Simulation studies on Support Vector Machine in a SUSY search

Marco Costa

Scuola Normale Superiore/University of Pisa

Supervisor: Dirk Krücker

September 7, 2017

Abstract

This work is meant to show that the support vector machine (SVM) can be a useful tool in high energy physics search. After a brief introduction in SUSY, we proceed with the description of the strategy implemented for the stop search analysis. Then we give an overview on the main ideas behind SVM. Finally, we show the performance of SVM on different figures of merit and if the results obtained show a dependance on the evaluation sample.

Contents

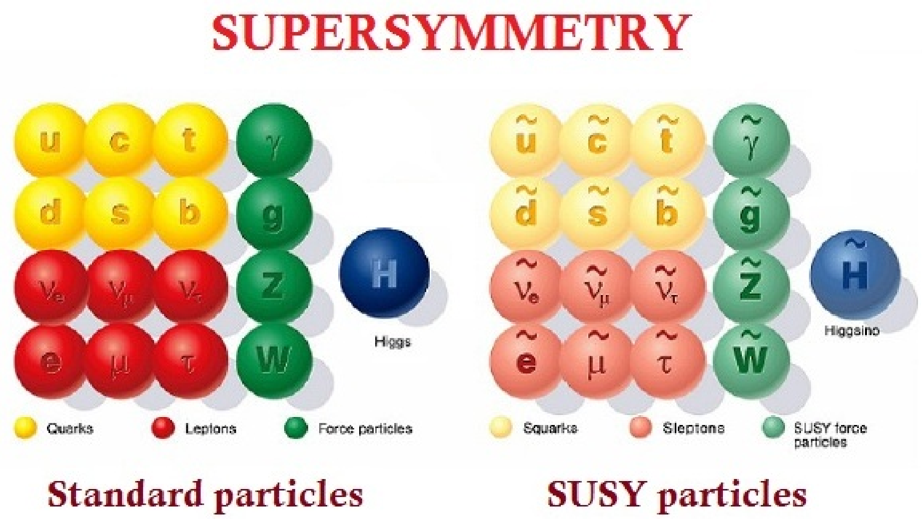

Figure 1: Table of the SM and SUSY particles

1. SUSY introduction

The Standard Model (SM) is the current best model we have to explain fundamental interactons. Some of its predictions have been verified with very high precision, such as the $g-2$ factor of the electron or the existance of the Higgs boson. Nevertheless, there are still phenomena that the SM ultimately fails to explain, like the presence of dark matter or the masses of nuetrinos. In order to solve these problems, physicists came up with several theories which try to extend the standard model. One of the most popular one is Supersymmetry (SUSY). The main feature of SUSY is a new symmetry included in the SM which predicts the existence of "supersymmetric" partners for each SM particle, listed in figure1. SUSY partners of bosons are predicted to have semi-integer spin, while partners of fermions have integer spin. At the moment, there is no evidence for SUSY at the LHC, meaning that the symmetry is broken. Even minimal models that tries to explain this phenomenon depend on the values of more than 100 parameters. The model we are going to study predicts that the two lightest SUSY particles are the partner of top quark, $\tilde{t}$ ($m_{\tilde{t}}$ = 900GeV), and the so-called lightest supersymmetric particle LSP (m_{LSP} = 100GeV, neutral and weakly interacting). All the other SUSY particles have higher masses, so that the only particles that we can hope to observe currently at LHC are these two. For this reasons, our search is focused on stop decay into top and LSP.

2. Analysis strategy

Since in our model the only SUSY particles light enough to be produced at LHC are the stop and the LSP, we are going to search for the former. We are going to individuate signal and background for our search and to generate via MC simulation our events, so that we can later on use these samples to train the SVM and study its performance.

2.1. Signal

Our main signal is due to stop pair production, followed by the decay into top and LSP(figure 2). The top quark decays almost 100% to a b quark and W boson. The

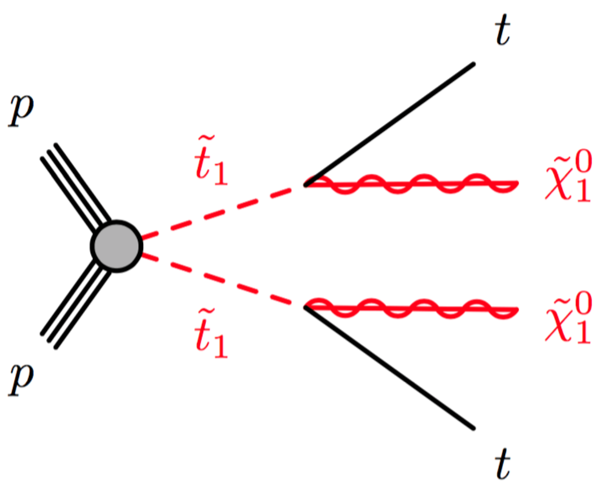

Figure 2: Stop decay

latter can either decay into a lepton + neutrino or hadronize.

2.2. Background

The main background for our signal is mostly due to top pair production. (figures 3 and 4).
The tops decay chain is the same mentioned previously.

2.3. Event selection

Our strategy is to select only events which have exactly one good reconstructed lepton as final state. In this way, the background mostly comes from 1-lepton decay of $t\bar{t}$ and dileptonic decay where one lepton has been lost. We are not going to consider events with a final τ, sicne their reconstruction is hard.

2.4. Event generation

Once we have chosen the events we want to analyze, we can generate them via Monte Carlo simulations. The programs used for event generation are PYTHIA8 [1] and DELPHES3 [2].
The processes simulated with Pythia8 are $q\bar{q}$ or gg into a quark/squark pair. For the Delphes setup we used the CMS default card.

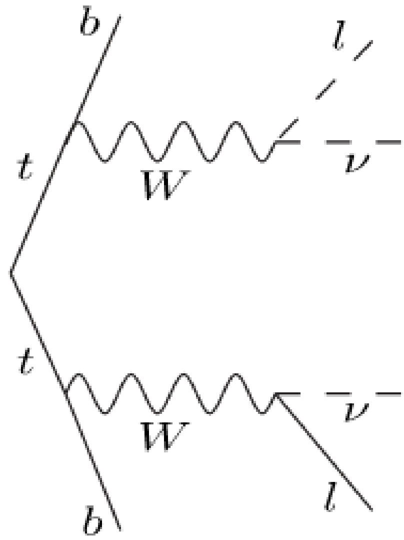

Figure 3: $t\bar{t}$ dileptonic decay

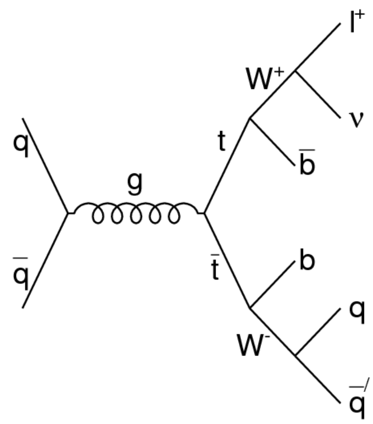

Figure 4: $t\bar{t}$ 1-lepton decay

2.5. Choice of the variables

In order to separate signal from background we need to study the dsitribution of physical quantities which are different in the two different cases. The main differences in signal and background events are mostly due to two features:

- The stop decays into LSP: since LSP is predicted to be neutral and weakly interacting, it is not going to be detected, thus leading to high missing energy.
- The stop is heavy: "invariant-mass"-like variables are expected to be distributed into high mass zones. Also, we expect the events to be more centrally distributed

	Variable
low-level	$p_{\mathrm{T},l}$
	η_l
	$p_{\mathrm{T},jet(1,2,3,4)}$
	$\eta_{jet(1,2,3,4)}$
	$p_{\mathrm{T},b\,jet1}$
	$\eta_{b\,jet1}$
	n_{jet}
	$n_{b\,jet}$
	$\not{E}_{\mathrm{T}}$
	H_{T}

high-level	m_{T}
	m_{T2}^{W}
	$\Delta\phi(W, l)$
	$m(l, b)$
	Centrality
	Y
	H_{T}-ratio
	$\Delta \mathrm{r}_{\min}(l, b)$
	$\Delta\phi_{\min}(j_{1,2}, \not{E}_{\mathrm{T}})$

Figure 5: Table of the chosen variables

$\lvert\eta_l\rvert$	$<$	2.4
$p_{T,l}$	$>$	30 GeV
$p_{T,jets}$	$>$	40 GeV
$p_{T,jet1}$	$>$	80 GeV
$p_{T,jet2}$	$>$	60 GeV
$\not{E}_T$	$>$	200 GeV
H_T	$>$	300 GeV
n_{jet}	$>$	3
n_{bjet}	$>$	0

Table 1: Cuts required for events to be selected (after Delphes level)

in the detector.

A complete list of the variables used in the analysis can be found in 5. The variables can be divided in two subsets:

- Low level variables: physical quantities that can be measured directly within the decector
- High level variables: mathematically more complex, chosen because of physical motivations.

The definition of the chosen variables can be found in appendix A. In order to eliminate immediately some background, we apply a preselection on the low level variables, shown in table 1. The events that passed the selection were organized in 100 files, each containing 10^4 signal and 10^4 background events.
Plots for a selection of low level variables can be found in figure 6, while in figure 7 are plotted some of the high level variables used. Notice that the histograms were plotted

for the same number of signal and background events (10^5 each): we haven't taken into account the much smaller stop cross section.

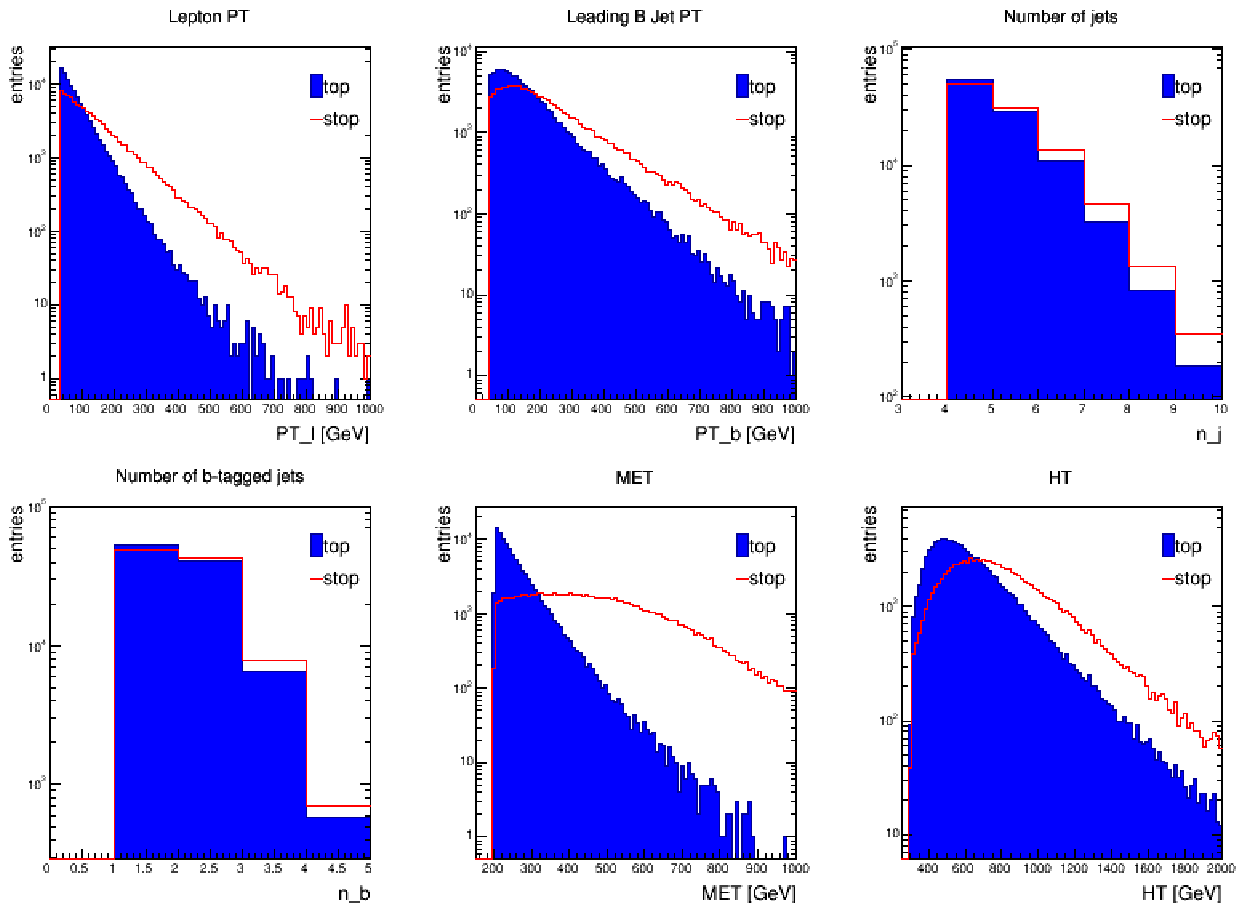

Figure 6: Selection of low level variables

2.6. Weighting the events

In our files we have the same number of signal and background events, but this is of course not what is going to happen in a real data set. We need to calculate how many events we expect to observe in the CMS detector for each event in our data set, both for signal and background. The first thing we need to do is to calculate the total number N of signal and background events observed in total. We can easily calculate this quantity through the formula

$$N = \epsilon \cdot \sigma \cdot L \tag{1}$$

where ϵ is the detector efficiency, σ the cross section of the process, L the integrated luminosity. In table 2 the values used for both signal and background are given. We are also interested in calculating how many real events correspond to a single event in our samples (the *weights* w, last column of the table 2). The calculations of the weight is

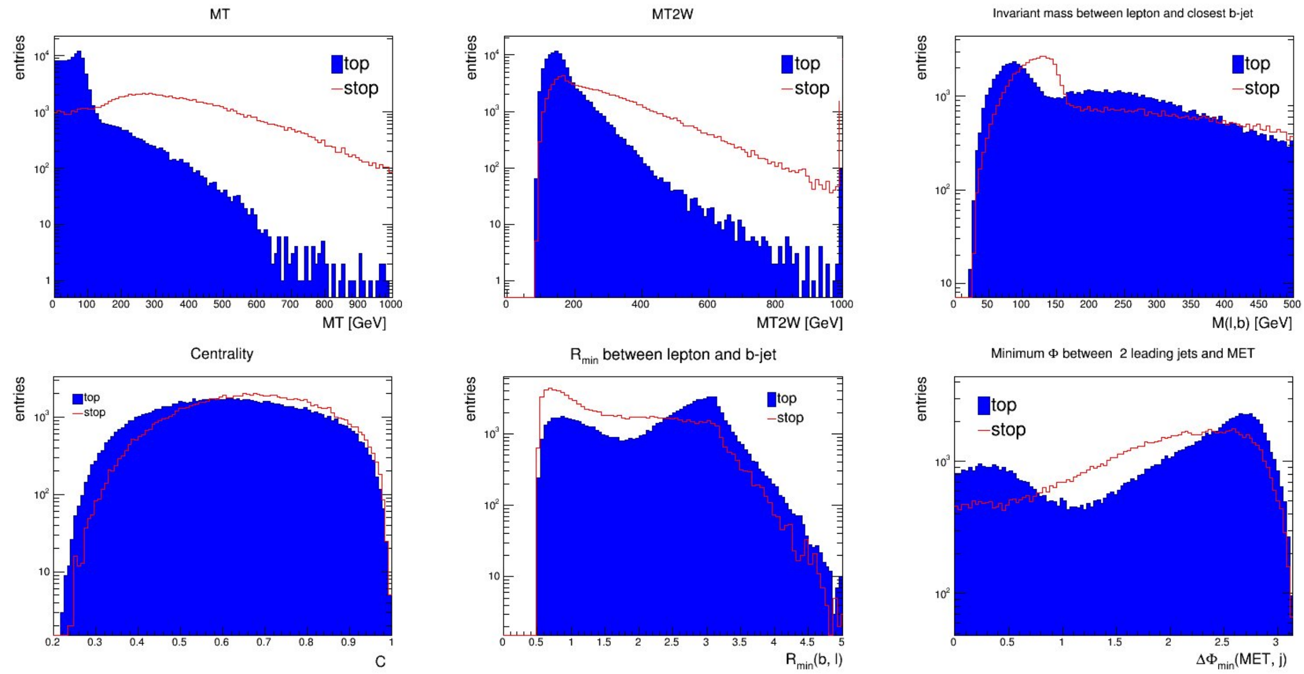

Figure 7: Some high level variables

done by a simple proportion, dividing N by the number of events of each type in the sample (10^4). Notice that in our case the weight of the signal is much lower than the weight of the background.

type	ϵ	σ(pb)	$L(\mathrm{fb}^{-1})$	w
top	$8.2 \cdot 10^{-4}$	844	300	20.7
stop	0.059	0.0177	300	0.031

Table 2: Values used in weight calculations

The efficiency was calculated by taking the ratio between the number of the remaining events after the cuts and the number of events generated on Pythia level. The cross sections for stop and top pair production were taken from [7]and [6] respectively. The integrated luminosity was taken as 300fb^{-1}, according to what expected for the end of the LHC run in year 2023.

3. Support vector machine

3.1. Motivation

Support vector machine (SVM) is a method used in machine learning to classify the points of a certain data seta into known labels (for example, signal and background). First, the machine must be trained on a *training set*, whose points have known labels.

Our N points (in our case the *events*), are characterized by n different real quantities (called *features*), so that they can be represented as points of $\mathbb{R}^n$. Since we generated them via MC, we know which event is a signal and which belongs to the background. Thus we can give them a label $y = \pm 1$ which tells us the class the point belongs to. We wish to find a way, if there is any, to separate the two classes. The easiest one is to use a $n-1$ dimensional hyperplane to divide the two sets, one on each side of the plane. Once we have found this plane, we can build a decision function to predict the label of a new point not belonging to the training set.

3.2. Linearly separable data

Let's assume at first that our data set is *linearly separable*, meaning that actually it exists at least a plane dividing the two classes, as in figure 8. We wish to find the best plane which divides the two classes. What do we mean by best separating plane? We

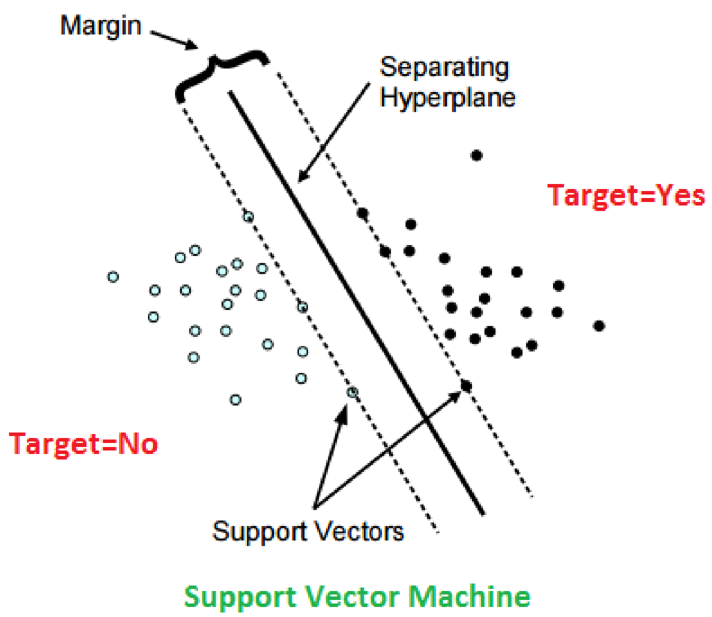

Figure 8: Set of linearly spearable points with 2 features

have to formulate the problem in a mathematically rigorous way. First of all, let the plane satisfy the equation:

$$\vec{w} \cdot \vec{x} + b = 0 \tag{2}$$

We want the best separating plane to have two properties:

- Maximizing the margin between the two classes (*principle of minimal risk*).
- Have points from different classes on different sides

To satisfy the first condition, we first have to define "the edge" of a class. We do so by picking a certain subset of the points, the so called *support vectors* (also indicated in figure 8). We can assume without loss of generality that the support vectors satisfy the relation

$$\vec{w} \cdot \vec{x_\pm} + b = \pm 1 \tag{3}$$

where the signs on the right hand side tells us which side of the hyperplane the support vector is. It's natural then to consider as the quantity to maximize the *margin* ρ between the support vectors:

$$\rho(\vec{w}, b) = \frac{\vec{w} \cdot x_+}{|w|} - \frac{\vec{w} \cdot x_-}{|\vec{w}|} = \frac{2}{|\vec{w}|} \tag{4}$$

Maximizing ρ is of course equivalent to minimizing $|\vec{w}|^2$. However the choice of $\vec{w}$ is not completely arbitrary! The second condition in 3.2 indeed is equivalent to require that for every point the following relationship must hold:

$$y_i(\vec{w} \cdot \vec{x} + b) \geq 1 \tag{5}$$

We have then a minimum problem with contraints. This can be treated with lagrangian formalism, minimizing the quantity

$$\mathcal{L} = \frac{1}{2}|\vec{w}|^2 - \sum_i^N \alpha_i(y_i(\vec{x_i} \cdot \vec{w} + b) - 1), \quad \alpha_i \geq 0 \tag{6}$$

Since we are looking for a stationary solution, we can set to 0 the derivatives of the lagrangian respect to $\vec{w}$ and b. By doing so we get the so called *dual lagrangian*, which is function of the Lagrange multipliers α_i:

$$\mathcal{L} = -\frac{1}{2}\sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \vec{x_i} \cdot \vec{x_j} + \sum_{i=1}^N \alpha_i \qquad \alpha_i \geq 0, \quad \alpha_i(y_i(\vec{w} \cdot \vec{x_i} + b - 1) = 0 \tag{7}$$

where N is the number of points used for training. Notice that in this new formulation, only the support vectors have non null α_i.
Once we have found the α_i which satisfy the minimum conditions, we can build a decision function $\hat{f}$ to predict the class $\hat{y}$ of a new event $\vec{u}$ not belonging to the training sample.

$$\hat{y} = \hat{f}(\vec{u}) = \mathrm{sign}(\sum_{k=1}^{N_{SV}} y_k \alpha_k \vec{x_k} \cdot \vec{u} - b) \tag{8}$$

where the sum is restriced to support vectors (the other α_i are null!).

3.3. Overlapping data

What if there is no separating hyperplane, as in figure 9? The best way to solve the

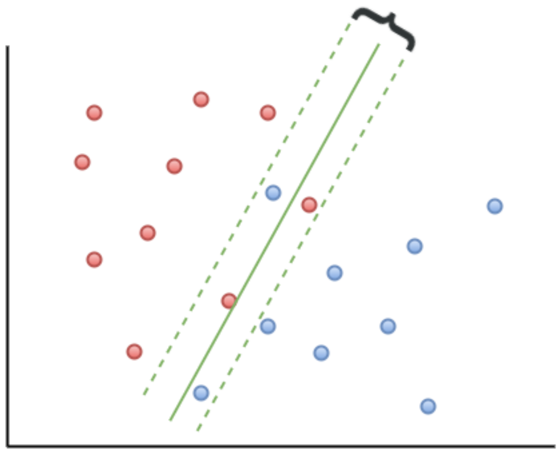

Figure 9: Overlapping classes: no separating hyperplane exists!

problem is to allow some points of different classes to be on the same side of the plane. In order to do so, we must modify the condition 5:

$$y_i(\vec{w} \cdot \vec{x_i} + b) \geq 1 - \xi_i \tag{9}$$

and introduce in the lagrangian a penalty term proportional to the "slackness": $C \sum_i \xi_i$ $(C, \xi_i \geq 0)$. The lagrangian then becomes:

$$\mathcal{L} = \frac{1}{2}|\vec{w}|^2 + C \sum_i \xi_i - \sum \alpha_i (y_i(\vec{w} \cdot \vec{x} + b) - 1 + \xi_i) + \sum \beta_i \xi_i \tag{10}$$

Doing the same maths of the previous section, we can arrive at the following dual lagrangian:

$$\mathcal{L} = -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x_i} \cdot \vec{x_j} + \sum \alpha_i \qquad C \geq \alpha_i \geq 0 \tag{11}$$

It's very important to realize that C is a free parameter that must be set prior the calculations for the best plane: choosing the right value will be an important part of training the SVM.

3.4. Non linearly separable data

Let's assume now that the data is distributed according to figure 10. We can transform the problem into a linear one if we map our feature space in a different feature space through a continuous mapping Φ. The dual lagrangian then becomes

$$\mathcal{L} = -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\vec{x_i}, \vec{x_j}) + \sum \alpha_i \qquad C \geq \alpha_i \geq 0 \tag{12}$$

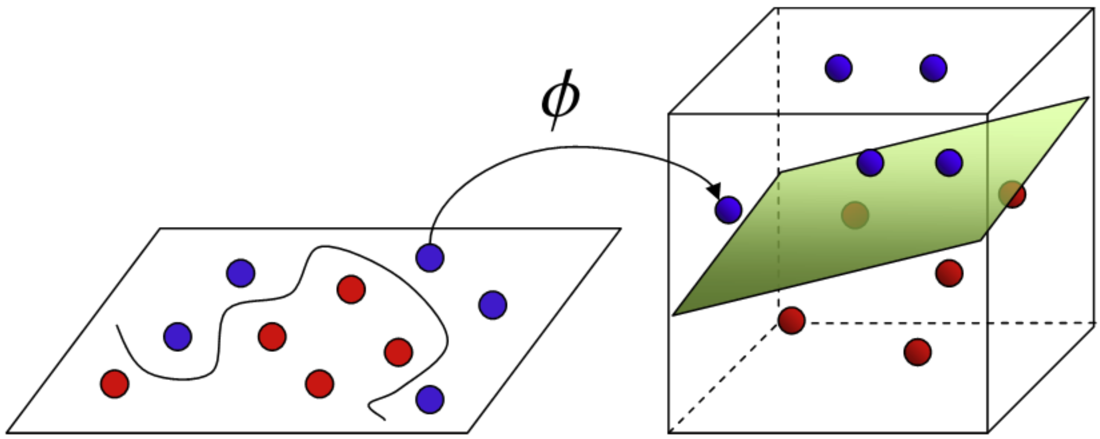

Figure 10: The data can be separable with a proper transformation

where $K(\vec{x_i}, \vec{x_j}) = \Phi(\vec{x_i}) \cdot \Phi(\vec{x_j})$ (also called *kernel*). We can then apply what learnt in the previous sections and minimize the lagrangian in this new space! The problem with this kind of approach is that we usually don't know which mapping will work for us. We can overcome this problem by noticing that in the dual lagrangian we only have the scalar product. If we can find somehow an expression for $K(\vec{x_i}, \vec{x_j})$, we can ignore finding the exact mapping Φ. There are many choices for the kernel. The most popular is the so called RBF kernel (*radial basis function*):

$$K(\vec{x_i}, \vec{x_j}) = \exp(-\gamma|\vec{x_i} - \vec{x_j}|^2) \tag{13}$$

There is no formal proof of why this choice works almost every time, but the idea is that the problem is mapped into an infinite dimensional Hilbert space, where hopefully there will be enough dimensions to find a separating hyperplane. Notice that also γ is a free parameter that must be set prior the training. Obviously, in this case the decision function will contain the chosen kernel:

$$\hat{f}(\vec{u}) = \text{sign}(\sum_{k=1}^{N_{SV}} y_k \alpha_k K(\vec{x_k}, \vec{u}) - b) \tag{14}$$

3.5. Probabilistic outcome

The decision functions we have built until now have only a binary outcome: either a point is in one class or in the other. We can improve this model by trying to get out not only the class prediction, but a probability p that the point belongs tothe predicted class. An obvious requirement for this new decision function is that points very far away from the separating plane must have p close to 1 (if they are on the correct side), while the ones near the plane must have lower p, since we are less sure about their class. According to this, a nice model for the probability estimate is to take a sigmoid function (ranging from 0 to 1 for obvious reasons) of some sort of indicator of the distance of the point from the separating plane. This task can be easily accomplished by the decision function $\hat{f}$, so we will have as a model for the probability the following:

$$p(1|\hat{f}) = \frac{1}{1 + \exp\left(A + B\hat{f}\right)} \tag{15}$$

where A, B are parameters fitted from the training set.

3.6. Parameter tuning

In our analysis we are going to use SVM with RBF kernel and allowing for overlap. How to set the values of the two free parameters C, γ? We want to choose the pair which somehow maximizes the performance of the SVM. To do so we first must define a way to measure the performance by choosing adequate figures of merit.

3.7. Figures of merit

In machine learning, there are many popular variables used to study the performance of the implemented algorithm. We will focus on two particular figures of merit: accuracy and Asimov significance. Accuracy is defined as the ratio between the correctly predicted events and the total number of events in a given evaluation sample. Notice that accuracy doesn't really take into account the true number of signal/background events seen in a real data set, but depends only on the number of events in the evaluation sample (in other words: it does not depend on the luminosity). Asimov significance instead is an approximate form of discovery significance for a Poisson-distributed background (see [8]).

$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)}\right]\right)\right]^{1/2} \tag{16}$$

It can also account for uncertainty σ_b in the background estimate b. The s, b in 16 are calculated in the following way:

- If a true signal event of the evaluation sample is classified as signal, we add to the s value in eq. 16 the signal weight, calculated in section 2.6.
- If a true background event of the evaluation sample is predicted as signal, we add to the b the background weight.

The idea behind this definition is that we are going to have some events predicted as signal in a real CMS data set. In the only-background hypothesis we would see b number of events predicted as signal. Z_A calculates how far away we are from this hypothesis (if a signal exists) by taking into account the s term!

3.8. Probability cut

When calculating the Asimov significance, we can improve the signal/background ratio by considering in the calculations only the events that are predicted as signal over a certain probability threshold p. In this way we can vastly reduce the number of background events predicted as signal in the b calculation. In figure 11 are shown the numbers of signal s and background b events (after weighting) predicted as signal as a function of value of p. In this case the training has been done on a fixed (C, γ) pair. As one can see, while s is almost constant, b decreases up to a order of magnitude, so this trick helps us getting higher values of Z_A. In figure 12 one can see how this is still true

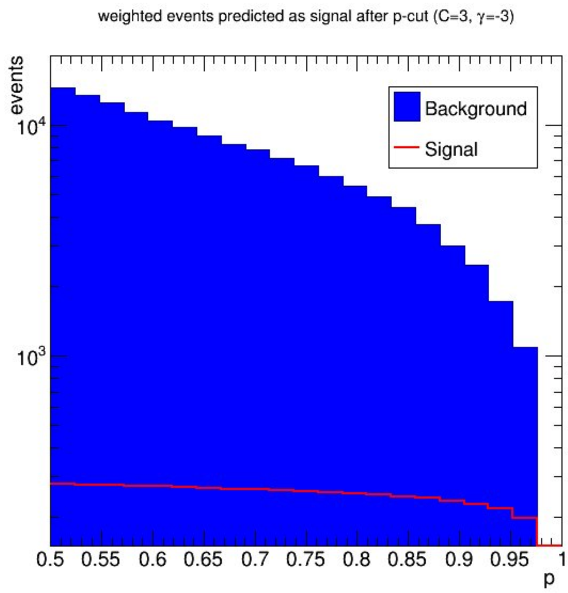

Figure 11: Number of events predicted as signal in function of the p threshold.

for almost every (C, γ) pair.

4. Training and performance

Once we have generated the sample files, we can start training the SVM and study its performance. As told in 3.7, we are going to use as figures of merit accuracy and Asimov significance. The training of the SVM has been done using the LIBSVM library and tools ([9]). In the appendix B are given the results also for $\frac{s}{\sqrt{s+b}}$: although in our case there are less physical reasons to use this figure of merit rather than Z_A (which is the statistically correct choice), it is still a popular performance measure (especially in BDT analysis).

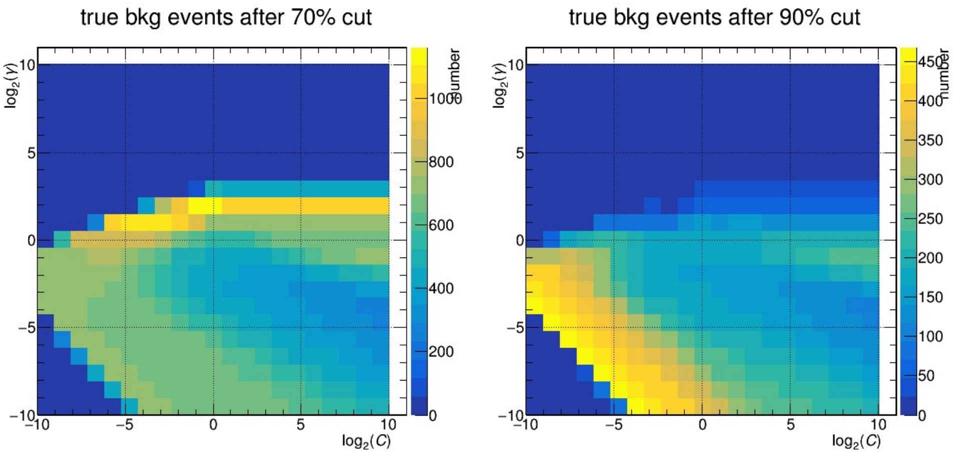

Figure 12: Number of weighted background events predicted as signal after 70% and 90% p-cut as function of C, γ on a fixed evaluation sample.

4.1. Parameter search

4.1.1. Accuracy

In order to find the best parameter pairs, we are going to implement a brute force grid search on a $\log_2$-spaced lattice in the (C, γ) plane, and see how the SVM trained with each of the lattice point scores on a given evaluation sample (called evaluation sample A). We always do the training on a fixed training sample and fixed evaluation sample (10^4 signal, 10^4 background events, as always in our study). In figure 13 are reported the results for the accuracy as a function of the different lattice points. There is a large zone in the parameter space where the SVM gets high accuracy values (around 90%). Outside this region, the SVM doesn't work properly and the accuracy quickly goes down to 50% value: in this scenario the SVM predicts almost every event of the evaluation sample to be of the same class (thus being correct only half of the time).

4.1.2. Z_A

The Asimov significance values after different probability cuts for different parameter values are given in figure 14. We have set the relative uncertainty on background equal to 10% in the significance calculations. The Asimov significance in the high accuracy region is around 0.8. This isn't a very high value, but it is something we should have expected. If we assume for a 90% accuracy SVM that it gets 90% of the time correct prediction for both signal and background, the weighted values for s and b with a 10% uncertainty give us a Z_A in the order of unity. It is also worth to notice that the best scoring (C, γ) is in the region of the paramter zone where the SVM accuracy begins to go down.

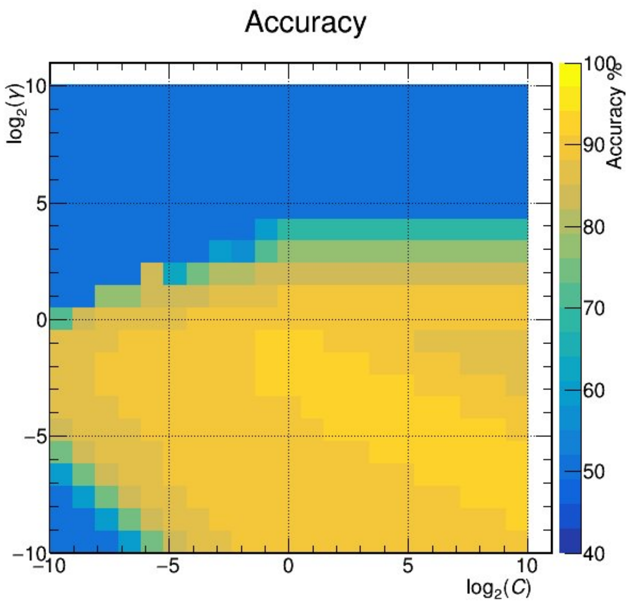

Figure 13: Accuracy as a function of C, γ on evalation sample A.

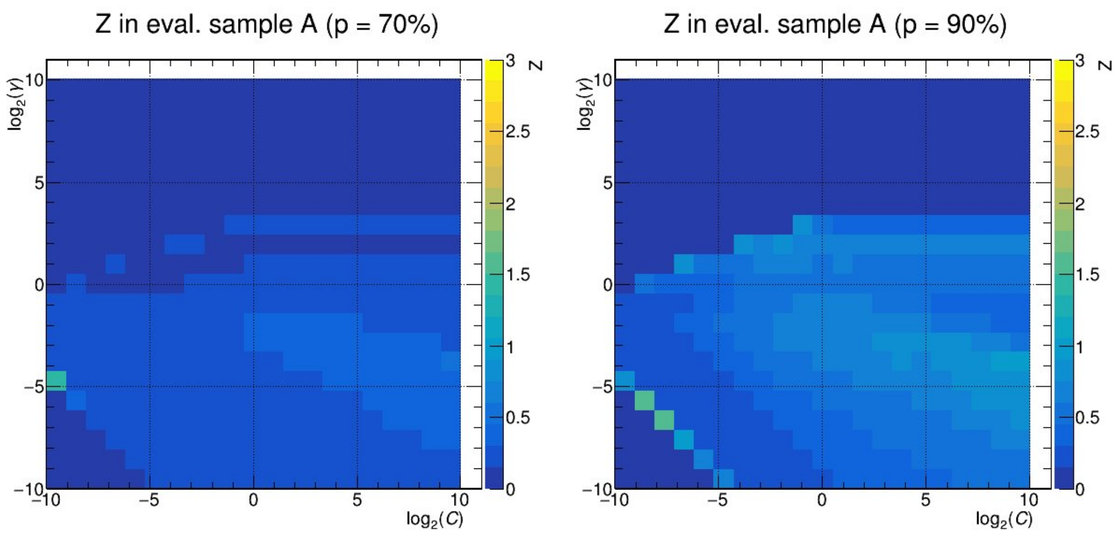

Figure 14: Asimov significance after 70% and 90% probability cut as a function of C, γ (evaluation sample A)

4.2. Replicability on different evaluation samples

With the same training sample, we would like to see if the best scoring (C, γ) for accuracy in evaluation sample A is also the best scoring pair on different evaluation samples. We picked the $(-2, 2)$ pair ($\log_2$ values) for testing accuracy (90% for the evaluation sample A). On the remaining 98 evaluation samples, the accuracy was always around 90%. So accuracy is really something that doesn't depend in a significant way on the chosen

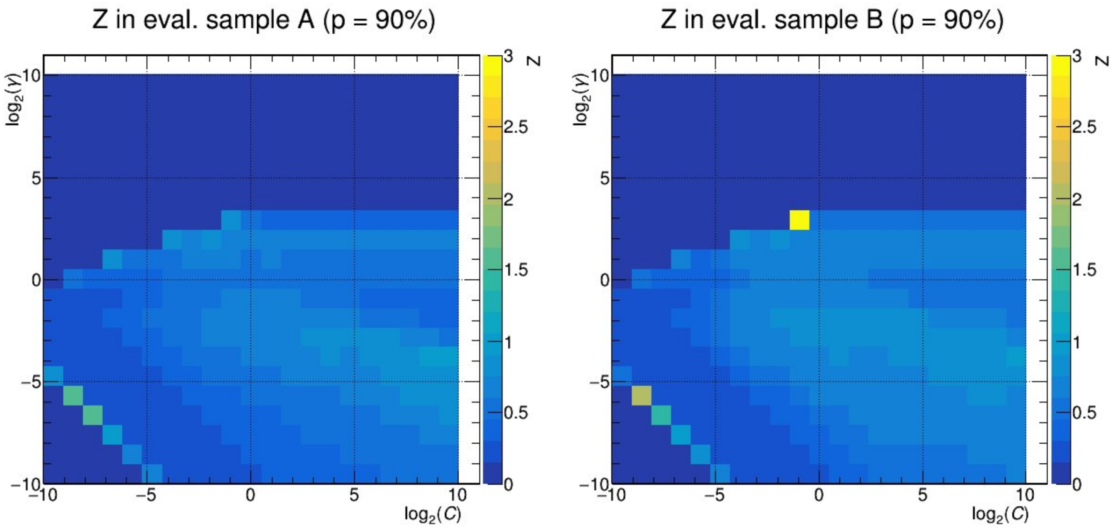

Figure 15: Comparison on the Z results as function of C, γ in different evaluation samples with p = 90% cut.

evaluation sample. For Asimov significance we have to be more careful. If we pick the best scoring pair on a given evaluation sample, it is not necessarily going to be a good one for another sample. Let's take as an example the results in figure 15. As one can see, in the evaluation sample B the (-1, 3) pair ($\log_2$ values) is really high scoring for Asimov significance ($Z_A = 4.0$), but on evaluation sample A its performance is not good. The main cause for this discrepance in different samples is due to the fact the the best scoring pair for the significance is in the region of not so high accuracy: there are very few events left after a sufficiently high probability cut (both signal and background), thus leading to artifically high and not easily replicable values of Z_A. On the other hand, if we pick the pairs in the high accuracy region, they are going to score approximately the same significance in all the different evaluation samples. So the proper approach to replicate the significance scoring is to avoid the "artificially" best performing parameter pairs by excluding the ones belonging to the low accuracy region of the parameter space.

5. Conclusions

We have generated the samples for our stop search and chosen tha variables to use for the analysis. We have then given a brief overview on SVM method. We have trained the SVM on the same sample with different (C, γ) values. We observed high values for accuracy, and Z_A around unity. We also checked that the good (C, γ) pairs for accuracy do not depend on the chosen evaluation sample, while the best scoring pair for Z_A strongly depends on the sample. The discrepancy is due to the fact that only results of pairs in the good accuracy zone can be replicated.

A. Variables definitions

A.1. Low level variables

As low level variables we have:

- the pseudorapidity η_l and transverse momentum $p_{T,l}$ of the selected lepton
- η, p_T of the 4 leading jets
- η, p_T of the leading b jet (even if it might have appeared among the 4 leading jets).
- the number of jets n_{jets} and b-tagged jets $n_{b\text{jets}}$
- the scalar sum H_T of the module of the transverse momentum of the selected jets
- the missing energy in the transverse plane $\not{E}_T$

A.2. High level variables

As high level variables we have:

- the transverse mass m_T, defined as $\sqrt{2p_{T,l}\not{E}_T(1-\cos\Delta\phi(l,\not{E}_T))}$.
- m_{T2}^W. Its value is the minimal mass of the mother particle compatible with the event in the topology shown in figure 3. Further information is given in [3] and [4]. For the m_{T2}^W calculation we used the bisection algorithm given in [5]
- $r_{min}(l,b)$, which is the distance in the minimum distance (η,ϕ) plane between the lepton and a b-jet.
- $m_{(l,b)}$ is the invariant mass between the lepton and the closest (r-wise) jet.
- Centrality $C = \sum_{lep,jets} p_T / \sum_{lep,jets} p$.
- $\not{E}_T$ significance or $Y = \frac{\not{E}_T}{\sqrt{H_T}}$
- H_T-ratio $= \frac{\sum_{jets\in A} p_T}{H_T}$, where A is the set of jets which lie in the same half of the transverse plane as the $\not{E}_T$.
- $\Delta\phi(W,l)$, which is the angle in the transverse plane between the lepton and the $\not{E}_T$.
- $\Delta\phi_{min}(j_{1,2},\not{E}_T)$, which is the minimum angle in the transverse plane between the missing energy and the two leading jets.

B. $s/\sqrt{s+b}$ performance

In figure 16 are given the results for $s/\sqrt{s+b}$ with different probability cut on the same evaluation sample.

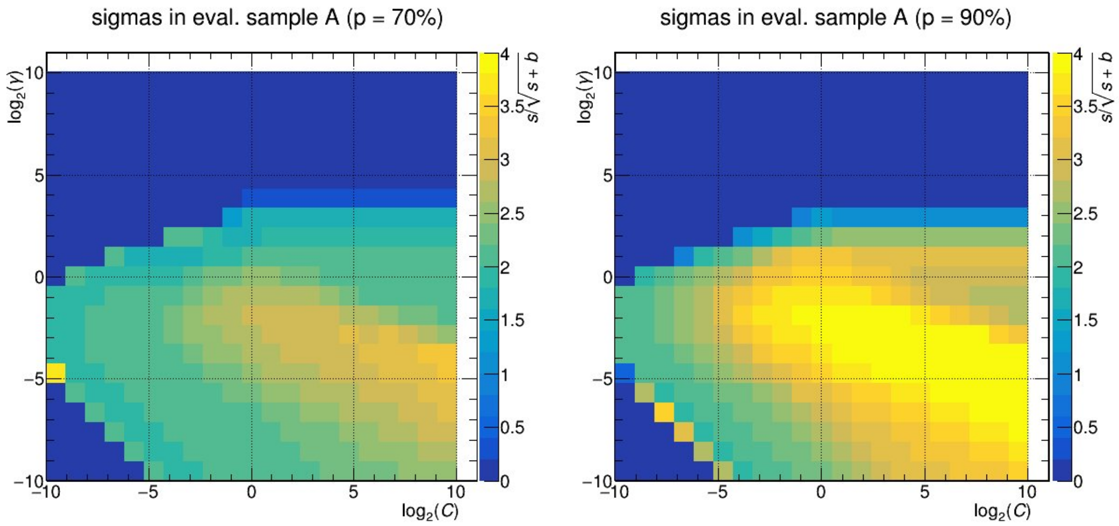

Figure 16: $s/\sqrt{s+b}$ in evaluation sample A after 70% and 90% cut.

In figure 17 are given the results for different evaluation samples. Notice that also for this figure of merit one can only replicate the results that are well inside the good accuracy zone.

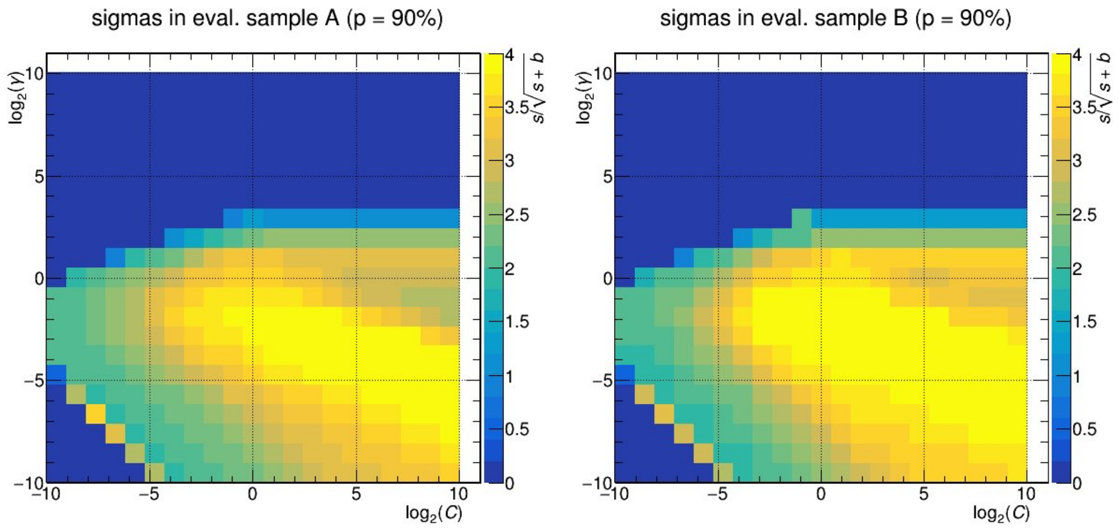

Figure 17: $s/\sqrt{s+b}$ in different evaluation samples

Acknowledgements

Thanks to my supervisor Dirk Krücker for his patience and helpful advice.
Special thanks also to Brian, Ceren, Felix, Lorenzo, Meadhbh for having been fantastic mates in this months.

References

[1] T. Sjöstrand, S. Mrenna and P. Skands, JHEP05 (2006) 026, Comput. Phys. Comm. 178 (2008) 852.

[2] JHEP 02 (2014) 057 [arXiv:1307.6346 [hep-ex]]

[3] H.C. Cheng, Z. Han, JHEP 0812:063,2008, arXiv:0810.5178v2 [hep-ph]

[4] Y. Bai, H.C. Cheng, J. Gallicchio, J. Gu, JHEP 1207 (2012) 110, arXiv:1203.4813v1 [hep-ph]

[5] https://cmssdt.cern.ch/lxr/source/PhysicsTools/Heppy/src/mt2w_bisect.cc

[6] U. Langenfeld, S. Moch, P. Uwer, Phys.Rev.D80:054009,2009, arXiv:0906.5273 [hep-ph]

[7] C. Borschensky, M. Krmer, A. Kulesza, M. Mangano, S. Padhi, T. Plehn, X. Portell, arXiv:1407.5066 [hep-ph]

[8] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, European Physical Journal C 71 (Feb., 2011) 1554, arXiv:1007.1727 [physics.data-an].

[9] Chang, Chih-Chung and Lin, Chih-Jen, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology* **2** (2011) 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`