Previously on SVMs...

Soft Margin SVM

\[
\begin{align*}
\mathbf{w} \cdot \mathbf{x} - b &\geq 1 - \xi_+, k \\
\mathbf{w} \cdot \mathbf{x} - b &\geq -1 + \xi_o, k \\
\xi_k &\geq 0
\end{align*}
\]

Slack variables

\[\xi_k = 0 \text{ standard SVM}\]

\[
\minimize \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{k=1}^{K} \xi_k
\]

\(C\) TRADEOFF PARAMETER BETWEEN ERROR AND MARGIN
Previously on SVMs…

Formulation based on

**INNER PRODUCT** \( \mathbf{w} \cdot \mathbf{x} \)

compute the inner product in the feature space

Define the kernel function

\[
K(\mathbf{w}, \mathbf{x}) = \phi(\mathbf{w}) \cdot \phi(\mathbf{x})
\]

No explicit data transformation

\[
\phi(x) = \phi \left( \begin{array}{c} x_1 \\ x_2 \end{array} \right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)
\]

\[
K(\mathbf{w}, \mathbf{x}) = \phi(\mathbf{w}) \cdot \phi(\mathbf{x}) = (1 + w_1x_1 + w_2x_2)^2
\]
**Popular Kernels**

**Previously on SVMs**

- **Polynomial Kernel**
  \[ K(w, x) = (\gamma (w \cdot x) + r)^d \]

- **Radial Basis Function Kernel (RBF)**
  \[ K(w, x) = e^{-\gamma |w-x|^2}, \gamma > 0 \]

- **Sigmoid Kernel**
  \[ K(w, x) = tanh(w \cdot x + r) \]

**Usage:**
```
svm-train [opt] train_file [model_file]
```

**Options:**
- `-t kernel_type`: set type of kernel function
  - 0 -- linear: u'*v
  - 1 -- polynomial: (gamma*u'*v + coef0)^degree
  - 2 -- radial basis function: exp(-gamma*|u-v|^2)
  - 3 -- sigmoid: tanh(gamma*u'*v + coef0)
- `-d degree`: set degree in kernel function (default 3)
- `-g gamma`: set gamma in kernel function (default 1/n)
- `-r coef0`: set coef0 in kernel function (default 0)
- `-c cost`: set the parameter C of C-SVC (default 1)
- `-e epsilon`: set tolerance of termination criterion (default 0.001)
- `-b probability_estimates`: whether to train a model for probability estimates, 0 or 1 (default 0)
- `-q`: quiet mode (no outputs)
  ...

Motivations

Model Selection

How do we select the “optimal” model and parameter(s) for a given classification problem?

Performance estimation

How do we estimate the performances for a couple model-problem?
How well a model generalizes to new data?

Model complexity:
- Low bias, low variance: Low complexity
- High bias, high variance: High complexity

Prediction error:
- Low error: Well-generalized model
- High error: Overfitting or underfitting

Graphs show:
- Training error decreases with increasing model complexity
- Test error decreases initially, then increases after a certain complexity level
Validation set approach

Training set = small part of entire

**FIRST IDEA:** USE ALL THE DATA

- **OVERFITTING**
- **UNDERESTIMATE ERRORS**
The holdout method

Training set
Test set

Entire dataset

Difficult for sparse dataset
Misleading: highly variable
Only a subset is validated
Only a subset is fitted

Prediction Error

Stopping point

Test
Train

OVERESTIMATE ERRORS
Random Subsampling

Entire dataset

Exp. 1

Exp. 2

Exp. K

**ERROR ESTIMATION**

\[
E = \frac{1}{K} \sum_{k=1}^{K} E_k
\]
K-Fold Cross-validation

Entire dataset

Exp.1

Exp.2

Exp.2

Exp.K

all the samples are used for training

ERROR ESTIMATION

\[ E = \frac{1}{K} \sum_{k=1}^{K} E_k \]
Leave-one-out Cross Validation

Entire dataset

Exp.1

Exp.2

Exp.2

Exp.N

Single sample

ERROR ESTIMATION

\[ E = \frac{1}{N} \sum_{k=1}^{N} E_k \]
Which approach?
How many folds are needed?

Large number
+ Very accurate
- Large variance
- More time

Small number
+ Small variance
+ Less time
- Conservative

Practically
Depends on the dataset size
Large dataset $\Rightarrow K \rightarrow 2$
Sparse dataset $\Rightarrow K \rightarrow N$
Common choice $\Rightarrow K \approx 10$
• Divide the available data into training, validation and test set
• Select architecture and training parameters
• Train the model using the training set for each fold
• Evaluate the model using the validation set
• Repeat using different architectures and training parameters
• Select the best model and train it using data from the training and validation set
• Assess this final model using the test set
• DO NOT tune the model any further
Three-way data splits
Practical session
- Start with RBF Kernel (only 2 parameters)
- Try exponentially growing sequences
  \[ C = 2^{-5}, 2^{-3}, \ldots, 2^{15} \]
  \[ \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3} \]
- Conduct a finer search on the best neighborhood
Questions?