Subject: Re: ML Project
From: Priyank Jain <jain206@purdue.edu>
Date: 10/10/2016 11:48 AM
To: Philip Michael Van Every <pvanever@purdue.edu>

No worries Phil, I too was caught up with my RA work. Thanks for the clarification on one class SVM. So yes, that's an interesting algo to report our results against.

Regarding your query, I don't have an answer now, since I am not exactly sure how feature selection would work with an SVC, but I'll update you in the weekdays as soon as I find an answer.

Regards,
//
Priyank Jain
Developer Team, Purdue IronHack
Research Center for Open Digital Innovation (RCODI)
Purdue University, Discovery Park
West-Lafayette, 47907-IN
Phone: +1-765-476-3000
Email: jain206@purdue.edu
Skype: live:priyank.purdue
About me: www.purdue.edu/opendigital/about/students
\\
On 10/09/2016 03:41 PM, Philip Michael Van Every wrote:

> Cool. Thank you. I am getting the Jupyter notebook set up now...
>
> One-class SVM is for anomaly detection. It would apply to our data if we consider the loans that default to be the anomalies. SVDD works the same way. In either case, we would train with only good loans. Then, when we test, we should detect bad loans as anomalies, outside of our hyperplane or hypersphere... These may actually be easier to work with on our data set because I think we have a lot more non-faulty loan data than faulty loan data. The binary svm ideally needs close to an equal amount of both faulty and non faulty loans to draw a boundary between them, but the OCSVM and SVDD just need non-faulty data create their boundaries.
>
> I have one more question about the feature selection phase: In class, we learned about selecting features that help the most with curve fitting the data: basically - fitting a plane in the higher dimension space that best fits the data. SVM classification, however, doesn't seek to *fit* a curve or plane to the data, but rather to create a plane that *separates* data. In many cases, the plane that fits the data would actually be orthogonal to a boundary that separates it wouldn't it? I guess we can still use curve fitting to discover which features have the most co-information with the data labels though right? Is that the idea? If that is the case, then the theta parameters used in feature selection would be very different from the theta parameters used in classification, because the former seeks to fit a plane to *match* the data and the latter

seeks to create a plane that *splits* the data... Am I thinking about this the right way?

Incidentally, the linear SVC library you sent is built on libsvm, which is the SVM classification library I have used heavily before and am fairly familiar with. I used it in Matlab instead of Python but the basic idea is the same. It is built on underlying, optimized c libraries. It uses sequential minimal optimization,SMO, (rather than gradient descent or other optimization methods).

I am looking into normalization now. Probably, for the sake of reducing computational complexity for the SMO we could probably just normalize data to 0-1 values or perhaps shift each feature by its mean and scale by its variance or standard deviation. These are fairly easy things to do so I will cook up some modules that do this. Using a 'Normalizer' abstract parent class (or abstract interface - I'm not sure what the preferred terminology is in Python quite yet- but the basic idea is the same), I'll make some concrete child classes that do the 0-1 and Gaussian normalization described above, as well as some child classes that do different stuff. This will be a gentle introduction to Python coding for me, so that's good. I'll start pushing stuff to the repo as I finish it.

I know we planned on talking again tonight, but some other obligations have come up for me so I think I'll have to miss it. Sorry about that. I think however, that with our end-to-end system sort of mapped out and each of us working on a subsystem in the pipeline we are off to a good start. I'm pretty sure I know what to do with normalization, and it sounds like you have a handle on what to do with feature selection.

If you have any questions or comments just let me know. I may be slow to respond the rest of the night because I have to update Ubuntu, but I'll get back to you asap.

Cheers,
Phil

From: Priyank Jeevanlal Jain
Sent: Saturday, October 8, 2016 5:18:50 PM
To: Philip Michael Van Every
Subject: Re: ML Project

Hi Phil,
Jupyter notebooks are pretty handy since we would deal with code, output and charts at the same time. Jupyter notebooks allow you to have them all in one place. https://ipython.org/notebook.html Comes pre-installed if you are using anaconda version of python. Which I highly recommend.

I think we'll have to use this for SVM classification. http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
Our proposal mentions we would play with one class svm and binary svm. Not able to grasp how would we use one class svm in this setting? Is it more

of anomaly detection scenario?

Regards,
//
Priyank Jain
Developer Team, Purdue IronHack
Research Center for Open Digital Innovation (RCODI)
Purdue University, Discovery Park
West-Lafayette, 47907-IN
Phone: +1-765-476-3000
Email: jain206@purdue.edu
Skype: live:priyank.purdue
About me: www.purdue.edu/opendigital/about/students
\\
On 10/08/2016 11:37 AM, Philip Van Every wrote:

> Sounds good! Do you want to meet or should we skype?
>
> On Oct 8, 2016 10:04 AM, Priyank Jeevanlal Jain <jain206@purdue.edu> wrote:
>
> > Hi Phil,
> >
> > Good morning! Hope you're doing well.
> >
> > Do you want to start working on the ML project today at 3:30?
> > Please let me know, thanks!
> >
> > Regards,
> > Priyank Jain
> > Graduate Student,
> > Masters in Computer Science,
> > Purdue University
> > +1-765-476-3000