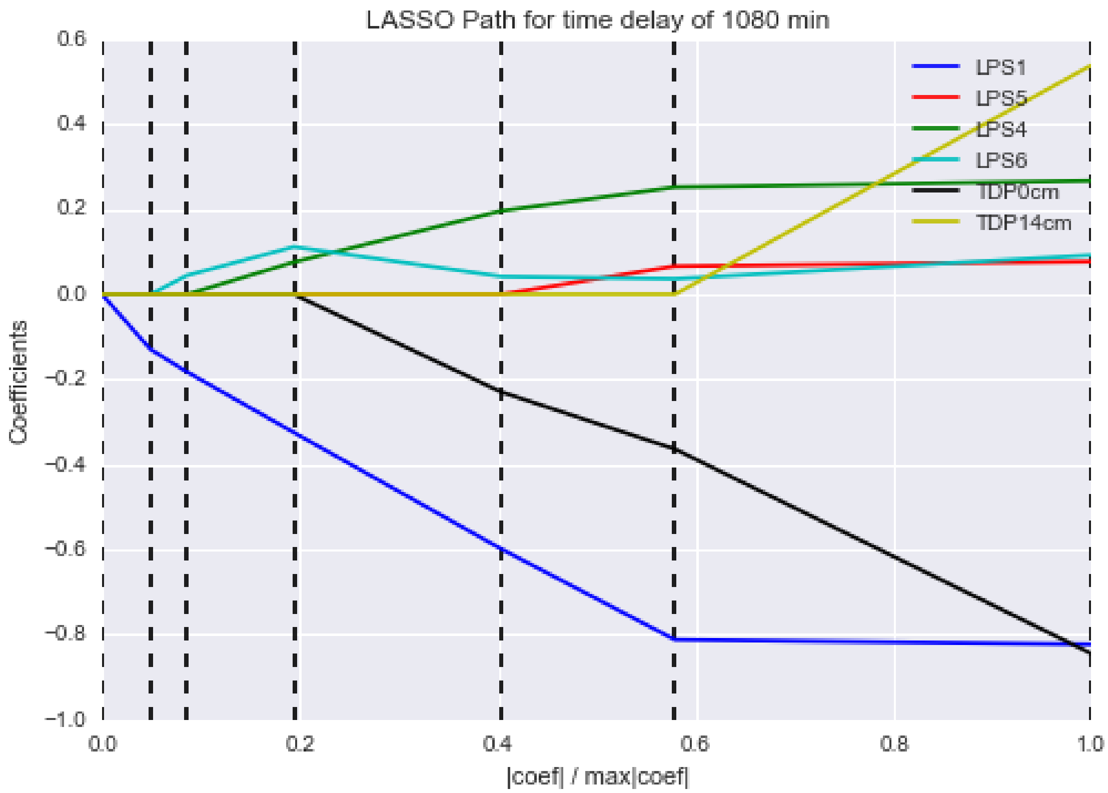

LASSO Path for time delay of 960 min
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

1020

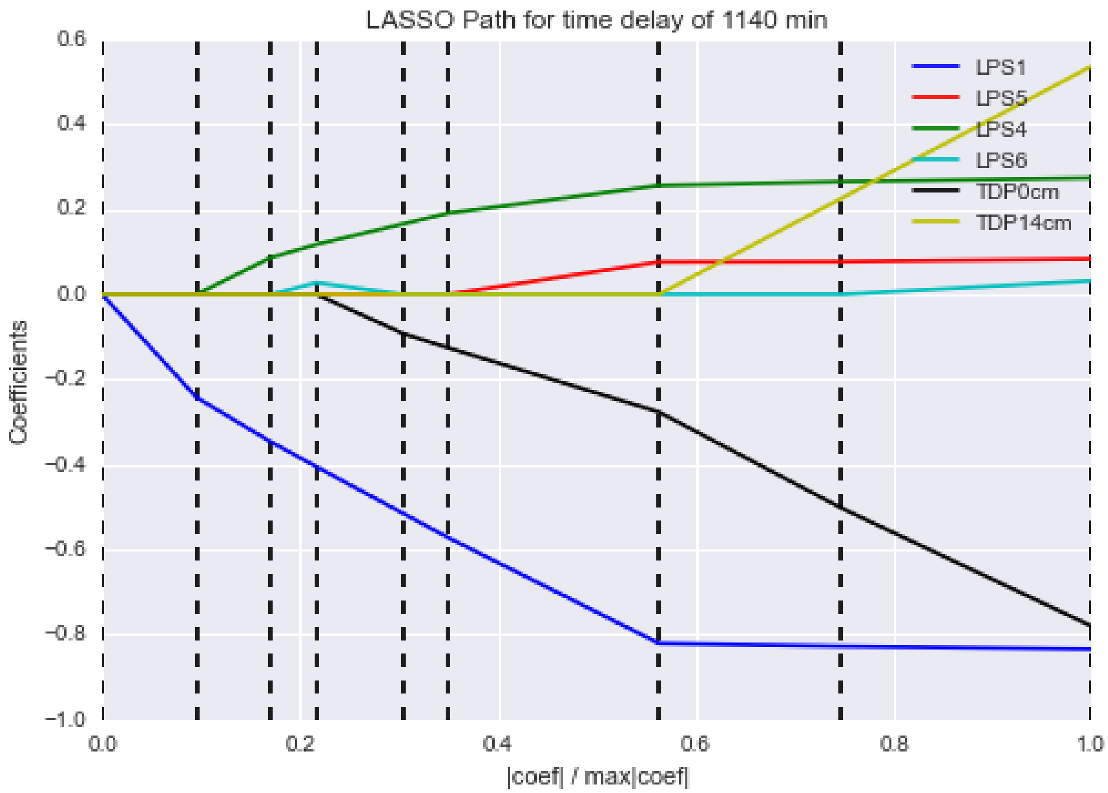
.


LASSO Path for time delay of 1020 min
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

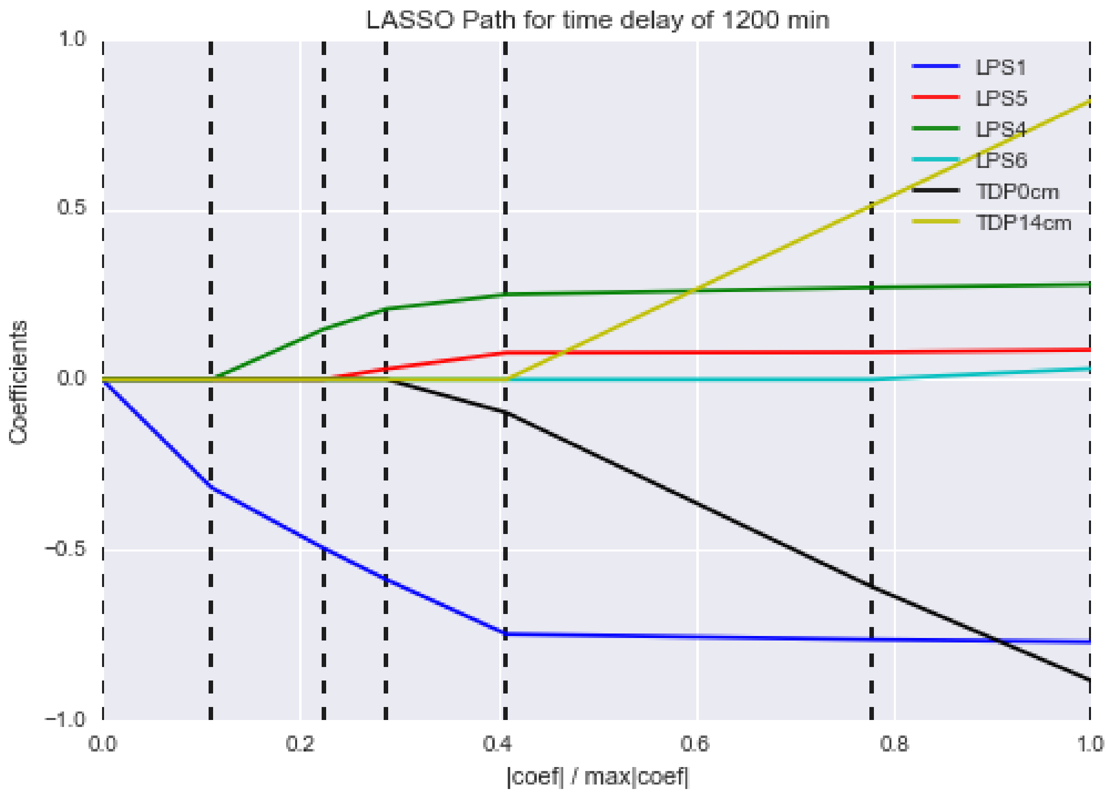
1080
.


LASSO Path for time delay of 1080 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

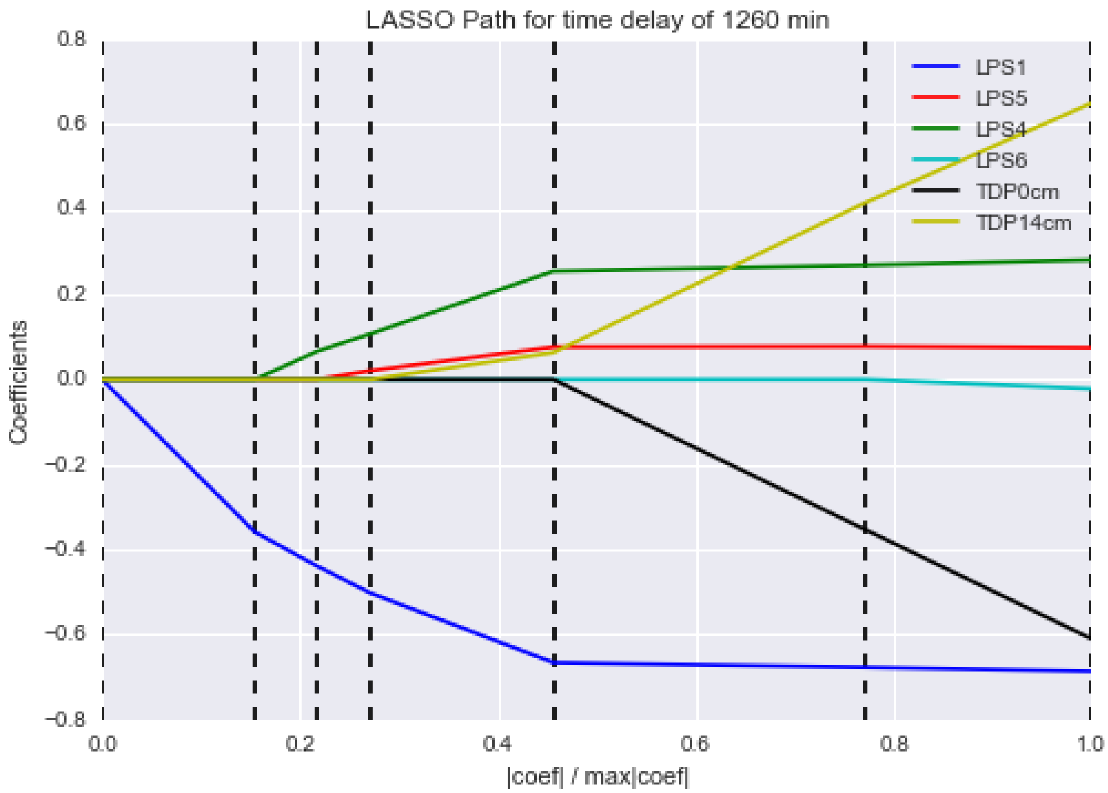
1140
.


LASSO Path for time delay of 1140 min
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Coefficients
0.0
0.2
0.4
0.6
0.8
1.0
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm

1200
.

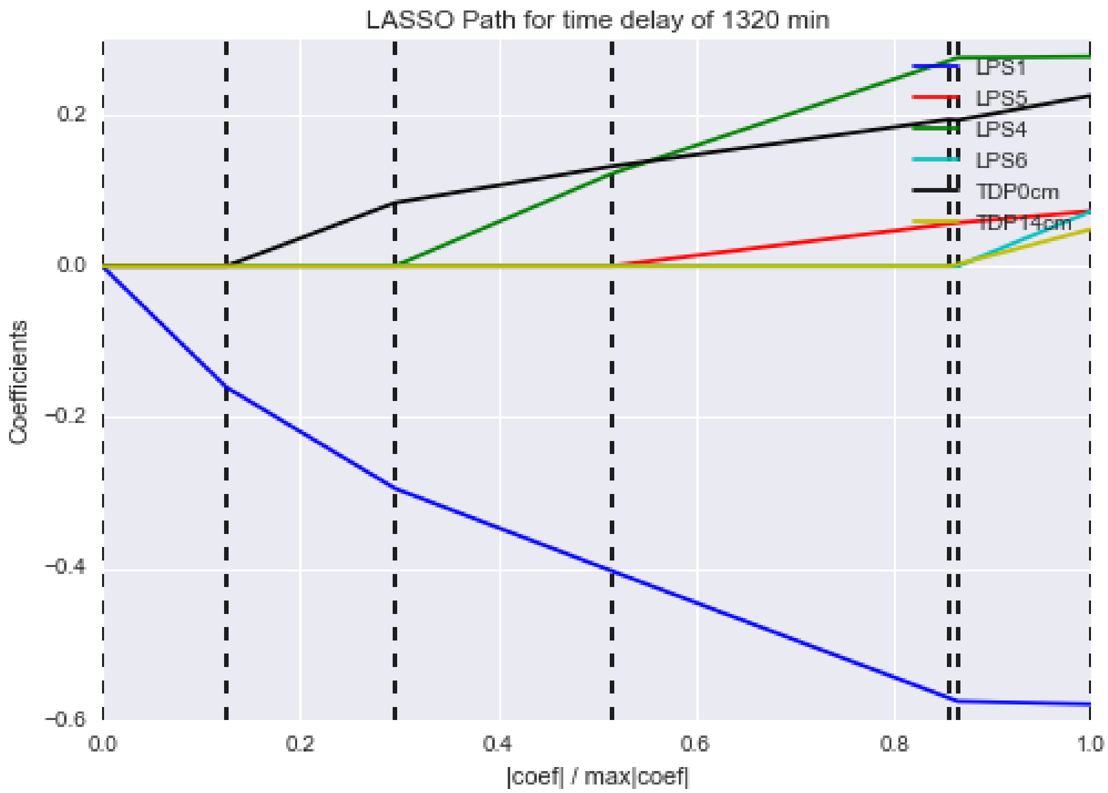

LASSO Path for time delay of 1200 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

1260
.


LASSO Path for time delay of 1260 min
Coefficients
|coef| / max|coef|
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.0
0.2
0.4
0.6
0.8
1.0

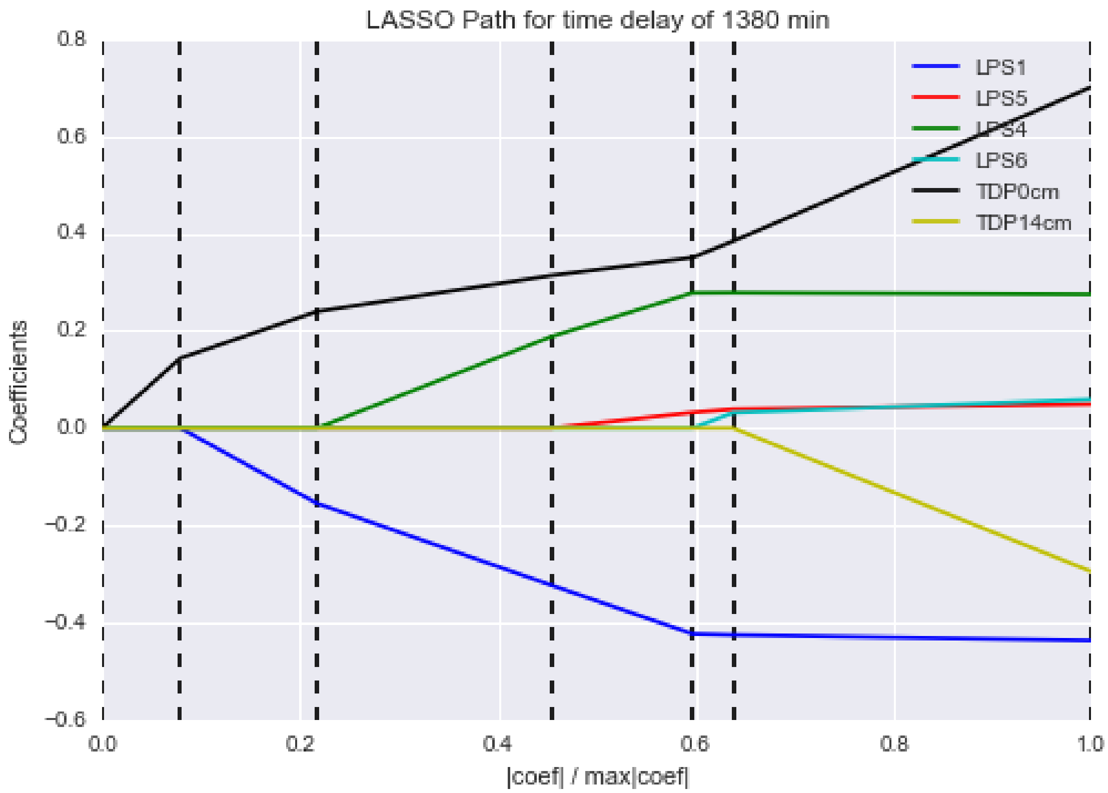
1320

.


LASSO Path for time delay of 1320 min
LPS1
LPS5
LPS4
LPS6
TDP0cm
TDP14cm
Coefficients
|coef| / max|coef|
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0

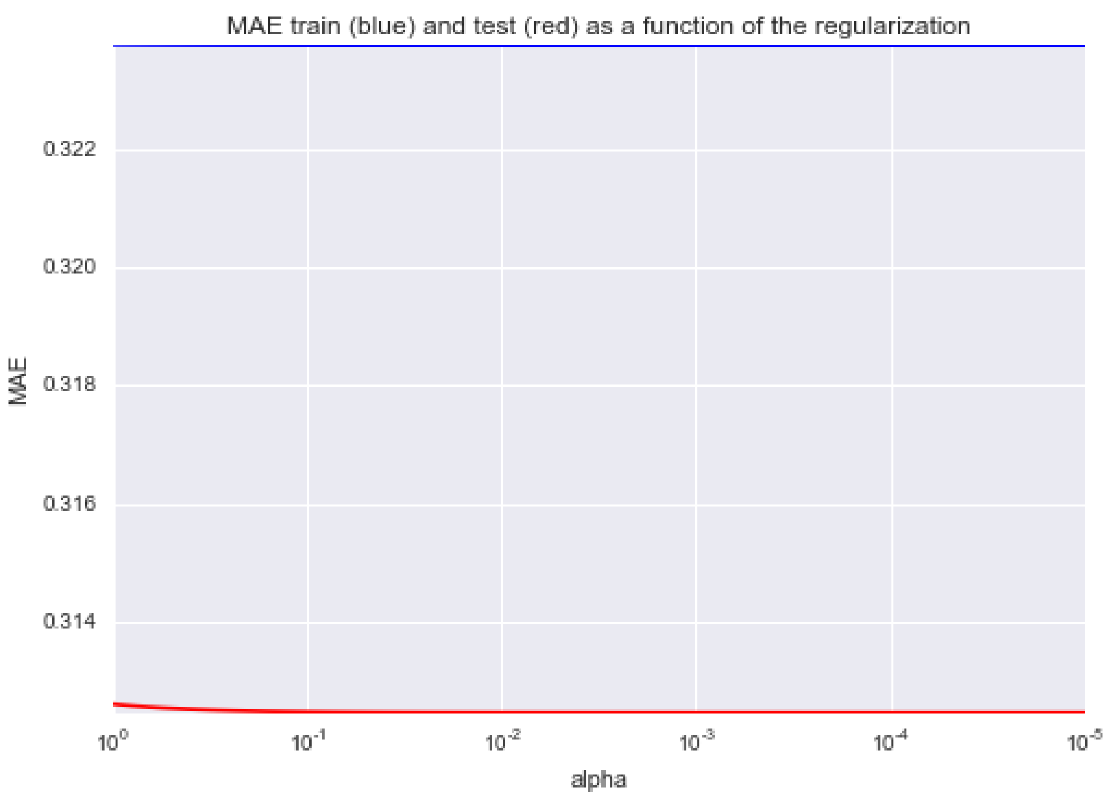
1380

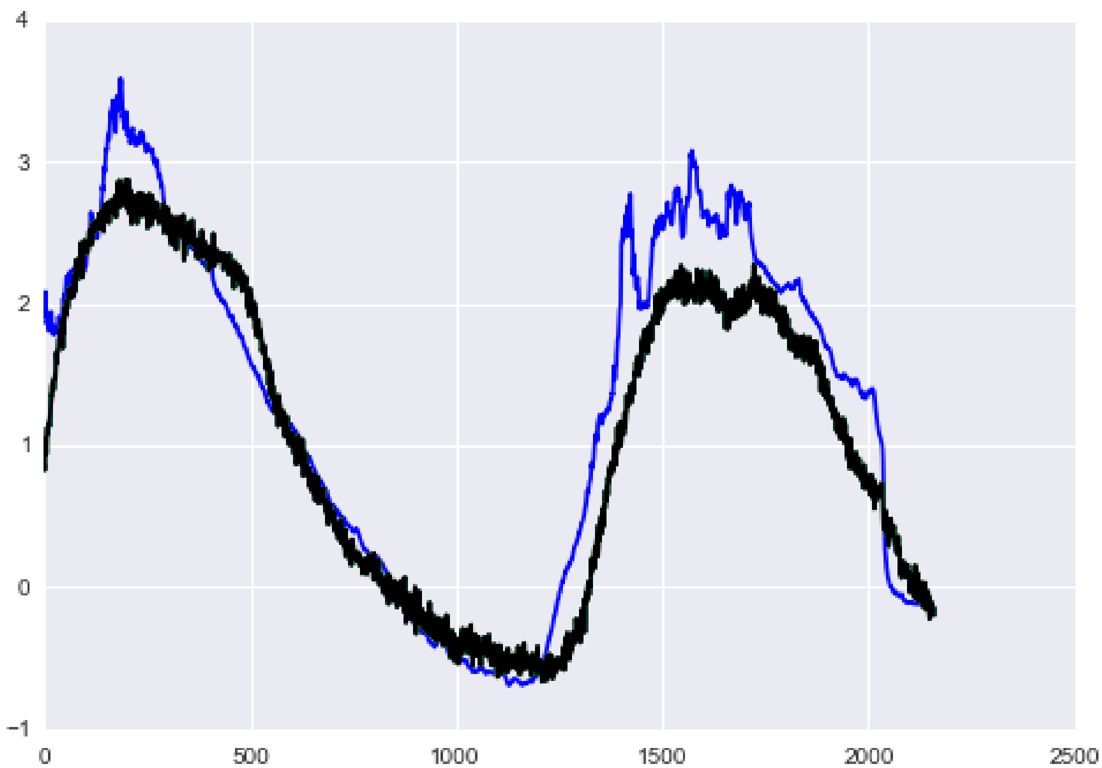
.



In [10]:
```
# EXPERIMENT: ridge regression with optimization of regul parameter
# train and test data
X_norm_train = X_norm[0:-STEPS_FOR_TESTING,:]
Y_norm_train = Y_norm[0:-STEPS_FOR_TESTING]

X_norm_test = X_norm[-STEPS_FOR_TESTING:,:]
Y_norm_test = Y_norm[-STEPS_FOR_TESTING:]

regul_a = 100
regul_a = np.logspace(-5, 0, regul_a)
clf = linear_model.Ridge(fit_intercept=False,normalize=False)
coefs = []
predictions = []
MAE_train = []
MAE_test = []
for a in regul_a:
    clf.set_params(alpha=a)
    clf.fit(X_norm_train, Y_norm_train)
    coefs.append(clf.coef_)
    # train error
    Y_pred = clf.predict(X_norm_train)
    MAE_train.append(np.mean(np.absolute(Y_pred-Y_norm_train)))
    # test error
    Y_pred = clf.predict(X_norm_test)
    predictions.append(Y_pred)
    MAE_test.append(np.mean(np.absolute(Y_pred-Y_norm_test)))
```

In [11]:
```
# plot MAE in function of regul parameter
ax = pl.gca()
ax.set_color_cycle(['b', 'r', 'g', 'c', 'k', 'y', 'm'])

ax.plot(regul_a, MAE_train)
ax.plot(regul_a, MAE_test)
ax.set_xscale('log')
ax.set_xlim(ax.get_xlim()[::-1])  # reverse axis
pl.xlabel('alpha')
pl.ylabel('MAE')
pl.title('MAE train (blue) and test (red) as a function of the regularization')
pl.axis('tight')
pl.show()

# plot predictions
ax = pl.gca()
ax.set_color_cycle(['b', 'r', 'g', 'c', 'k', 'y', 'm'])
pl.plot(Y_norm_test)
pl.plot(predictions[0])
pl.plot(predictions[40])
pl.plot(predictions[80])
pl.plot(predictions[99])
pl.show()
```





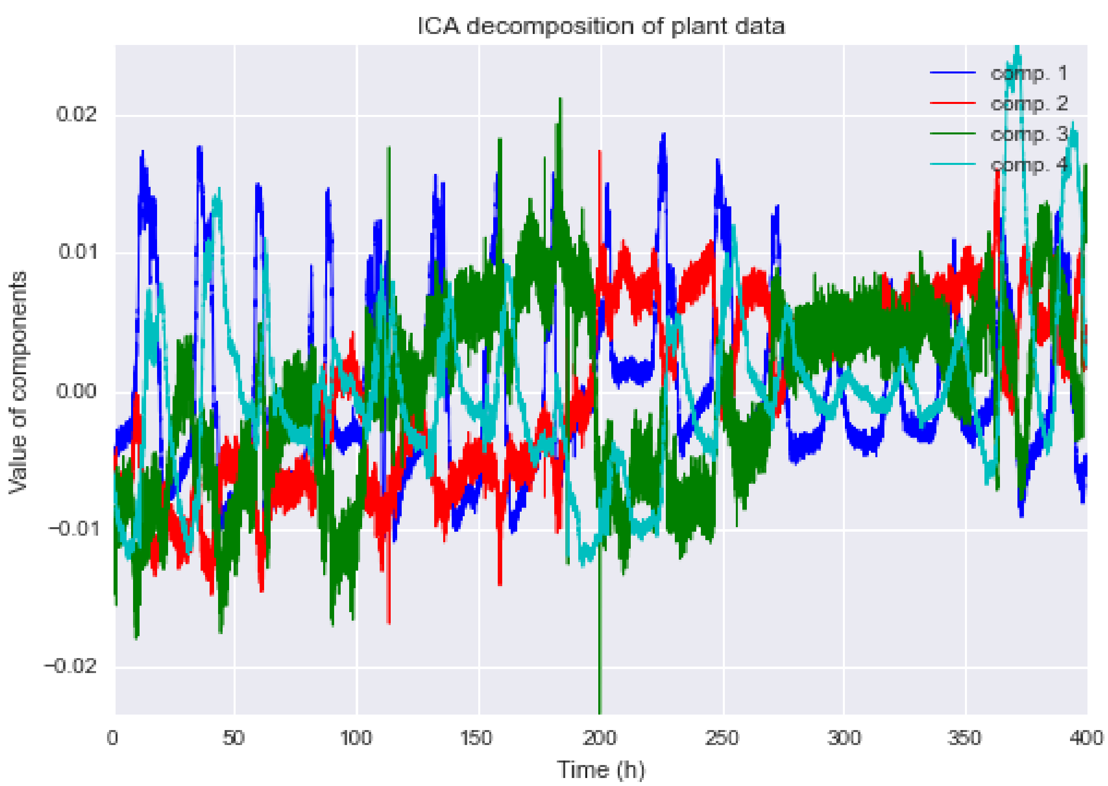

```
In [12]: from sklearn.decomposition import FastICA
         from sklearn.preprocessing import Imputer

         imputer = Imputer()

         data = pd.DataFrame.from_csv('20140907_data_plants_trial.csv')

         data = data.interpolate()

         data = (data - data.mean(0))/data.std()

         print data.describe()
```

```
temperature           LPS1          LPS5          LPS4          LPS6  \
count  2.405100e+04  2.405100e+04  2.405100e+04  2.405100e+04  2.405100e+04
mean   2.657157e-14  1.780464e-14  2.544632e-14  1.764868e-15 -9.013108e-15
std    1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00
min   -2.194354e+00 -3.638295e+00 -1.992857e+00 -3.335460e+00 -4.215856e+00
25%   -6.169427e-01 -4.749786e-01 -8.314994e-01 -7.865209e-01 -4.175932e-01
50%   -1.071095e-01  2.672273e-01  2.511648e-01  1.270013e-02  3.353692e-01
75%    5.394310e-01  6.527244e-01  8.735698e-01  7.104083e-01  6.897272e-01
max    3.773351e+00  2.476011e+00  1.478647e+00  3.722511e+00  3.267634e+00

             TDP0cm       TDP14cm
count  2.405100e+04  2.405100e+04
mean  -1.135076e-14  9.181546e-15
```

```
std    1.000000e+00  1.000000e+00
min   -8.272552e-01 -7.558887e-01
25%   -7.508075e-01 -6.941850e-01
50%   -6.017369e-01 -6.148720e-01
75%    6.317689e-01  4.679811e-01
max    2.707448e+00  3.134027e+00
```

```
In [13]: ICAmodel = FastICA(n_components=4)

         ICAmodel.fit(data.values)

         loadings = ICAmodel.transform(data.values)
```

```
In [14]: ax = pl.gca()
         ax.set_color_cycle(['b', 'r', 'g', 'c', 'k', 'y', 'm'])

         for i in range(4):
             ax.plot(np.linspace(0,len(loadings)/60, len(loadings)), loadings[:,i], label = 'comp. %s'%
         pl.xlabel('Time (h)')
         pl.ylabel('Value of components')
         pl.title('ICA decomposition of plant data')
         pl.axis('tight')
         plt.legend()

         pl.show()
```



```
In [15]: mixing = pd.DataFrame({data.columns[i] : ICAmodel.mixing_[i] for i in range(7)})
         print mixing
```

```
LPS1        LPS4        LPS5        LPS6      TDP0cm     TDP14cm  \
0  -34.920037  -28.335745    3.469700 -134.123031  130.426792  128.950162
1   59.200934  135.732807  135.232255   31.590661  -43.018814  -45.923116
2   12.764331  -53.125824   72.350158  -25.345953    4.451432    6.693564
3 -135.488352  -43.689535   19.703178  -57.173264   68.716174   68.562892

   temperature
0    86.751593
1     0.126005
2     3.290010
3   121.783849
```

We see that the data can be divided in different components that act on different timescales. The loadings can be found above.

```
In []:
```