Inference with the Universum

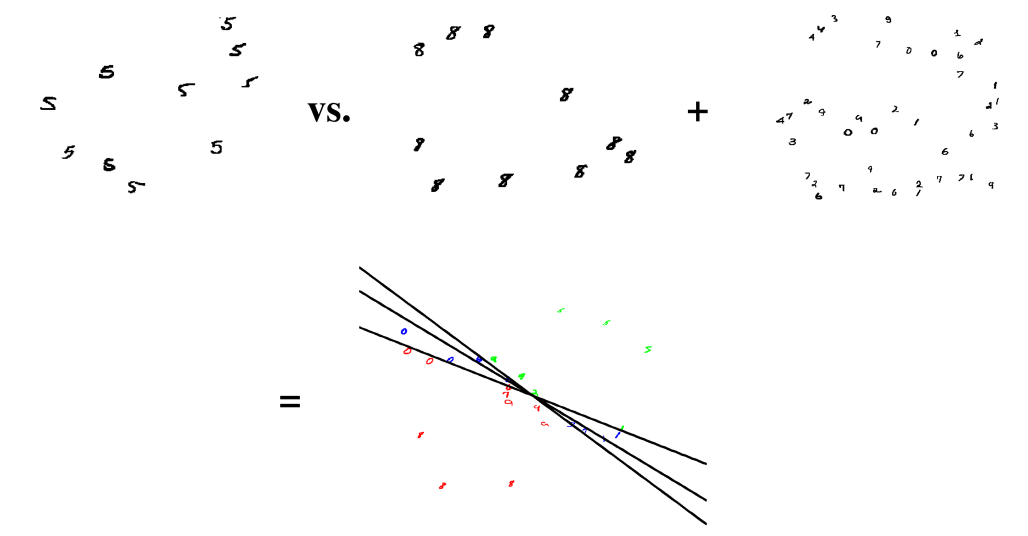
Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, Vladimir Vapnik

NEC Labs America (Princeton, NJ), **Max Planck Institute for Biological Cybernetics** (Tübingen, Germany)

BIOLOGISCHE KYBERNETIK

Introduction

What is inference with the universum about?

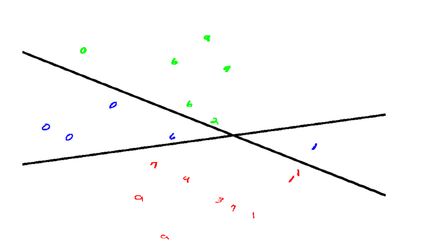


Most regularizers are agnostic to specific data distributions

- Given a *data distribution* $\mathcal{P}$ and a *function class* $\mathcal{F}$ to choose a decision function from, find a function that has minimal error on the training data and generalizes well.
- The decision function is found by means of an *optimization problem*, where the *empirical error* is minimized together with a *regularizer* that controls the generalization error.
- While the choice of $\mathcal{F}$ influences the regularizer and the empirical error, $\mathcal{P}$ effects only the empirical risk minimization: *Most regularizers are agnostic to the distribution $\mathcal{P}$ given by the data at hand.*

How can we incorporate prior knowledge in the regularizer? Given data $(x_1,y_1),\ldots,(x_m,y_m)$ and $x_{m+1},\ldots,x_{m+k}$ and the set of equivalence classes $\boldsymbol{F}=\{[f_1],\ldots,[f_r]\}$ on $\mathcal{F}$:

- **MAP**: Define a *prior P* over $\mathcal{F}$ and choose $[f_i]\in\boldsymbol{F}$ that has minimal empirical error and maximises $\int_{[f_i]} dP(f)$
- **Universum [Vapnik, 1998]**: Use another set $\mathfrak{U}=\{x_1^*,\ldots,x_{|\mathfrak{U}|}^*\}$ to measure the "quality" of F_i (call this set *Universum*)

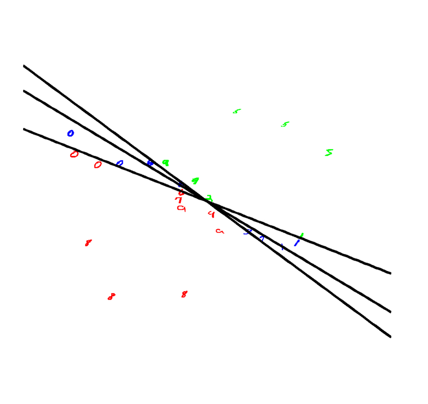


- Use of a priori information in $\mathfrak{U}$: Choose a $[f^*]\in\boldsymbol{F}$ that *has low empirical risk* and has a *maximum number of contradictions* on $\mathfrak{U}$, i.e. $\max|\{x\in\mathfrak{U}|\exists g,h\in[f]:g(x)h(x)<0\}|$.
- $\mathfrak{U}$ is from the *same domain* and *same problem category*, but *not* from the same distribution.

- In contrast to semi-supervised learning, $\mathfrak{U}$ is *not from the same distribution* and in contrast to the virtual support vector method or noise injection $\mathfrak{U}$ *does not need to be labeled*.
- $\mathfrak{U}$ reflects prior knowledge about the *admissible set of examples* whereas a prior over functions represents prior knowledge about the *admissible set of decision functions*.
- Universum examples can be *constructed* or *collected* in many problem settings

Approximation and Implementation

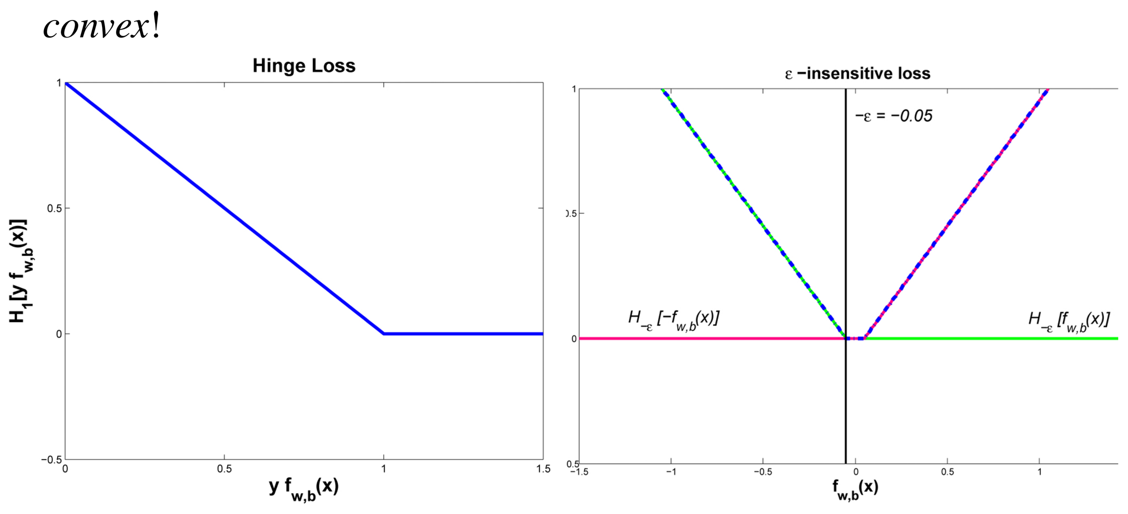
Approximation of Contradiction Maximization on $\mathfrak{U}$



- Approximate maximization of contradictions by putting $x^*\in\mathfrak{U}$ close to the decision boundary $f_{w,b}=\langle \boldsymbol{w},\cdot\rangle+b$
- A small change in $f_{w,b}$ will cause a contradiction on x_i^*
- Choose $f_{w,b}\in\mathcal{F}$ with minimal real valued output on x_i^*

Implementation in Support Vector Machines: The $\mathfrak{U}$SVM

- Set of labeled examples: $\mathcal{L}=\{(x_1,y_1),\ldots,(x_{|\mathcal{L}|},y_{|\mathcal{L}|})\}$
- Set of universum examples: $\mathfrak{U}=\{x_{|\mathcal{L}|+1},\ldots,x_{|\mathcal{L}|+|\mathfrak{U}|}\}$
- Express all loss functions in terms of Hinge loss $H_a[t]=\max\{0,a-t\}$
- Use Hinge loss $H_1[yf_{w,b}(x)]$ for each labeled example $x\in\mathcal{L}$ as in standard SVM
- Use ε-insensitive loss $U_\varepsilon[f_{w,b}(x)]=H_{-\varepsilon}[f_{w,b}(x)]+H_{-\varepsilon}[-f_{w,b}(x)]$ for each universum example $x\in\mathfrak{U}$. Note that applying $U_\varepsilon[f_{w,b}(x)]$ to an example x is equivalent to applying $H_{-\varepsilon}[f_{w,b}(x)]$ to two identical copies of x with opposite labels.
- All loss functions are convex, therefore the optimization problem is *convex*!



$\mathfrak{U}$SVM primal formulation:

$$\text{minimize}_{w,b}\ \frac{1}{2}||\mathbf{w}||_2^2+C\sum_{i=1}^{|\mathcal{L}|}H_1[y_i f_{w,b}(x_i)]+C_{\mathfrak{U}}\sum_{j=1}^{|\mathfrak{U}|}U_\varepsilon[f_{w,b}(x_{|\mathcal{L}|+j})]$$

$\mathfrak{U}$SVM dual formulation:

- For $i=1,\ldots,|\mathfrak{U}|$ set

$$(x_{|\mathcal{L}|+i},y_{|\mathcal{L}|+i})=(x_{|\mathcal{L}|+i},+1)$$
$$(x_{|\mathcal{L}|+|\mathfrak{U}|+i},y_{|\mathcal{L}|+|\mathfrak{U}|+i})=(x_{|\mathcal{L}|+i},-1)$$

- The dual formulation is identical to the dual formulation of a standard SVM except for the linear part of the objective function:

$$\max_{\alpha}\ \sum_{i=1}^{|\mathcal{L}|+2|\mathfrak{U}|}\rho_i\alpha_i-\frac{1}{2}\sum_{i,j=1}^{|\mathcal{L}|+2|\mathfrak{U}|}y_iy_j\alpha_i\alpha_j(x_i\cdot x_j)$$

$$\text{s.t.}\ \begin{cases}0\le\alpha_i\le C & \text{for } i=1\ldots|\mathcal{L}|\\ \rho_i=1 & \text{for } i=1\ldots|\mathcal{L}|\\ 0\le\alpha_i\le C_{\mathfrak{U}} & \text{for } i=|\mathcal{L}|+1\ldots|\mathcal{L}|+2|\mathfrak{U}|\\ \rho_i=-\varepsilon & \text{for } i=|\mathcal{L}|+1\ldots|\mathcal{L}|+2|\mathfrak{U}|\\ \text{and}\ \ \sum_{i=1}^{|\mathcal{L}|+2|\mathfrak{U}|}y_i\alpha_i\ =\ 0\end{cases}$$

Experiments

MNIST

- Task on the MNIST dataset: Separate the class 5 from class 8
- Considered Universa:
  - (i) $\mathfrak{U}_{Noise}$ - images of "random noise" by generating uniformly distributed pixel features ("null hypothesis")
  - (ii) $\mathfrak{U}_{Rest}$ - the other digits 0-9 excluding 5 and 8
  - (iii) $\mathfrak{U}_{Gen}$ - create an artificial image by generating each pixel according to its discrete empirical distribution on the training set
  - (iv) $\mathfrak{U}_{Mean}$ - create an artificial image by first selecting a random 5 and a random 8 from the training set, and then constructing the mean of these two digits
  - (v) $\mathfrak{U}_i$ - class i of the remaining digits 0-9 excluding $i=5$ and $i=8$

- Results using universa (i)-(iv) with constant size of $|\mathfrak{U}|$:

Method	Training subset size			
	500	1000	2000	3000
SVM	1.96	1.38	0.99	0.83
$\mathfrak{U}_{Noise}$-SVM	1.95	1.37	0.99	0.82
$\mathfrak{U}_{Rest}$-SVM	1.60	1.10	0.75	0.55
$\mathfrak{U}_{Gen}$-SVM	1.72	1.17	0.81	0.64
$\mathfrak{U}_{Mean}$-SVM	1.68	0.99	0.73	0.57

- Results using universa (i)-(iv) with constant size of $|\mathcal{L}|$:

Train. examples	Number of Universum examples				
	500	1000	3000	5000	10000
3000	0.66	0.64	0.60	0.57	0.58

- Results using universa (v) with mean correlation ρ of elements in $\mathfrak{U}_i$ to digits 5 and 8:

$\mathfrak{U}$	Training subset size			Correlation	
	all	1000	200	ρ_5	ρ_8
$\mathfrak{U}_0$	0.27	0.97	3.03	0.32	0.29
$\mathfrak{U}_1$	0.16	1.01	2.95	0.24	0.36
$\mathfrak{U}_2$	0.21	0.94	3.21	0.24	0.34
$\mathfrak{U}_3$	0.05	0.62	2.97	0.33	0.37
$\mathfrak{U}_4$	0.21	0.93	3.03	0.27	0.32
$\mathfrak{U}_6$	0.16	0.84	2.40	0.26	0.32
$\mathfrak{U}_7$	0.16	1.08	3.23	0.25	0.30
$\mathfrak{U}_9$	0.21	0.89	2.78	0.30	0.37
$\mathfrak{U}=\emptyset$	0.21	1.19	3.03	-	-

Reuters & WinMac (20 newsgroups dataset)

- Task on the Reuter dataset: Separate the class $C15$ from the remaining classes in toplevel category *CCAT*
- Considered Universa:
  - Reuters:
  - (vi) $\mathfrak{U}_{M14}$ - class $M14$ from toplevel category *MCAT*
  - (vii) $\mathfrak{U}_{MoC}$ - mean of closest from 10 randomly sampled examples of each class
  - WinMac (20 newsgroups)
  - (viii) $\mathfrak{U}_{Mean}$ - create an artificial bag of words by first selecting one random example from each class and then constructing the mean of those two
- Results on Reuters using universa (vi)-(vii):

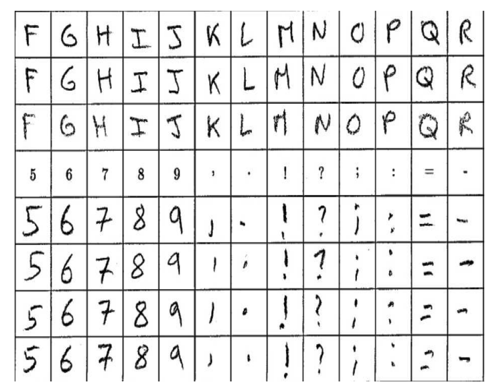
Method	Training subset size				
	50	100	200	500	1000
SVM	21.1	13.1	11.0	8.6	7.6
$\mathfrak{U}_{M14}$-SVM	15.7	12.7	10.2	8.2	7.6
$\mathfrak{U}_{MoC}$-SVM	19.4	12.6	10.8	8.6	7.6

- Results on WinMac using universe (viii):

Method	Training subset size				
	10	25	50	75	100
SVM	45.2	31.7	20.3	14.7	11.7
$\mathfrak{U}_{Mean}$-SVM	33.0	24.3	15.2	12.3	11.0

AbcdEtC

- We collected a new dataset consisting of upper and lower case letters, digits and symbols.
- Download at: http://www.nec-labs.com/∼jasonw/abcdetc/
- Task on AbcdEtc: Separate class "a" from "b"



- Considered universa:
  - (ix) $\mathfrak{U}_{Lowcase}$ - the set of lower case letters c-z
  - (x) $\mathfrak{U}_{Upcase}$ - the set of upper case letters C-Z
  - (xi) $\mathfrak{U}_{Digits}$ - the set of digits
  - (xii) $\mathfrak{U}_{Symbols}$ - the set of symbols
- Results on AbcdEtc using universa (ix)-(xii):

Method	Training subset size				
	20	50	100	150	200
SVM	9.93	5.71	5.16	4.53	3.85
$\mathfrak{U}_{Lowcase}$-SVM	8.75	5.09	4.21	3.89	3.39
$\mathfrak{U}_{Upcase}$-SVM	8.79	5.52	4.88	3.65	2.84
$\mathfrak{U}_{Digits}$-SVM	8.37	5.56	4.26	3.97	3.49
$\mathfrak{U}_{Symbols}$-SVM	8.62	5.75	5.17	4.40	3.67

Data Dependent Regularization

The universum algorithm can be seen as data dependent regularization for which the choice of a specific universum determines the kind of regularizer. Certain choices of universa can recover common regularizer.

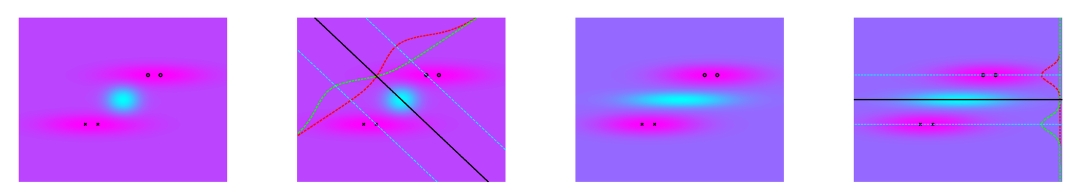
L_2 Regularizers

- For recovering the *isotropic L_2 regularizer* assume $b=0$, let $\mathfrak{U}_L:=\{x_k^*|\,x_{kj}^*=\delta_{kj},\,k=1,\ldots,n\}$ and use quadratic loss $U_{L_2}[f_{w,b}(x_i^*)]=|f_{w,b}(x_i^*)|^2$ for the points in U_L. Then:

$$\sum_{i=1}^{|\mathfrak{U}_L|}U_{L_2}[\,f_w(x_i^*)\,]=\sum_{i=1}^{|\mathfrak{U}_L|}(w\cdot x_i^*)^2=\sum_{k=1}^{n}w_k^2=||w||_2^2$$

- For recovering the *anisotropic L_2 regularizer* assume a universum with mean 0 and covariance matrix C. Then:

$$\sum_{i=1}^{|\mathfrak{U}|}U[\,f_{w,b}(x_i^*)\,]=\sum_{i=1}^{|\mathfrak{U}|}(\,w^\top x_i^*+b\,)^2=|\mathfrak{U}|\,(\,w^\top Cw\,+\,b^2\,)$$



L_1 Regularizer

- For recovering the *linear L_1 regularizer* assume $b=0$, use the same universum $\mathfrak{U}_L$ as for the isotropic L_2 regularizer and use L_1 *loss* $U_{L_1}[f_{w,b}(x_i^*)]=|f_{w,b}(x_i^*)|$ for the points in $\mathfrak{U}_L$. Then:

$$\sum_{i=1}^{|\mathfrak{U}_L|}U_{L_1}[\,f_w(x_i^*)\,]=\sum_{i=1}^{|\mathfrak{U}_L|}|w\cdot x_i^*|=\sum_{k=1}^{n}|w_k|=||w||_1$$

- A *Non-linear L_1 regularizer* is usually not possible because of the high dimension of the feature space, but using $\mathfrak{U}_L$w the $\mathfrak{U}$SVM will still perform a form of input selection even for nonlinear kernels. *The table shows results from a 20D AND and a 6D XOR toy problem each having only 2 relevant and $(n-2)$ noise features ($n=20,6r$).*

Method	Toy problem		Method	Toy problem	
	Linear	Non-Linear		Linear	Non-Linear
SVM$_{linear}$	16.0	49.2	$\mathfrak{U}_{L_1}$-SVM$_{linear}$	6.2	48.5
SVM$_{poly}$	15.6	23.0	$\mathfrak{U}_{L_1}$-SVM$_{poly}$	6.2	12.1
SVM$_{rbf}$	14.4	23.8	$\mathfrak{U}_{L_1}$-SVM$_{rbf}$	6.3	19.2

Summary and Conclusion

- We proposed an implementation of inference with an Universum as proposed by Vapnik 1998
- Our approximation yields a *convex quadratic problem*, that can be solved with standard SVM optimizers
- Universum is a method to incorporate *prior knowledge* about the problem *via data points* not priors on functions
  - *Universum examples* can often be *constructed* or easily *collected*
  - Universum might be *more intuitive* than prior over functions
  - The Universum makes use of additonal data like *noise injection* or *virtual examples* but does *neither require* the data to be from the *same distribution nor* to be *labeled*
  - Our approximation Universum can be seen as *data dependent regularizer*
- Future investigations
  - Effect of different universa on the choice of functions
  - Relate universum to Bayes priors on functions: How to get a universum from a prior and vice versa?