
SVMmulticlass

2016-11-21

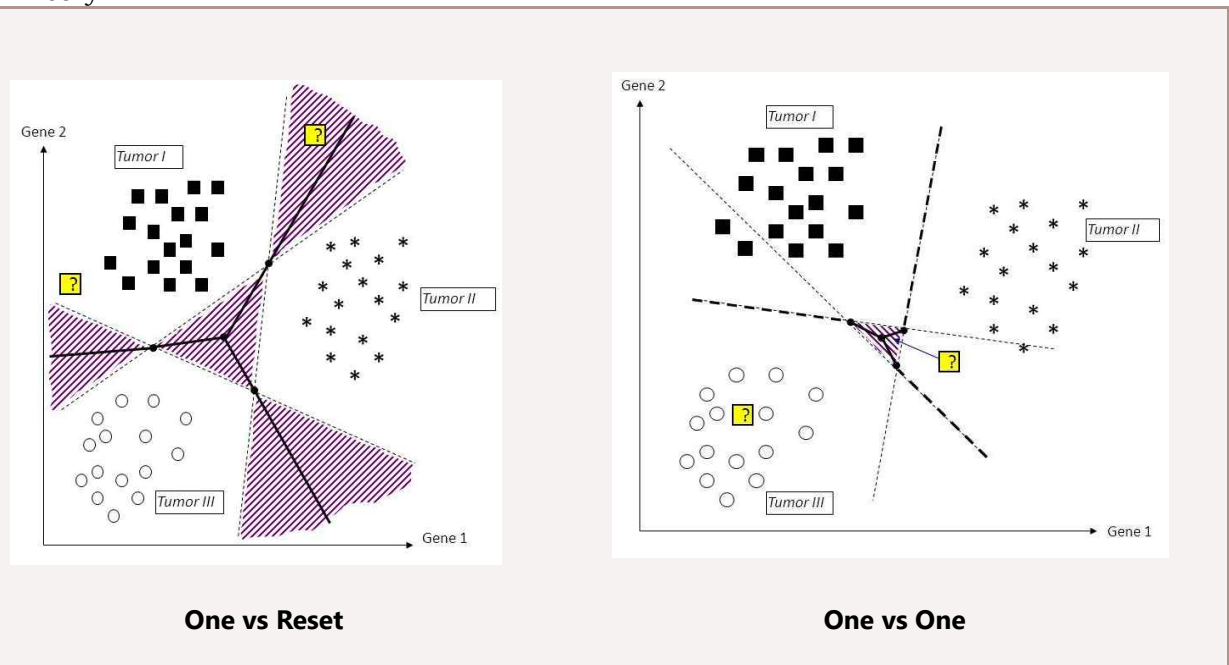
SVMmulticlass 의 기본적인 사용법을 정리하고 여러 연구영역에 적용 가능하도록 한다.

SVM_{multiclass}

Overview

설명	
Version	2.20(2008/08/14)
URL	http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html
Language	C
Method	Support Vector Machines (SVMs)
Feature	Multiclass Classifier Toolkit

Theory



Multiclass Support Vector Machine(Multiclass SVM)

기본적으로 svm은 이진 분류기지만 이진분류법을 확장해서 멀티클래스 분류를 하는 방법이 있는데 대표적으로 one vs rest approach(one vs all) 와 one vs one approach가 있다.

1. One vs Rest

전체 class의 수가 M 개 라고 하면 i 번째 class의 부류 와 i 클래스를 제외한 나머지 $M-1$ 개의 클래스가 속하는 클래스 로 이진화 시키고 분류기를 만들고, 이와 같은 작업을 M 번 만큼 한다. 그러면 총 M 개의 이진분류기가 만들어 지게 된다. 즉 i class에 속하는 샘플을 $+1$ 라벨을 붙이고 나머지 샘플에 -1 라벨을 붙인다. 그래서 훈련집합(training set)을 만들게 된다. svm의 결정 초 평면도 M 개가 만들어 지는데 j 번째 초 평면을 $d(j)$ 라 할 수 있다.

실제 Test를 할 때에는 M 개의 초 평면에 모두 test를 하게 되는데 m 번의 분류에서 1가지만 양수를 출력하고 나머지는 모두 음수를 출력한다면 문제 될 것이 전혀 없지만, 항상 그렇게 된다는 보장은 없다.

따라서, m 번의 test중 가장 큰 $d(j)$ 를 갖는 것을 예측된 class로 한다. 이 방법은 크게 2가지 문제를 가지고 있다. 첫 번째로, 초 평면값 $d(j)$ 의 크기를 단순 비교해서 값을 찾는 다는 것이 비합리적일 수 있다. 두 번째로는 이진 분류기의 훈련집합이 불균형을 이룬다는 것이다. ($1 : m-1$)

2. One vs One

m 개의 클래스 중 2개를 선택하여 2클래스에 대한 결정 초 평면을 만든다. 그렇게 되면 결정 초 평면과 분류기는 mC_2 개만큼 생기게 된다. $M(M-1)/2$ 개 된다. 이제 test 할 때는 투표개념을 도입하여 분류하는데 새로 들어온 sample x 에 대해서 초 평면 $d_{ij}(x)$ 가 x 를 class i 로 분류하면 class i 에 +1점, j 로 분류하면 class j 에 +1 점을 준다. 즉 mC_2 개의 분류기가 M 개의 class에 대해서 투표를 하는 것이다. 이렇게 모든 초 평면에 대해서 분류 했을 때 가장 높은 점수를 획득한 class가 predicted된 클래스이다. 이때 얻을 수 있는 최대 표 값은 $M-1$ 개 이고 이 방법은 one vs rest가 가지고 있는 문제를 가지진 않는다. 그러나, 클래스의 개수 M 이 커지면 이진 분류기의 수가 많아지고 결국 learning 과 test에 걸리는 시간이 많아진다.

실제적으로 가장 많이 사용되는 분류방법은 one vs rest 방법이며 이진 svm을 멀티클래스 svm으로 확장하는 방법에 대한 연구도 진행 중이다.

Training Step


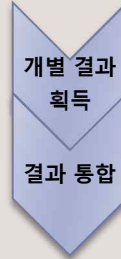
실행 방법	./svm_multiclass_learn [-option] -c default_number train_file model_file	
Argument	default_number	Training Error와 margin의 trade-off
	train_file	학습을 위한 Text File
	model_file	학습된 결과로 생성되는 Model File
File Format (Train_file)	Line Format	<code><line> .=. <target> <feature>:<value> <feature>:<value> ...</code> <code><feature>:<value> # <info></code> <code><target> .=. <int></code> <code><feature> .=. <integer> "qid"</code> <code><value> .=. <float></code> <code><info> .=. <string></code>
	LINE	Case 1개를 의미 → 예) 문서분류일 경우 1문서
	Target	결과 Class, Multiclass Classifier이므로 Class의 번호를 기입
	Feature	ID로 표현된 출현한 자질. ID 번호로 increasing order로 Sorting 되어있어야 함. → 1:value 8:value 99:value ... n:value
	Value	Float value, 자질의 weight값 TF-IDF, χ^2 분포값, ... 등의 값을 weight로 사용할 수 있음

Evaluation Step

실행 방법	./svm_multiclass_classify [options] example_file model_file output_file	
Argument	Example_file	실험을 위한 Text File
	Model_file	실험에 사용될 Model File
	Output_file	Example 파일에 대해 문장단위로 결과 float value들이 출력된 파일
File Format (Example_file)	Train_file과 동일	
File Format	Line Format	<code><line> .=. <value></code>

(output_file)	value	데이터에 가능한 클래스 각각의 score값. 클래스 개수마다 출력되는 score의 개수가 다르다 가장 높은 값을 가진 클래스가 선택됨
Result Example	[wilowisp@nlpcat svm_temp]\$../svmlight/svm_multiclass_classify test01.input train01.model Reading model...OK. (1570 support vectors read) Classifying test examples..done Runtime (without IO) in cpu-seconds: 0.00 Accuracy on test set: 96.00% (24 correct, 1 incorrect, 25 total) Precision/recall on test set: 100.00%/66.67%	
	Accuracy	전체 발생한 경우 중 맞춘 개수
	Precision	클래스에 속한다고 할당한 것 중 맞춘 비율(2개 할당, 모두 맞춤)
	Recall	클래스에 속한 것 중에서 할당한 비율(전체 3개 중 2개 맞춤)

Multiple Class Problem

대상	대부분의 문제는 Class가 다수 존재 → 어떻게 binary classifier를 확장?	
N class problem 처리 과정	문제재정의	모든 class 마다 class에 속하는 경우, class에 속하지 않는 경우 가 정하여 binary classification 문제로 치환
	개별 학습	 <ul style="list-style-type: none"> •특정 Class에 대해 주어진 n-class tagged corpus를 class binary corpus로 변환 •위에서 생성한 학습 데이터를 가지고 SVMlight를 이용하여 이진 분류기 학습
	결과 통합	 <ul style="list-style-type: none"> •각 학습기마다 입력 자질에 대해 판단 결과를 가져옴 •weight값의 대소 비교를 통해 가장 높은 값을 가지는 것으로 Class 결정
제약사항	자질	결과 값을 서로 비교할 수 있어야 하므로, 자질의 집합은 동일한 것을 사용해야 한다. SVM의 출력 결과는 통계적으로 정규화 된 것이 아니므로 서로 다른 입력 차원의 경우 비교 대상이 될 수 없다.
	가중치판단	학습에서 출현하지 않은 경우 등에 대해 모든 classifier에서 음의 값을 제공하는 경우도 있음 각 classifier 들이 제공하는 음의 값 중에서 가장 큰 것을 고르는 경우 전체적으로 성능이 긍정적인 영향이 있음(정확률에서는 다소 떨어질 수 있으나, 재현율에서 +)

Examples

Original Tagged File

```
<1,1> greeting : 아/름/nq_per 아/j 잘/ma 자/pv 앓/ep 니/ef ?/sf L_STT-nq_per nq_per-jj-ma ma-pv
pv-ep ep-ef ef-sf sf-L_END
<1,2> greeting : 네/i ./sp 잘/ma 자/pv 앓/ep 습니다/ef ./sf L_STT-i i-sp sp-ma ma-pv pv-ep ep-ef
ef-sf sf-L_END
<1,3> request : 아/름/nq_per 아/j 일정/ncn 확인/ncp 좀/ma 하/pv 아/ef 주/px 어/ef ./sf L_STT-
nq_per nq_per-jj-ncn ncn-ncp ncp-ma ma-pv pv-ef ef-px px-ef ef-sf sf-L_END
<1,4> ask_ref : 언제/np 일정/ncn 올/j 확인/ncp 하/xsp ㄹ까요/ef ?/sf L_STT-np np-ncn ncn-jj-ncp
ncp-xsp xsp-ef ef-sf sf-L_END
<1,5> response : 다음달/ncn 일정/ncn 좀/ma 알려주/px 어/ef ./sf L_STT-ncn ncn-ncn ncn-ma ma-
px px-ef ef-sf sf-L_END
...
```

SVMLight multi-file 예제

Multi-tagged File	관련 Data
1 1:7945.66088625861 4:6017.38540453272 8:4897.02766119255 11:4179.2751295621 14:3873.26997419427 22:3177.30296564952 58:1632.99710926134 62:1548.52112972171 267:195.0542232421 1 3:7164.23498910579 4:6017.38540453272 18:3512.11663501681 37:2192.23814986522 45:1990.20930778189 66:1476.93835347272 96:1051.9636892823 97:1033.29719751867 104:899.925171944036 11:4783.377974509735 267:195.0542232421 7 1:7945.66088625861 5:5502.54909636758 6:5399.76183467778 7:5341.71892361314 8:4897.02766119255 11:4179.2751295621 12:4137.52870500283 14:3873.26997419427 20:3472.29639552326 22:3177.30296564952 26:2932.03863261593 35:2311.77852772425 62:1548.52112972171 64:1508.52962447446 66:1476.93835347272 71:1390.14318029496 72:1313.90868489066 124:726.6600872903 142:601.079111093293 168:454.601517155478 267:195.0542232421 4 1:7945.66088625861 23:3120.09282966685 38:2184.33063096293 43:2083.46899931013 51:1837.23047793168 52:1818.50859809272 56:1649.11128854405 64:1508.52962447446 66:1476.93835347272 75:1259.69141468807 86:1145.60686843271 104:899.925171944036 168:454.601517155478 178:396.495747755885 267:195.0542232421 6 3:7164.23498910579 58:1632.99710926134 64:1508.52962447446 157:524.636989304767 163:489.664967971109 267:195.0542232421	- Class ID 1 : greeting 7 : request 4 : ask_ref 6 : response - Feature File 아/름/nq_per : 1 아/j: 4 ... - Weight file 1: 7945.66088625861 4:6017.38540453272 ... - Weight 결정 방법 ■ TF-IDF ■ χ^2 -statistics ■ Correlation coefficient ■ ...

SVM multiclass 실행 시 반드시 참고!

```

40 #-----#
41 #----  SVM MULTICLASS  ----#
42 #-----#
43
44 svm_multiclass_classify: svm_light_hideo_noexe svm_struct_noexe svm_struct_api.o svm_struct/
ct/svm_struct_classify.o svm_struct/svm_struct_common.o svm_struct/svm_struct_main.o
45 $(LD) $(LDFLAGS) svm_struct_api.o svm_struct/svm_struct_classify.o svm_light/svm_comm
on.o svm_struct/svm_struct_common.o -o svm_multiclass_classify $(LIBS) -lm
46
47 svm_multiclass_learn: svm_light_hideo_noexe svm_struct_noexe svm_struct_api.o svm_struct_
learn_custom.o svm_struct/svm_struct_learn.o svm_struct/svm_struct_common.o svm_struct/sv
m_struct_main.o
48 $(LD) $(LDFLAGS) svm_struct/svm_struct_learn.o svm_struct_learn_custom.o svm_struct_a
pi.o svm_light/svm_hideo.o svm_light/svm_learn.o svm_light/svm_common.o svm_struct/svm_st
ruct_common.o svm_struct/svm_struct_main.o -o svm_multiclass_learn $(LIBS) -lm

```

'Makefile'의 line 45,48의 마지막에 반드시 '-lm'을 추가할 것. (-lm : 표준수학 library)

Reference

1. SVM visual tool. *Royal Holloway, University of London*. [Online] <http://svm.dcs.rhnc.ac.uk>.
2. SVM 설명. *aistudy*. [Online] http://www.aistudy.co.kr/pattern/support_vector_machine.htm.
3. svmmulticlass main page. [Online] http://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html