1. SVM primal formulation:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + C \sum \xi_i$$

subject to:

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i \leq n$$

where $H[\cdot] = \max(0, \cdot)$.

2. SVM dual formulation:

For $i = 1, \ldots, n$ set

$$K(x_i, x_j) = \langle x_i, x_j \rangle + 1$$

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

$$\min \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j K(x_i, x_j)$$

subject to:

$$\sum \alpha_i y_i = 0$$

and $\alpha_i \geq 0$.

### Inference with the Universum

What is inference with the universum about?

- 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 target function, $\mathcal{P}$ affects 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 $(x_1, y_1), \ldots, (x_n, y_n)$ and $x_{n+1}, \ldots, x_m$ and the set of equivalence classes $\mathcal{F} = \{[x_i], \ldots, [x_m]\}$ on $\mathcal{F}$.

- MAP: Define a prior $P$ over $\mathcal{F}$ and choose $[f] \in \mathcal{F}$ that has minimal empirical error and maximizes $P[f, \mathcal{P}]$.

- Universum [Yapap, 1998] Use another set $\mathcal{U} = \{x_1, \ldots, x_{n+1}\}$ to measure the “quality” of $[f]$ (call this set Universum).

- Use a prior of information in $\mathcal{U}$. Choose $f([f]) \in \mathcal{F}$ that has low empirical risk and has a maximum number of contradictions on $\mathcal{U}$, i.e. $\max \{i \in \mathcal{k} : |f([f]) &= y_i \}$.

- $\mathcal{U}$ is from the same domain and same problem category, but not from the same distribution.

- In contrast to semi-supervised learning, $\mathcal{U}$ is not from the same distribution and in contrast to the virtual support vector method or noise injection it does not need to be labeled.

- $\mathcal{U}$ reflects prior knowledge about the admissible set of examples whereas a prior over functions provides prior knowledge about the admissible set of decision functions.

- Universal examples can be constructed or collected in many problem settings.

### Approximation and Implementation

**Approximation of Contradiction Maximization on $\mathcal{U}$**

- Approximate maximization of contradictions by putting $x \in \mathcal{U}$ close to the decision boundary $f(x) = 0$.

- A small change in $f(x)$ will cause a contradiction on $x$.

- Choose $f([f]) \in \mathcal{F}$ with minimal real valued output on $\mathcal{U}$.

### Implementation in Support Vector Machines: The iSVM

- **Set of labeled examples:** $\mathcal{L} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

- **Set of universum examples:** $\mathcal{U} = \{(x_{n+1}, \ldots, x_m)\}$

- Express all loss functions in terms of Hinge loss $H[\cdot]$ = max(0, $\cdot$ - 1).

- Use Hinge loss $H[f(x)]$ for each labeled example $x \in \mathcal{L}$ in standard SVM.

- Use $\ell_1$-insensitive loss $\ell_1(f(x)] = H[f(x)] + \ell_1(-f(x)]$ for each universum example $x \in \mathcal{U}$. Note that applying $H[f(x)]$ to each $x \in \mathcal{U}$ is equivalent to applying $H[f_{\mathcal{P}}(x)]$ to two identical copies of $x$ with opposite labels.

- All loss functions are convex, therefore the optimization problem is convex.

### Examples

- **Results on WinMac using universe (iii):**

<table>
<thead>
<tr>
<th>Method</th>
<th>SVMtrain, SVMtest</th>
<th>50</th>
<th>100</th>
<th>050</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>SVMtrain, SVMtest</td>
<td>50</td>
<td>100</td>
<td>050</td>
<td>1000</td>
</tr>
<tr>
<td>iSVM</td>
<td>SVMtrain, SVMtest</td>
<td>50</td>
<td>100</td>
<td>050</td>
<td>1000</td>
</tr>
<tr>
<td>SVM</td>
<td>SVMtrain, SVMtest</td>
<td>50</td>
<td>100</td>
<td>050</td>
<td>1000</td>
</tr>
</tbody>
</table>

### Data Dependeral 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 universum can recover common regularizer $L_i$ Regualrizers.

- For recovering the isotropic $L_2$ regularizer assume $b = 0$, let $\mathcal{U}_i = \{x_1, x_2, \ldots, x_{k+1}\}$ and use quadratic loss $\ell_2[\cdot] = (\cdot - y)^2$ for the points in $\mathcal{U}_i$. Then:

  $$\sum_{U} \ell_2[\alpha_i] = \sum \alpha_i (y_i - x_i)^2 = \sum \alpha_i ||y_i - b||^2$$

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

  $$\sum_{U} \ell_2[\alpha_i] = \sum \alpha_i (y_i - x_i)^2 = \sum \alpha_i ||y_i - b||^2$$

### 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 can be often constructed or easily collected.

- Universum might be more intuitive than prior over functions.

- The universum makes use of additional 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 on different universum on the choice of functions.
  - Relate universum to Bayes priors on functions: How to get a universum from a prior and vice versa?